



MT-CNC Quick Reference Guide

Reference Manual

DOK-MT*CNC-REFER*GUIDE-ANW1-EN-P

278478

Using This Reference Guide

This document is a quick reference guide for individuals working with the Indramat MT-CNC. This guide should not be used as a complete system manual, but as a quick reference guide for plant floor programming.

Modules described in this manual include: MT-CNC, TVD power supplies, DDS drive amplifiers, MDD servo motors, TDA/KDA/RAC spindle amplifiers, spindle motors.

In this document **MUI** refers to the MT-CNC **User Interface** software, and **SPS** refers to the PLC programming system.

Hot Key Legend

<A>, : Press key "A" then press key "B".

<A>+: Press and hold key "A" while pressing key "B".

<Main Menu>: Always refers to the MT-CNC main Menu.

NOTE:

This reference guide is based on the MT-CNC software version 0x.17/XX.

Later MT-CNC software versions may contain functionality that deviates from the one described in this manual!

CAUTION:

Life threatening voltages present on amplifier and servo motor connectors. Do not connect or disconnect while power is on. Unexpected and dangerous machine movements may result if service is attempted by unqualified personnel.

FOREWORD

Special Notations:

Special notations are used in this manual to assist the reader in identifying unique conditions or information that is important. Three categories of notations are listed below in ascending order of importance.

Note: A **NOTE** is a tip, suggestion or emphasized procedure for operating the equipment.

Caution: A **CAUTION** appears when a condition exists which could cause operating faults or damage to the equipment.

Warning: **WARNING** statements identify conditions which could cause bodily harm and/or severe damage to the equipment if the operator is not careful operating the equipment. A **WARNING** typically describes the potential hazard, its possible effect, and measures that must be taken to avoid the hazard.

Please NOTE: The following notes pertain to the version of software releases covered in this manual.

No note: valid for all software releases indicated under revisions
(1): MT-CNC software releases 04.11/00 ... 04.13/xx only
(2): MT-CNC software releases 0x.14/xx
(3): MT-CNC software releases 0x.15/xx
(4): MT-CNC software releases 0x.16/xx
(5): MT-CNC software releases 0x.17/00

The Rexroth Corporation
Indramat Division
5150 Prairie Stone Parkway
Hoffman Estates, Illinois 60192

Phone. (847) 645-3600 | FAX (847) 645-6201

RECORD OF REVISIONS

Revision	Date	Description of Change
Rev A	Jan.1995	1st Preliminary Printing, valid for software release 0x.11/00 - 0x.13/xx
Rev B	Oct.1995	Improved, corrected and expanded version per software releases 0x.11/00 - 0x.15/xx
Rev C	Oct.1997	Improved, corrected and expanded version per software releases 0x.16/00 - 0x.17/00

CONTENTS

FOREWORD	III
RECORD OF REVISIONS	IV
1. MUI FLOW CHARTS	1-1
1.1 MT-CNC USER PROGRAMS & DATA - BACKUP.....	1-1
1.2 MT-CNC USER PROGRAMS & DATA - RESTORE	1-3
1.3 DIGITAL DRIVE PARAMETER (SERVO, SPINDLE) - ARCHIVE	1-5
1.4 ACTIVE USER DATA (EVENTS, VARIABLES, D-CORRECTIONS, ACTIVE TOOL LIST, OFFSET TABLE DATA) - ARCHIVE	1-6
1.5 NC PROGRAM (INCLUDING TOOL SETUP LIST) - ARCHIVE	1-7
1.6 MACHINE PARAMETER (SYSTEM, PROCESS, AXIS) - ARCHIVE	1-8
1.7 SPS (PLC) PROGRAM - ARCHIVE ^{(1),(2),(3)}	1-9
1.8 SPS (PLC) PROGRAM - ARCHIVE ⁽⁴⁾	1-10
1.9 SPS (PLC) PROGRAM - ARCHIVE ⁽⁵⁾	1-11
1.10 DIGITAL DRIVE PARAMETER (SERVO, SPINDLE) - EDITING	1-12
1.11 MACHINE PARAMETER (SYSTEM, PROCESS, AXIS) - EDITING	1-13
1.12 SPS (PLC) EDITING ^{(1),(2),(3)}	1-15
1.13 SPS (PLC) EDITING ^{(4),(5)}	1-16
1.14 USER MESSAGE - EDITING.....	1-17
1.15 DIGITAL DRIVE PARAMETER (SERVO, SPINDLE) - PRINT OUT	1-18
1.16 MACHINE PARAMETER (SYSTEM, PROCESS, AXIS) - PRINT OUT	1-19
1.17 NC PROGRAM - PRINT OUT.....	1-20
1.18 SPS DOCUMENTATION - COMPLETE AND PARTIAL PRINTOUTS ^{(1),(2),(3)}	1-21
1.19 SPS DOCUMENTATION - COMPLETE AND PARTIAL PRINTOUTS	1-22
1.20 GLOBAL DOCUMENTATION - PRINTING USER	1-23
1.21 FURTHER INFORMATION (HELP) - NC STATUS AND DIAGNOSTICS	1-24
1.22 NC PROGRAM VARIABLES	1-25
1.23 OFFSET TABLE DATA	1-26
1.24 SPS LOGIC DISPLAY SETUP.....	1-27
1.25 SPS LOGIC ANALYSIS ^{(1),(2),(3)}	1-28
1.26 SPS LOGIC ANALYSIS.....	1-29
1.27 PASSWORD SETUP (FOR MT-CNC MUI SOFTWARE)	1-30
1.28 SELECTION OF ONLINE / OFFLINE MODE (FOR MUI SOFTWARE)	1-31
1.29 SELECTION OF ONLINE / OFFLINE MODE (FOR MUI SOFTWARE)	1-32
1.30 EXITING (FROM ANY SCREEN BACK TO THE MUI MAIN MENU).....	1-33
1.31 OSCILLOSCOPE - SETUP, RECORD, VIEW, SAVE	1-34
1.32 SOT MESSAGE - DOWNLOADING	1-36
1.33 CURRENT NC PROGRAM - VIEWING.....	1-37
2. GUI FLOW CHARTS	2-1
2.1 SETUP, GUI (INDRAMAT DEFAULT).....	2-2
2.1.1 Online / Off-line Mode, GUI (Indramat default)	2-2
2.1.2 Password Setup, GUI (Indramat default).....	2-3
2.1.3 Exiting Menus, GUI (Indramat default)	2-4
2.1.4 Load SOT Messages, GUI (Indramat default).....	2-5
2.2 BACKUP, GUI (INDRAMAT DEFAULT).....	2-6
2.2.1 Total MT-CNC User Programs & Data Backup, GUI (Indramat default).....	2-6
2.2.2 Digital Drive Parameter Archive, GUI (Indramat default).....	2-7
2.2.3 Active User Data Archive, GUI (Indramat default).....	2-8
2.2.4 SPS (PLC) Project Archive, GUI ^{(1),(2),(3)} (Indramat default).....	2-9

2.2.5 SPS (PLC) Project Archive, GUI ⁽⁴⁾ (Indramat default).....	2-10
2.2.6 SPS (PLC) Project Archive, GUI ⁽⁵⁾ (Indramat default).....	2-11
2.2.7 Machine Parameter Archive, GUI (Indramat default).....	2-12
2.2.8 NC-program (w. Tool Setup List) Archive, GUI (Indramat default).....	2-13
2.2.9 Tool List Archive, GUI (Indramat default).....	2-14
2.3 RESTORE, GUI (INDRAMAT DEFAULT).....	2-15
2.3.1 Total MT-CNC User Programs & Data Restore, GUI (Indramat default).....	2-15
2.4 TROUBLESHOOTING AND DIAGNOSTICS, GUI (INDRAMAT DEFAULT).....	2-16
2.4.1 Further Information to Diagnostics, GUI (Indramat default).....	2-16
2.4.2 Offset Table Data, GUI (Indramat default).....	2-17
2.4.3 Active NC-Program, GUI (Indramat default).....	2-18
2.4.4 NC-Variables, GUI (Indramat default).....	2-19
2.4.5 Oscilloscope, GUI (Indramat default).....	2-20
2.4.6 PLC (SPS) Logic Display Setup, GUI (Indramat default).....	2-22
2.4.7 PLC (SPS) Logic Analysis, GUI ^{(1),(2),(3)} (Indramat default).....	2-23
2.4.8 PLC (SPS) Logic Analysis, GUI ^{(4),(5)} (Indramat default).....	2-24
2.5 EDITING, GUI (INDRAMAT DEFAULT).....	2-25
2.5.1 Machine Parameter Editing, GUI (Indramat default).....	2-25
2.5.2 Diagnostic Messages Editing, GUI (Indramat default).....	2-27
2.5.3 SPS (PLC) Program Editing, GUI ^{(1),(2),(3)} (Indramat default).....	2-28
2.5.4 SPS (PLC) Program Editing, GUI ^{(4),(5)} (Indramat default).....	2-29
2.5.5 Digital Drive Parameter Editing, GUI (Indramat default).....	2-30
2.6 DOCUMENTATION, GUI (INDRAMAT DEFAULT).....	2-31
2.6.1 Global Documentation, GUI (Indramat default).....	2-31
2.6.2 NC-Program Print Out, GUI (Indramat default).....	2-32
2.6.3 Machine Parameter Print Out, GUI (Indramat default).....	2-33
2.6.4 Digital Drive Parameter Print Out, GUI (Indramat default).....	2-34
2.6.5 SPS (PLC) Documentation Print Out, GUI ^{(1),(2),(3)} (Indramat default).....	2-35
2.6.6 SPS (PLC) Documentation Print Out, GUI ^{(4),(5)} (Indramat default).....	2-36
3. PARAMETERS.....	3-1
3.1 SYSTEM PARAMETERS.....	3-1
3.2 PROCESS PARAMETERS.....	3-2
3.3 AXIS PARAMETERS.....	3-4
3.3.1 Linear and Rotary Axes.....	3-4
3.3.2 Main Spindle Axis Parameters.....	3-9
3.3.3 Digital Comb. Turret Axis / Main Spindle, C-Axis Parameters *D.....	3-11
4. NC PROGRAMMING.....	4-1
4.1 NC PROGRAM STRUCTURE.....	4-1
4.2 NC MEMORY.....	4-1
4.3 ADVANCE PROGRAM.....	4-1
4.4 REVERSE PROGRAM.....	4-1
4.5 PART PROGRAM.....	4-2
4.6 PROCESS PROGRAM.....	4-2
4.7 NC-BLOCK ELEMENTS.....	4-2
4.7.1 NC-Block Numbers.....	4-2
4.7.2 NC-block which can be Skipped.....	4-2
4.7.3 NC-Labels.....	4-2
4.7.4 NC Message for MUI.....	4-2
4.7.5 NC Message for SOT.....	4-3
4.7.6 NC-Block Comment.....	4-3
4.8 G-CODES.....	4-3
4.9 AUXILIARY FUNCTIONS.....	4-11
4.10 PROCESS CONTROL COMMANDS.....	4-14

4.11 PROGRAM CONTROL NC COMMANDS	4-15
4.12 SUBROUTINES.....	4-16
4.13 REVERSE VECTOR	4-16
4.14 RE-POSITIONING AND NC-BLOCK RESTART (GUI).....	4-17
4.15 5 BRANCH COMMANDS	4-17
4.16 NC-EVENTS.....	4-18
4.17 INTERRUPTING NC-EVENTS.....	4-18
4.18 MATHEMATICAL EXPRESSIONS.....	4-19
4.19 NC-FUNCTIONS	4-20
4.20 TIME FUNCTIONS.....	4-22
4.21 POSITION STROBE.....	4-22
4.22 FUNCTIONS FOR DIGITAL DRIVES	4-22
4.23 NC-VARIABLES	4-22
4.24 SPINDLE SELECTIONS.....	4-23
4.25 OFFSET TABLES.....	4-23
4.26 AXIS SWITCHING	4-24
4.27 PATH ACCELERATION.....	4-25
4.28 MACHINE DATA	4-26
4.29 TOOL DATA.....	4-27
4.30 MODAL FUNCTIONS.....	4-27
4.31 NC-BLOCK LOOK AHEAD FUNCTION	4-27
4.31.1 Global Variable Definitions	4-27
5. SPS INSTRUCTION SET	5-1
5.1 ELEMENTARY DATA TYPES.....	5-1
5.2 DERIVED DATA TYPES	5-1
5.2.1 Structures (STRUCT).....	5-1
5.2.2 Arrays	5-1
5.3 INSTRUCTION LIST INSTRUCTIONS	5-2
5.3.1 Load and save instructions	5-2
5.3.2 Set and reset instructions.....	5-2
5.3.3 Logic Instructions.....	5-3
5.3.4 Jumps, Calls, Return (conditional and unconditional).....	5-4
5.3.4.1 Return from Function Blocks or Functions	5-4
5.3.5 Arithmetic Instructions	5-5
5.3.6 Comparators.....	5-7
5.4 LADDER DIAGRAM	5-9
5.4.1 Basic Elements in the Ladder Diagram.....	5-10
5.5 OPERATIONS.....	5-11
5.5.1 Logic Bit and Bit String Operations ‘:=’, ‘AND’, ‘OR’, ‘XOR’.....	5-11
5.5.2 Arithmetic Operations	5-12
5.5.3 Comparators.....	5-15
5.6 FUNCTIONS.....	5-17
5.6.1 Standard Functions	5-17
5.6.2 Numeric Functions	5-27
5.6.3 Functions for Time-to-Integer Conversion.....	5-27
5.6.3.1 Method of operation of the time-to-integer conversion.....	5-28
5.6.3.2 Error handling of the time-to-integer conversion	5-28
5.6.4 INTEGER-to-TIME Conversion	5-29
5.6.5 Bit string functions.....	5-29
5.6.6 Character String Functions	5-33
5.7 FIRMWARE FUNCTIONS.....	5-38
5.8 USER FUNCTIONS	5-38
5.8.1 Import rules	5-39
5.9 FUNCTION BLOCKS.....	5-39

5.10 STANDARD FUNCTION BLOCKS.....	5-39
5.10.1 Flip-Flops.....	5-39
5.10.2 Edge evaluation for rising and falling edges 'R_TRIG', 'F_TRIG'.....	5-40
5.10.3 Up and down counters.....	5-41
5.10.4 Time stages for pulses; ON/OFF delay.....	5-43
5.10.5 Function blocks for date and time.....	5-44
5.10.6 Function blocks for serial interfaces.....	5-45
5.10.6.1 Standard data type 'COM'.....	5-45
5.10.6.2 Error handling of the function blocks for serial interfaces.....	5-52
5.11 FIRMWARE FUNCTION BLOCKS.....	5-53
5.12 USER FUNCTION BLOCKS.....	5-53
5.12.1 Import rules.....	5-53
5.13 ERROR HANDLING.....	5-54
5.13.1 Basics of the Error Handling Concept.....	5-54
5.13.2 Error Handling Sequence.....	5-55
5.13.3 Error Handling of Multiple Errors.....	5-56
5.13.4 Error Handling in User Files.....	5-56
5.13.5 Overview of Possible Errors and Their Causes.....	5-56
5.13.6 Errors in functions and function blocks.....	5-57
5.13.7 Errors in operations and IL instructions.....	5-62
5.13.8 Errors in exceptional REAL operations.....	5-64
6. GATEWAY SIGNALS.....	6-1
6.1 INTRODUCTION TO CNC \Leftrightarrow SPS GATEWAY SIGNALS.....	6-1
6.2 DECLARING GATEWAYS IN SPS.....	6-1
6.3 PROCESS SIGNALS.....	6-2
6.4 POWER ON CONTROL.....	6-4
6.5 CNC OPERATING MODES.....	6-5
6.5.1 AUTOMATIC MODE.....	6-6
6.5.2 SEMI-AUTOMATIC MODE.....	6-7
6.5.3 MANUAL MODE (Setup).....	6-8
6.6 NC-PROGRAM SELECTION.....	6-10
6.7 PROCESS CONTROL SIGNALS (SPS \Rightarrow CNC).....	6-11
6.8 PROCESS STATUS SIGNALS (CNC \Rightarrow SPS).....	6-14
6.9 FEED AND SPINDLE OVERRIDE (SPS \Rightarrow CNC).....	6-18
6.10 PROCESS / PART PROGRAM SYNCHRONIZATION.....	6-19
6.11 SPINDLE CONTROL (SPS \Rightarrow CNC).....	6-23
6.12 SYNCHRONOUS AXES.....	6-24
6.13 MECHANISM CONTROL SIGNALS.....	6-26
6.14 MECHANISM STATUS SIGNALS.....	6-26
6.15 AXIS SIGNALS.....	6-27
6.15.1 Axes Hardware Limits (SPS \Rightarrow CNC).....	6-27
6.15.2 Axis Control Signals (SPS \Rightarrow CNC).....	6-28
6.15.3 Axis Status Signals (CNC \Rightarrow SPS).....	6-36
6.16 MT-CNC SPECIFIC APR PARAMETERS.....	6-39
6.16.1 SCARA Robots.....	6-39
6.16.2 DDS as Spindle Drive.....	6-41
6.16.3 Gear Change.....	6-42
6.16.4 Torque Reduction.....	6-43
6.16.5 Filter.....	6-43
6.16.6 Synchronous Axis.....	6-44
6.16.7 Measuring Function Torque / Lag Error.....	6-46
6.16.8 Combined Spindle / Turret Axis.....	6-47
6.16.9 General.....	6-47

6.17 GATEWAY SIGNAL LISTING (STRUCTURED DATA TYPE).....	6-52
6.17.1 Axis Status Signals (iAXIS).....	6-52
6.17.2 Process Status Signals (iPROC).....	6-52
6.17.3 Local Inputs Signals (iDEA).....	6-53
6.17.4 External Mechanism Status Signals (iMECH).....	6-54
6.17.5 Axis Control Signals (qAXIS).....	6-55
6.17.6 Process Control Signals (qPROC).....	6-55
6.17.7 Local Output Signals (qDEA).....	6-56
6.17.8 External Mechanism Control Signals (qMECH).....	6-56
7. TOOL MANAGEMENT.....	7-1
7.1 INTRODUCTION TO TOOL MANAGEMENT.....	7-1
7.2 TOOL SETUP LIST.....	7-1
7.2.1 Tool Lists and Setup Lists.....	7-1
7.2.2 Basic Tool Data of Setup List.....	7-1
7.2.2.1 Tool Name (ID).....	7-1
7.2.2.2 T-Number (T).....	7-1
7.2.2.3 Amount of Tool Edges.....	7-1
7.2.2.4 Time Unit.....	7-2
7.2.2.5 Length Unit.....	7-2
7.2.2.6 Comment.....	7-2
7.2.3 Tool Edge Data.....	7-2
7.2.3.1 Tool Edge Orientation.....	7-2
7.2.3.2 Maximum Tool Life.....	7-2
7.2.3.3 Tool Life.....	7-2
7.2.3.4 Maximum and Minimum Tool Length.....	7-2
7.2.3.5 Maximum and Minimum Tool Radius.....	7-2
7.2.3.6 Length Tool Wear Factor (L1, L2, L3).....	7-3
7.2.3.7 Radius Tool Wear Factor.....	7-3
7.2.4 Setup List Organization.....	7-3
7.2.5 Setup List Within the MT-CNC.....	7-3
7.3 TOOL LIST.....	7-4
7.3.1 Tool List Data.....	7-4
7.3.2 Basic Tool Data of Setup List.....	7-4
7.3.2.1 Tool Name.....	7-4
7.3.2.2 Tool Storage.....	7-4
7.3.2.3 Tool Location.....	7-4
7.3.2.4 Tool Index.....	7-4
7.3.2.5 Tool Correction Type.....	7-5
7.3.2.6 Amount of Tool Edges.....	7-5
7.3.2.7 Half Pocket Overlap.....	7-5
7.3.2.8 Old Location.....	7-5
7.3.2.9 Length Unit.....	7-6
7.3.2.10 User Tool Data (1...9).....	7-6
7.3.3 Tool Edge Data.....	7-6
7.3.3.1 Tool Edge Orientation.....	7-6
7.3.3.2 Remaining Tool Life.....	7-6
7.3.3.3 Warning Limit.....	7-6
7.3.3.4 Geometric Data.....	7-7
7.3.3.5 Tool Length Correction.....	7-7
7.3.3.6 Tool Radius Correction.....	7-7
7.3.3.7 User Edge Data.....	7-7
7.3.4 Tool List within the Control System.....	7-7
7.4 ACTIVE TOOL LIST.....	7-8
7.4.1 Data of the Active Tool List.....	7-8
7.4.2 Tool Edge Status Bits.....	7-13
7.5 TOOL MANAGEMENT PARAMETERS.....	7-14
7.5.1 System Parameters.....	7-14

7.6 NC TOOL MANAGEMENT CONTROL FUNCTIONS	7-18
7.6.1 Requirements	7-18
7.6.1.1 Defined Plane.....	7-18
7.6.1.2 Tool Selection.....	7-19
7.6.1.3 Tool Edge Selection (E).....	7-19
7.6.1.4 Tool Change.....	7-20
7.6.1.5 Spindle Selection for Tool Mgmt.....	7-20
7.6.2 Tool Storage Movement NC-commands	7-20
7.6.3 Tool Change NC-commands.....	7-23
7.7 SPS TOOL MANAGEMENT CONTROL FUNCTIONS	7-26
7.7.1 Standard Data Types	7-26
7.7.2 Tool Storage Reference.....	7-27
7.7.3 Position Tool Storage	7-28
7.7.4 Standard Tool Change FUNCTIONS.....	7-29
7.7.5 Tool Change Between Magazine and Spindle.....	7-30
7.7.6 Tool Transfer FUNCTIONS	7-32
7.7.7 Tool Transfer FUNCTION Overview.....	7-35
7.7.8 Combined Spindle / Turret Axis.....	7-37
7.7.9 Standard FUNCTION_BLOCKS for Tool Management.....	7-40
7.7.9.1 Standard FUNCTION_BLOCKS for Basic Tool Data.....	7-40
7.7.9.2 Standard FUNCTION_BLOCKS for Tool Edge Data	7-41
7.7.9.3 Standard FUNCTION_BLOCKS for Tool Data	7-42
7.7.9.4 Standard FUNCTION_BLOCK for D-Corrections.....	7-44
7.8 TOOL MANAGEMENT GATEWAY SIGNALS	7-44
7.8.1 Tool Management Control Signals (SPS⇒CNC)	7-44
7.8.2 Tool Management Status Signals (CNC⇒SPS).....	7-46
8. STATUS, DIAGNOSTICS AND SETUP	8-1
8.1 NC STATUS AND DIAGNOSTICS.....	8-1
8.2 FAULT LIST	8-2
8.3 HARDWARE AND SOFTWARE REQUIREMENTS	8-2
8.3.1 Computer Requirements (PC).....	8-2
8.3.2 MS-DOS Software Configuration Recommendations for compatible PCs	8-3
8.3.2.1 Example of the CONFIG.SYS file of compatible PCs	8-3
8.3.2.2 The CONFIG.SYS file for Indramat's BTV 01.....	8-3
8.3.2.3 Example of the AUTOEXEC.BAT File for compatible PCs.....	8-4
8.3.2.4 The AUTOEXEC.BAT File of Indramat's BTV 01.....	8-4
8.4 MT-RUN.BAT FILE.....	8-5
8.4.1 Software Options Which Can Appended to (TSRPG25I) Line of MT-RUN.BAT.....	8-5
8.4.2 Software Options Which Can be Appended to the MT-CNC Line of MT-RUN.BAT.....	8-6
8.5 HOT KEYS	8-6
8.6 WINDOWS NT3.51 SOFTWARE CONFIGURATION RECOMMENDATIONS FOR COMPATIBLE PCs.....	8-6
8.6.1 Windows NT Interface Drivers.....	8-7
8.6.2 Windows NT Task Response Time.....	8-8
8.6.3 PIF File Setting for the Window NT MS DOS Window	8-9
8.6.4 Batch File to Call Up the MT-CNC User Interface Software	8-11
8.6.5 Create MT-CNC User Interface Program Icon	8-12
8.7 DDE SERVER DEPLOYMENT CONDITIONS	8-13
8.7.1 Required Software for the DDE Server.....	8-13
8.7.2 Cooperation with other Software Components	8-13
8.8 GENERAL.....	8-14
8.9 PROGRAM CALL.....	8-14
8.10 PASSWORD ADMINISTRATION	8-15
8.10.1 First Execution of Password Administration	8-15
8.10.2 Register New User	8-16
8.11 DDE CLIENT CONNECT TO INDRAMAT'S DDE SERVER	8-16

8.11.1 Automatic Start.....8-16
8.11.2 Data Access Function Commands.....8-17

List Of Appendices

Appendix A: MT-CNC Control system
Appendix B: *this appendix is left intentionally blank*
Appendix CD: MT-CNC Module Rack MTCB
Appendix E: Central Processor module CPUB
Appendix F: SPS Module PLCB
Appendix G: Axis Processor Module APRB
Appendix H: Input/Output Module DEAB
Appendix IJ: *this appendix is left intentionally blank*
Appendix K: Operating, Programming and Visualizing Terminal BTV01.3
Appendix L: Configurable Operator Panel BTM13
Appendix M: Configurable Operator Panel BTM03
Appendix NO: Configurable Operator Panel BTM04
Appendix PQ: Station Operator Terminal SOT02
Appendix R: Remote I/O Rack RECO
Appendix S: Remote I/O Rack RECO02
Appendix Sch: Fiber Optic Repeater/Switcher LWR01
Appendix St: Firmware and Software selection for the MT-CNC Control System
Appendix T-V: Power Switching in E-Stop Chain
Appendix W: Hand Terminal BTC05
Appendix X: Master Index for Chapters 1 - 8

1. MUI FLOW CHARTS

This chapter contains flowcharts to perform commonly used functions with the standard Indramat softkey menu of the MT-CNC Machine User Interface Software

1.1 MT-CNC User Programs & Data - Backup

Figure 1-1: MT-CNC User Programs & Data - Backup (Sheet 1 of 2)

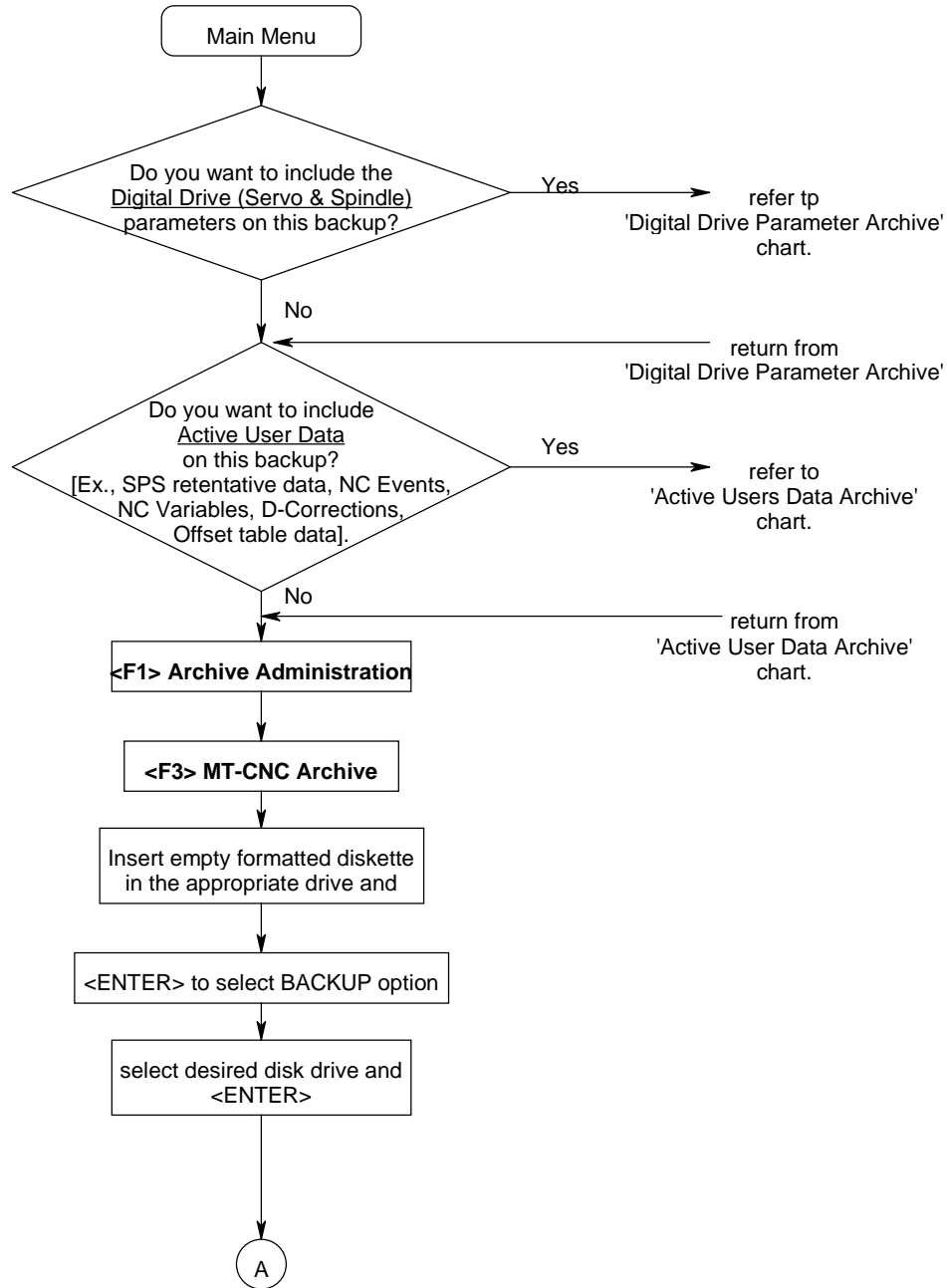
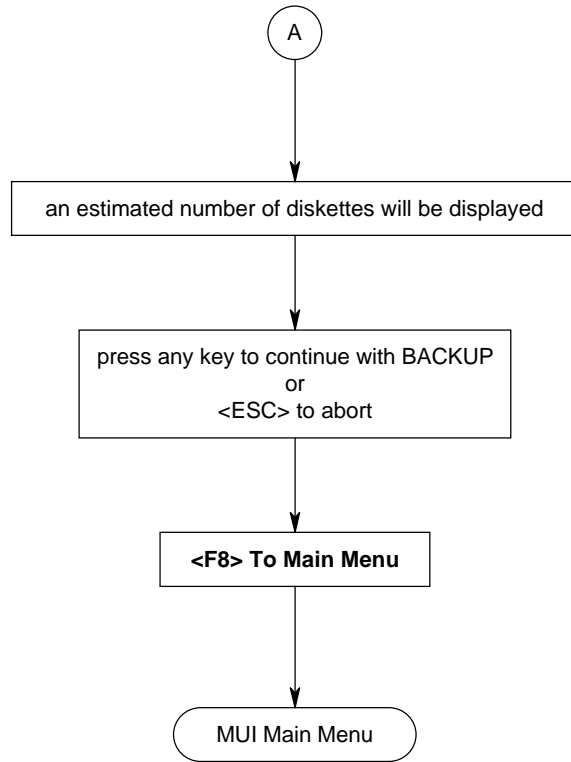


Figure 1-2 MT-CNC User Programs & Data - Backup (Sheet 2 of 2)



1.2 MT-CNC User Programs & Data - Restore

Figure 1-3 MT-CNC User Programs & Data - Restore - (Sheet 1 of 2)

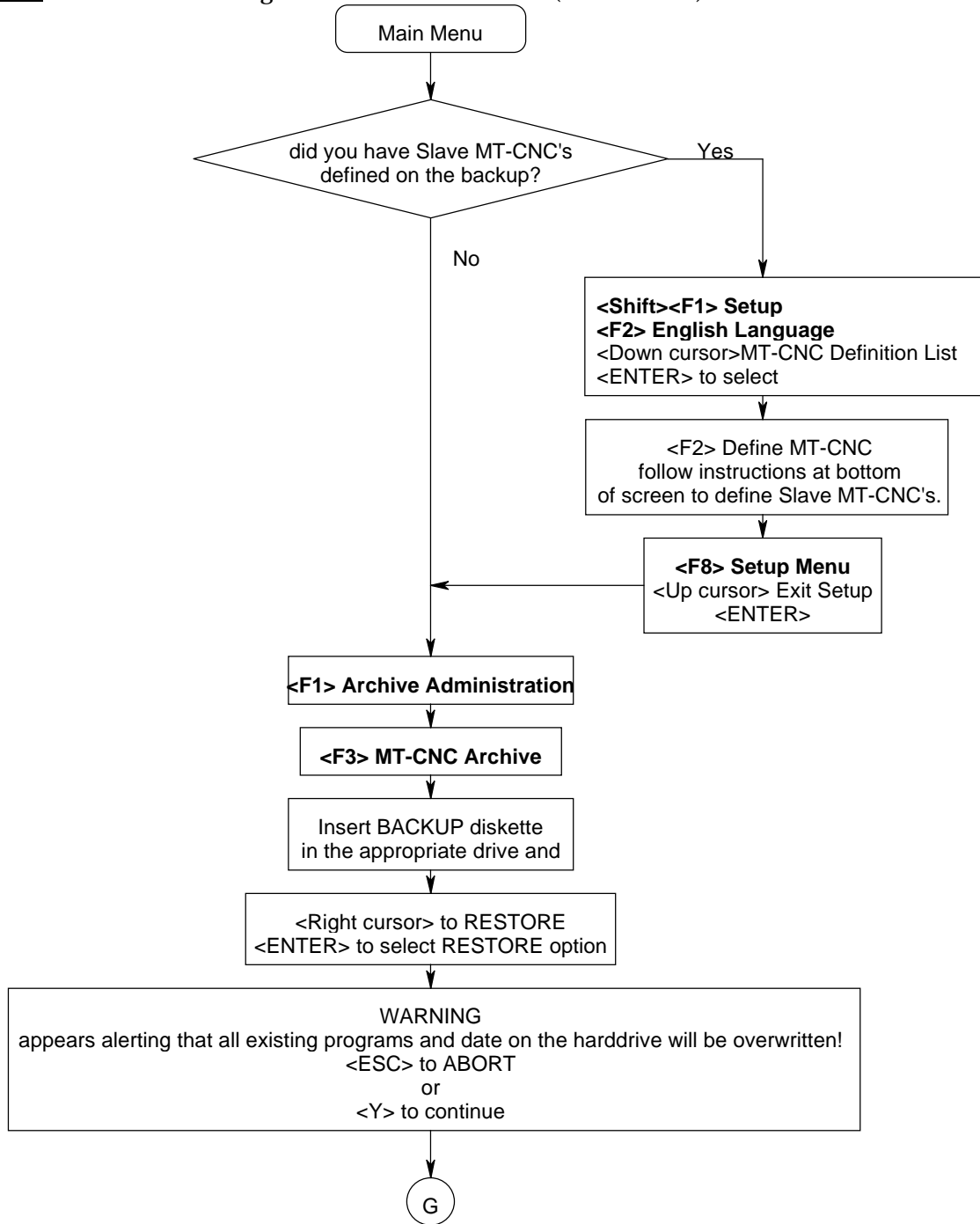
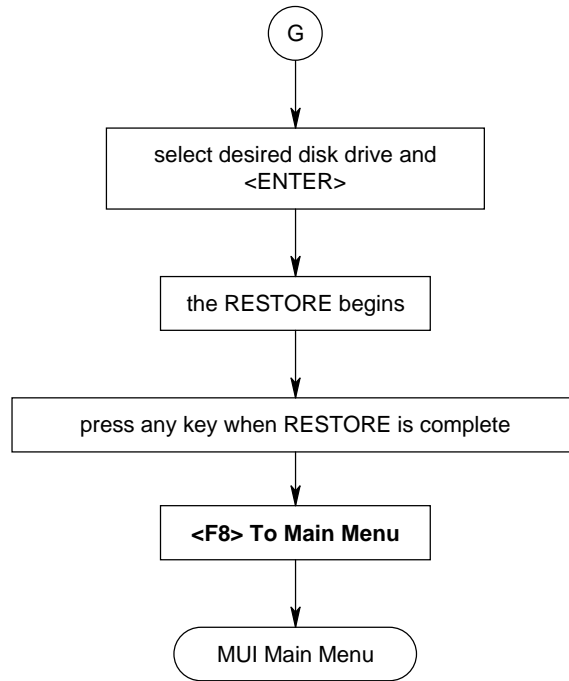
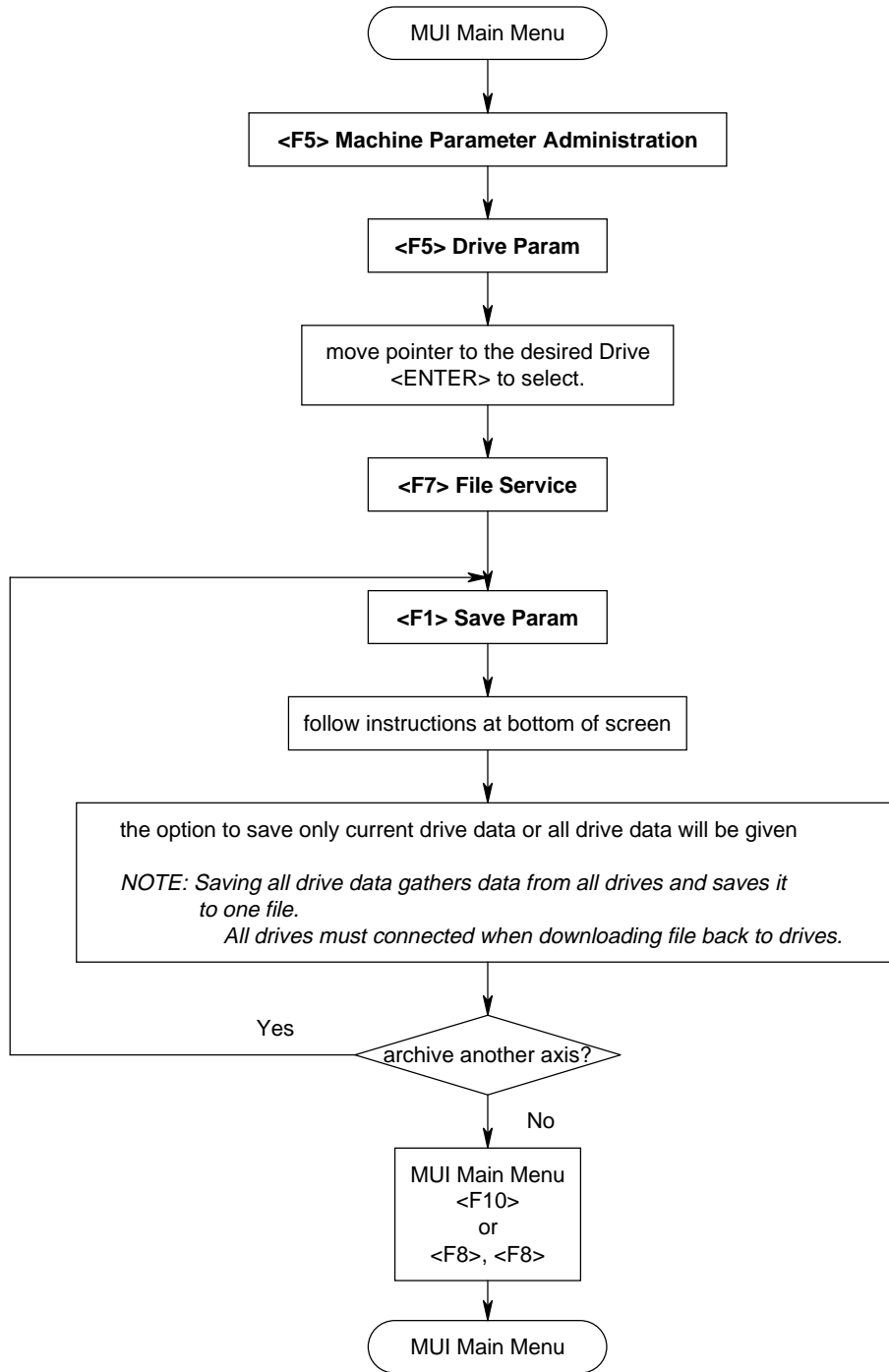


Figure 1-4 MT-CNC User Programs & Data - Restore - (Sheet 2 of 2)



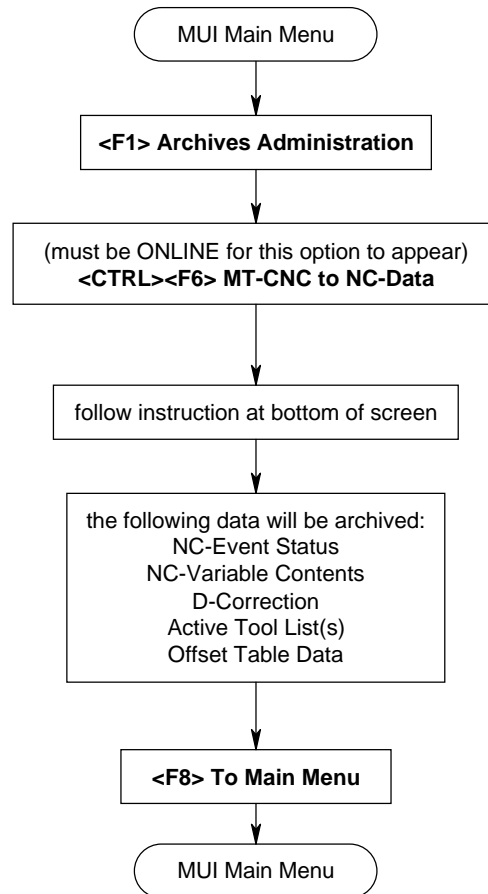
1.3 Digital Drive Parameter (Servo, Spindle) - Archive

Figure 1-5 Digital Drive Parameter (Servo, Spindle) - Archive

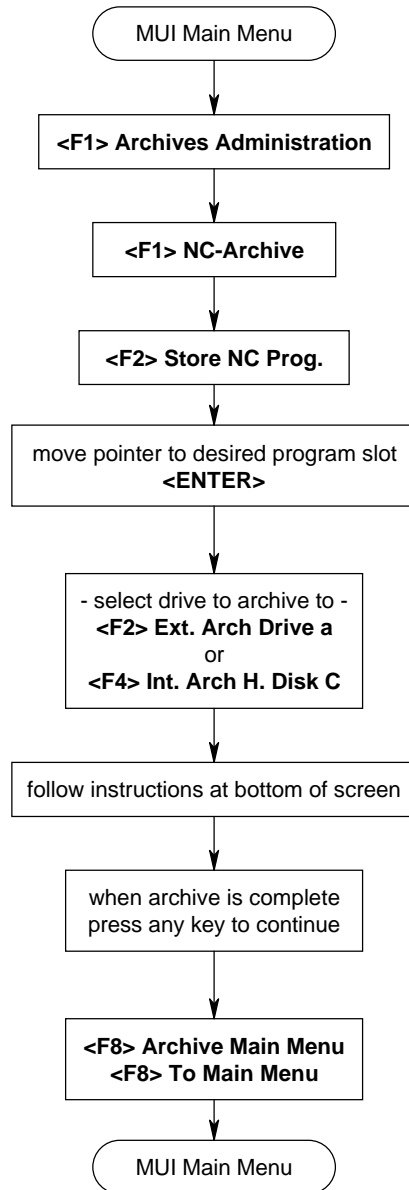


1.4 Active User Data (Events, Variables, D-Corrections, Active Tool List, Offset Table Data) - Archive

Figure 1-6 Active User Data (Events, Variables, D-Corrections, Active Tool List, Offset Table Data) - Archive

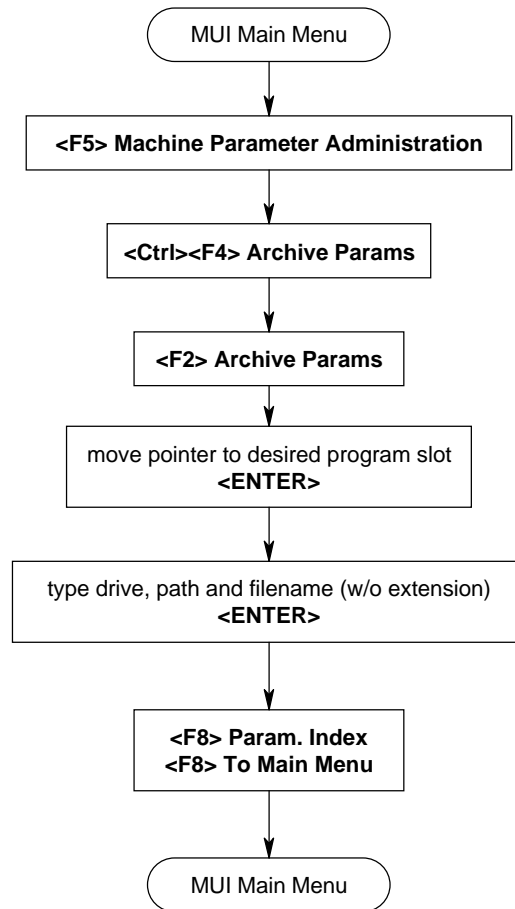


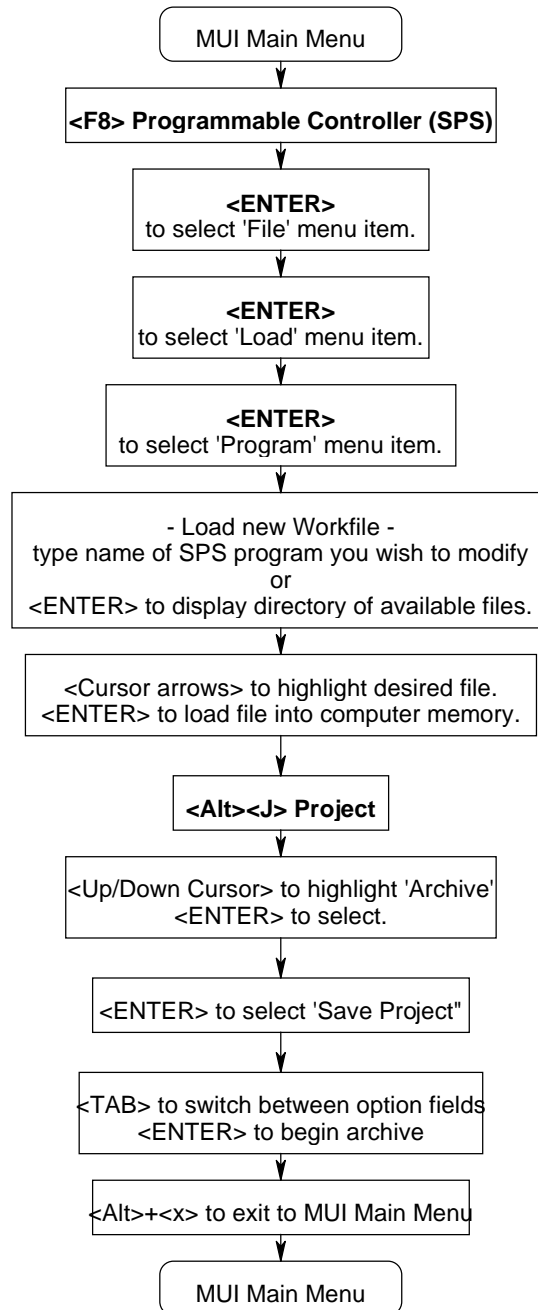
1.5 NC Program (Including Tool Setup List) - Archive

Figure 1-7 NC Program (Including Tool Setup List) - Archive

1.6 Machine Parameter (System, Process, Axis) - Archive

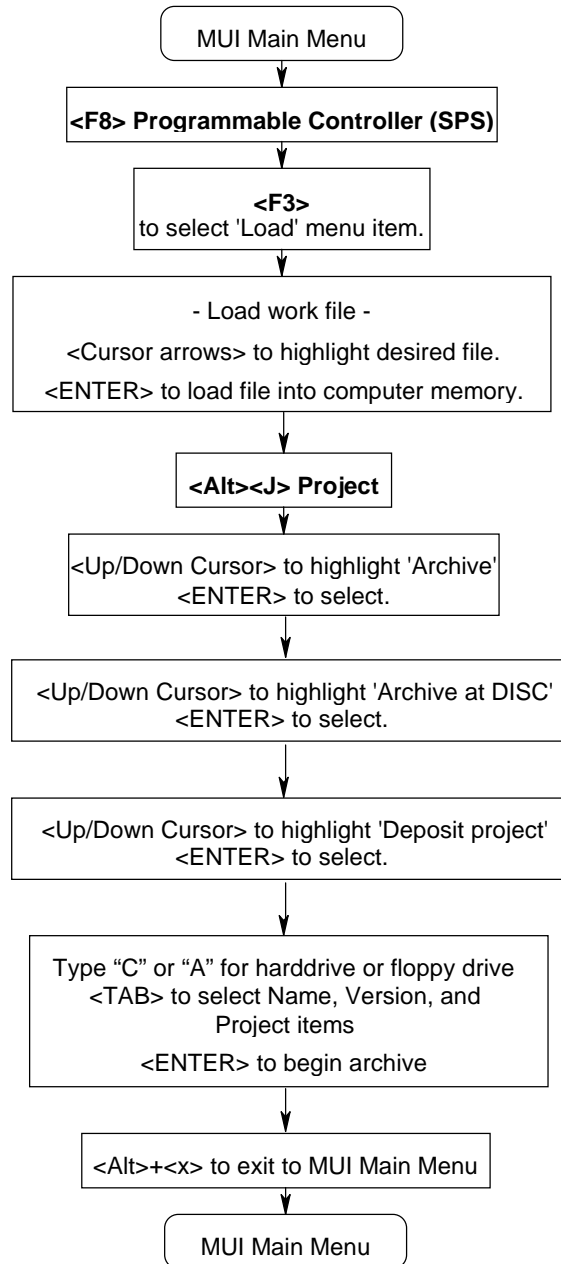
Figure 1-8 Machine Parameter (System, Process, Axis) - Archive

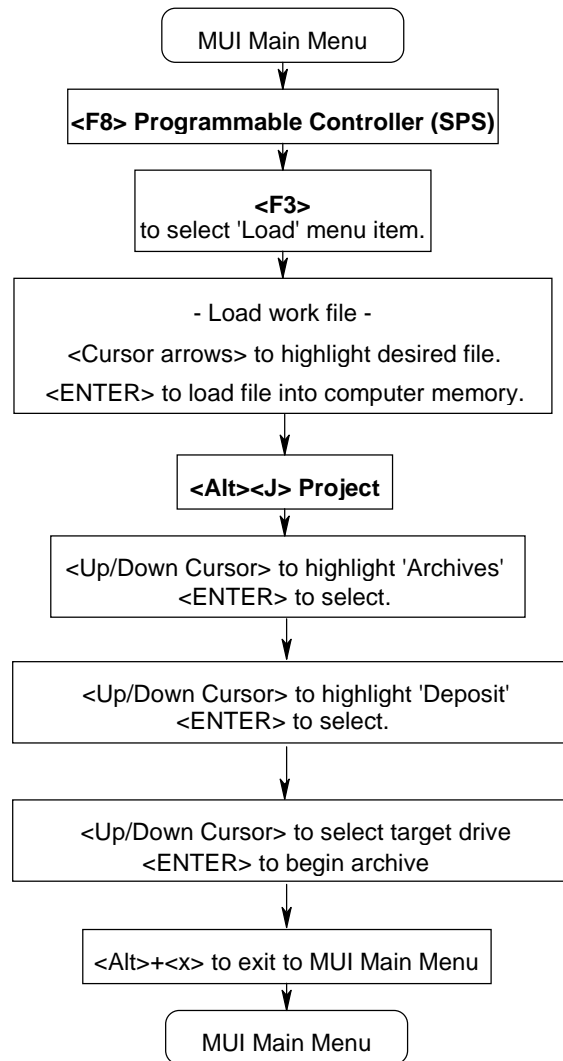


1.7 SPS (PLC) Program - Archive ^{(1),(2),(3)}**Figure 1-9 SPS (PLC) Program - Archive** ^{(1),(2),(3)}

1.8 SPS (PLC) Program - Archive ⁽⁴⁾

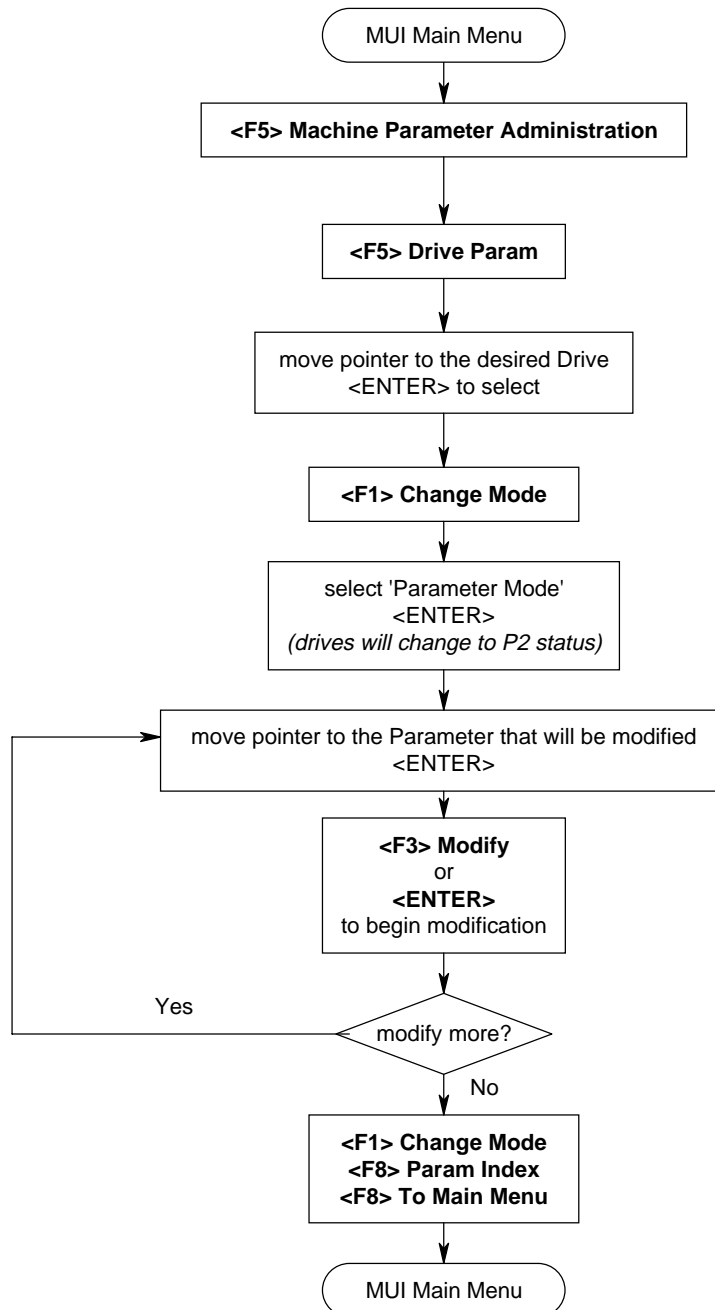
Figure 1-10 SPS (PLC) Program - Archive ⁽⁴⁾



1.9 SPS (PLC) Program - Archive ⁽⁵⁾**Figure 1-11 SPS (PLC) Program - Archive⁽⁵⁾**

1.10 Digital Drive Parameter (Servo, Spindle) - Editing

Figure 1-12 Digital Drive Parameter (Servo, Spindle) - Editing



1.11 Machine Parameter (System, Process, Axis) - Editing

Figure 1-13 Machine Parameter (System, Process, Axis) - Editing - (Sheet 1 of 2)

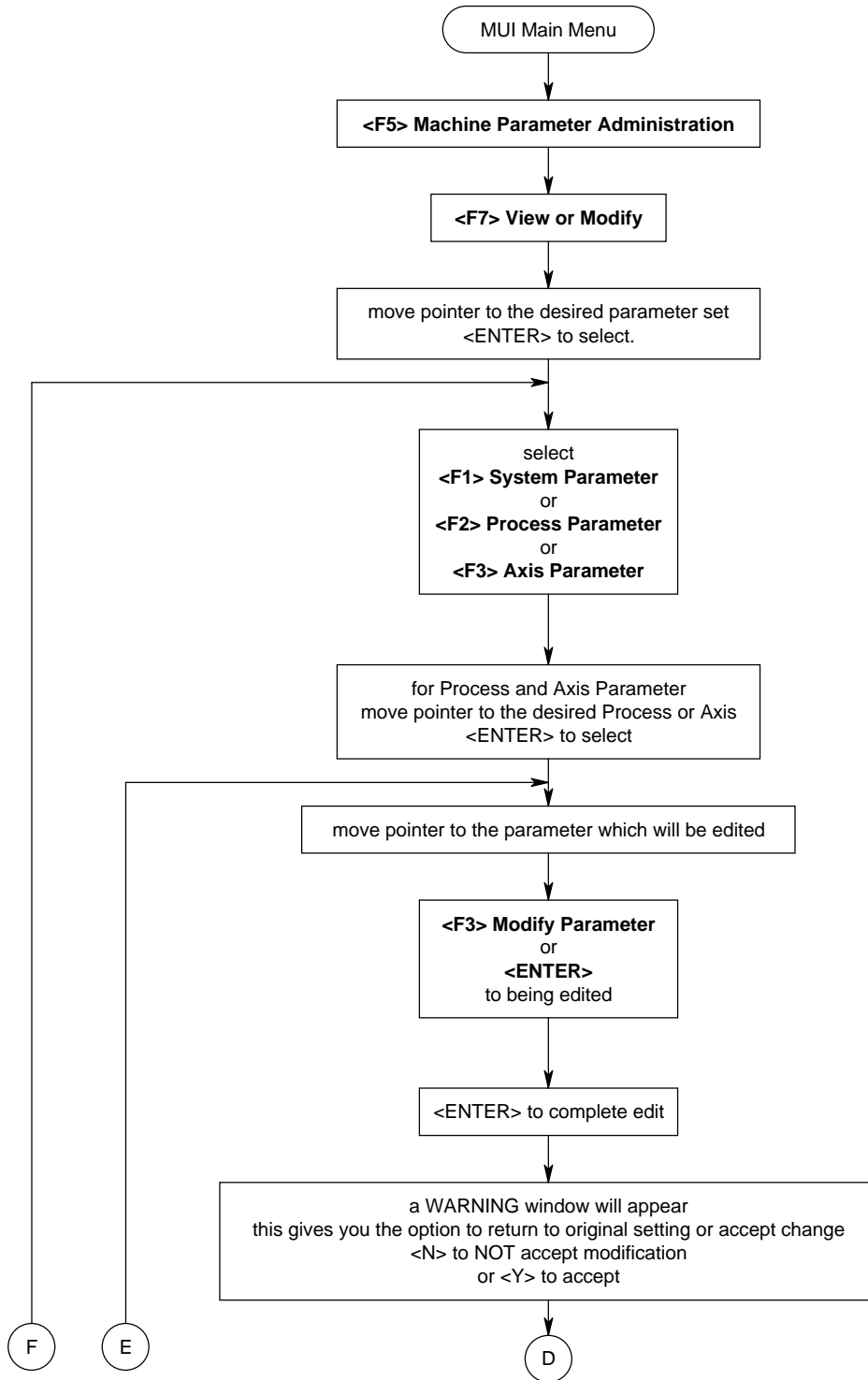
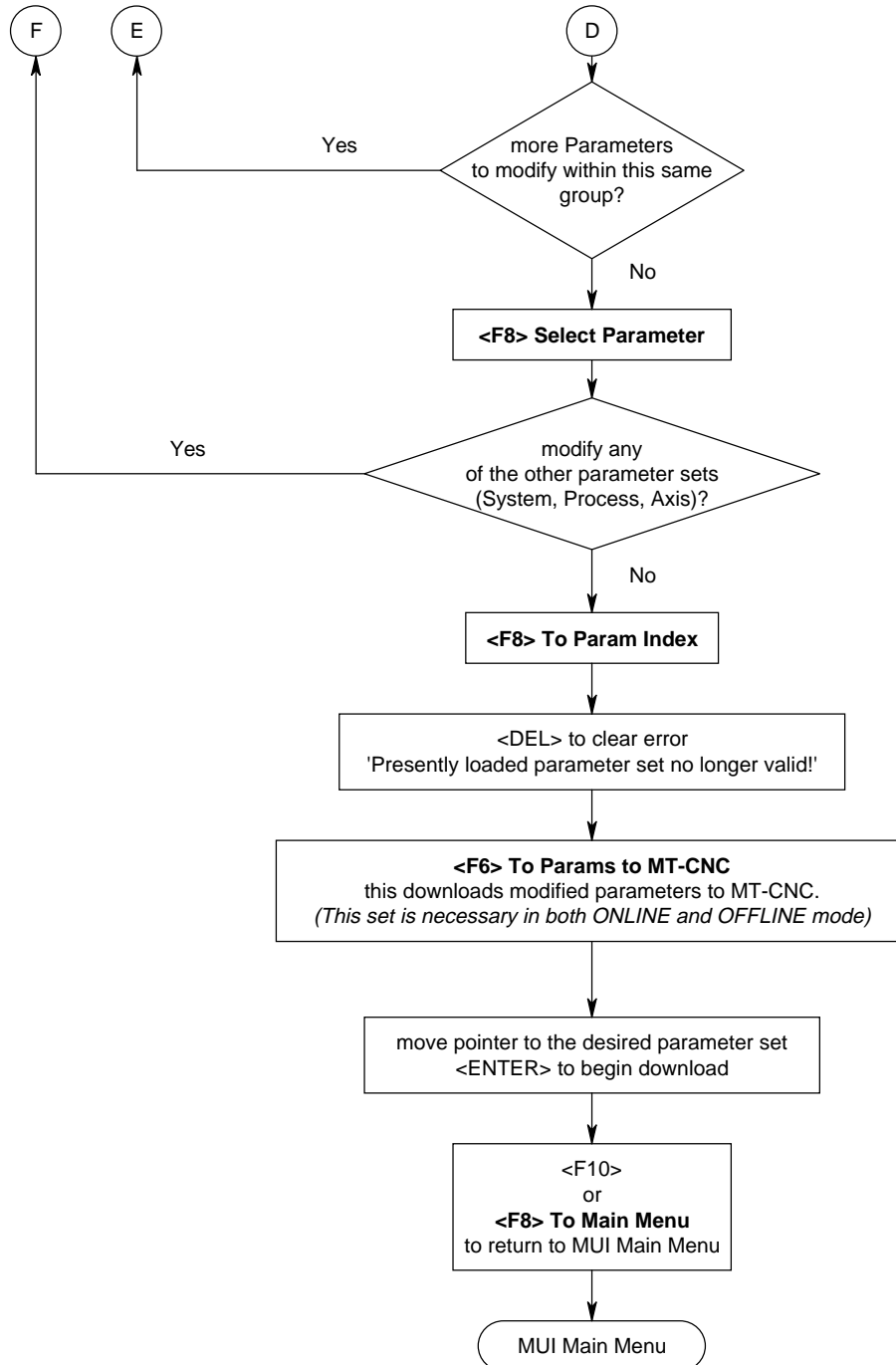
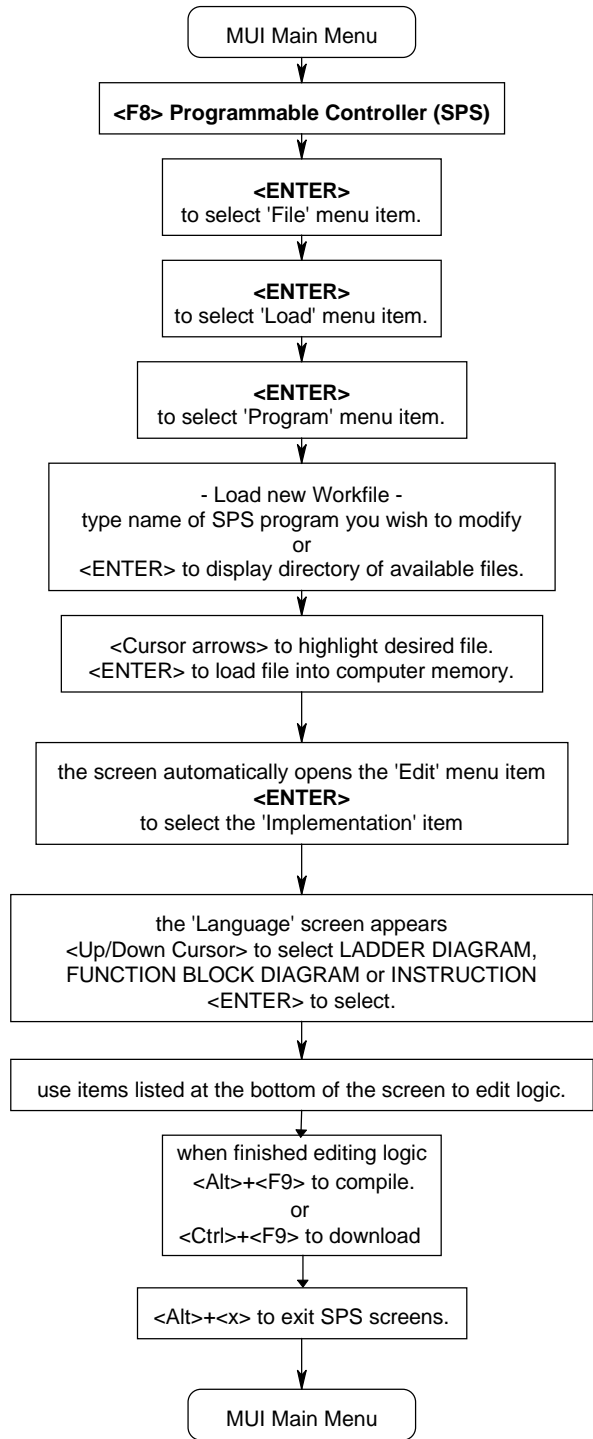


Figure 1-14 Machine Parameter (System, Process, Axis) - Editing - (Sheet 2 of 2)



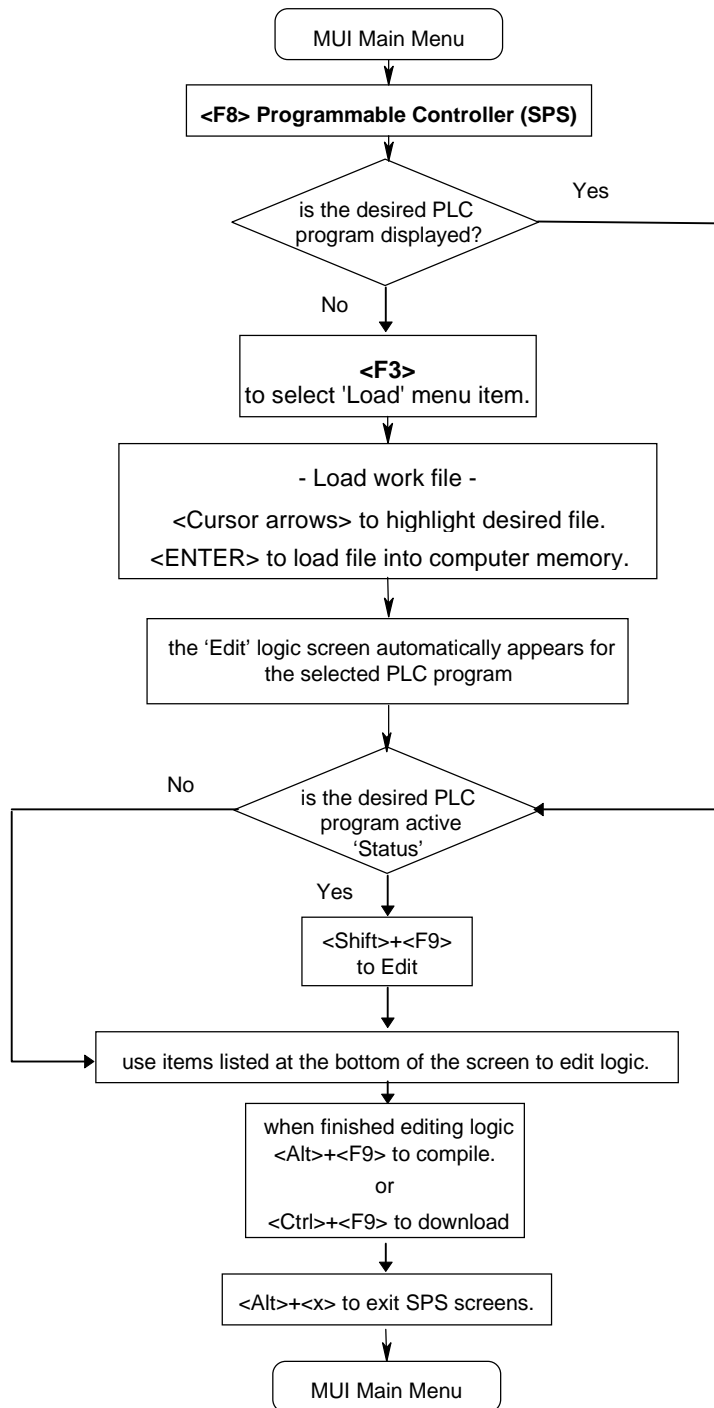
1.12 SPS (PLC) Editing ^{(1),(2),(3)}

Figure 1-15 SPS (PLC) Editing ^{(1),(2),(3)}

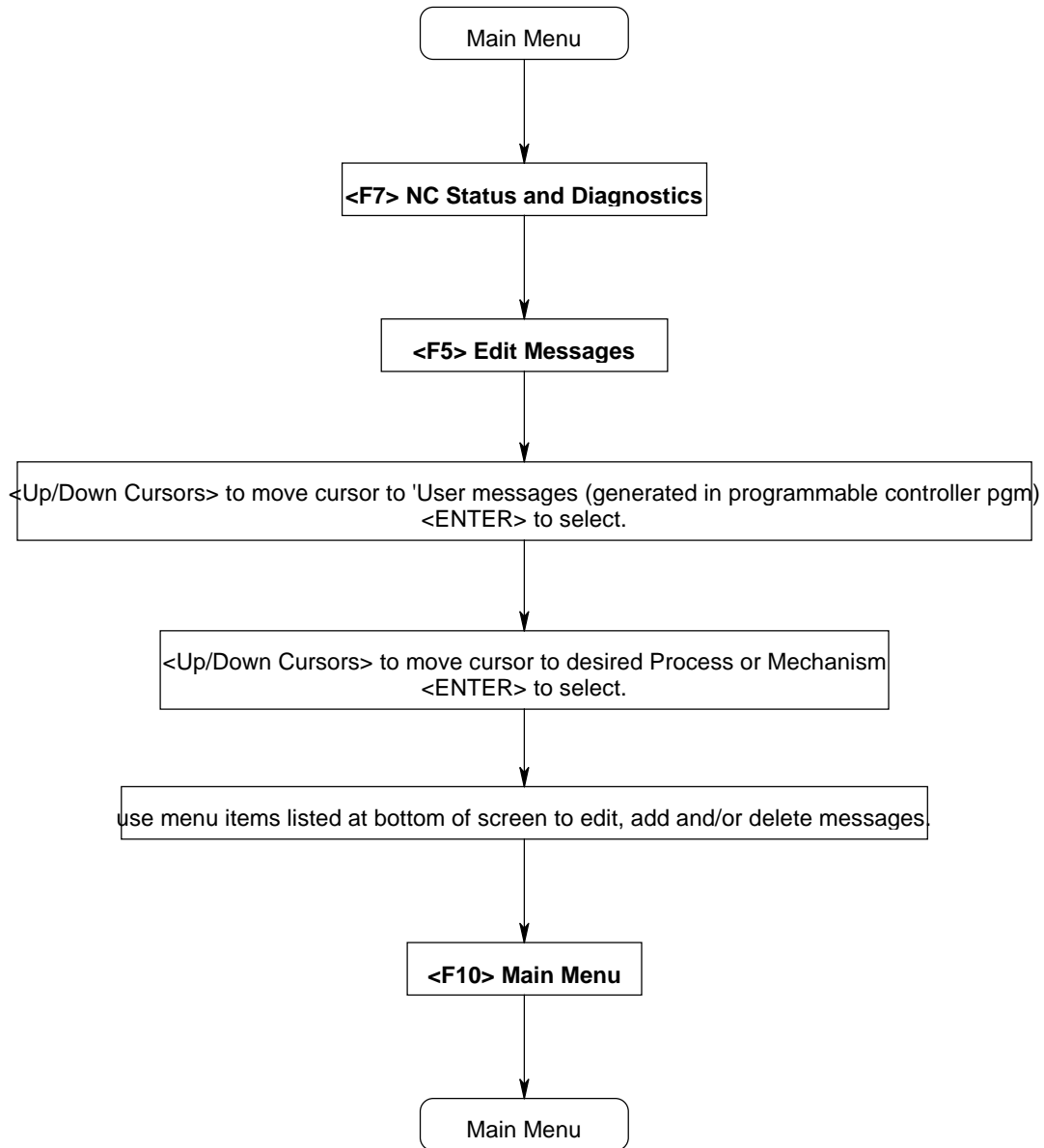


1.13 SPS (PLC) Editing ^{(4),(5)}

Figure 1-16 SPS (PLC) Editing ^{(4),(5)}

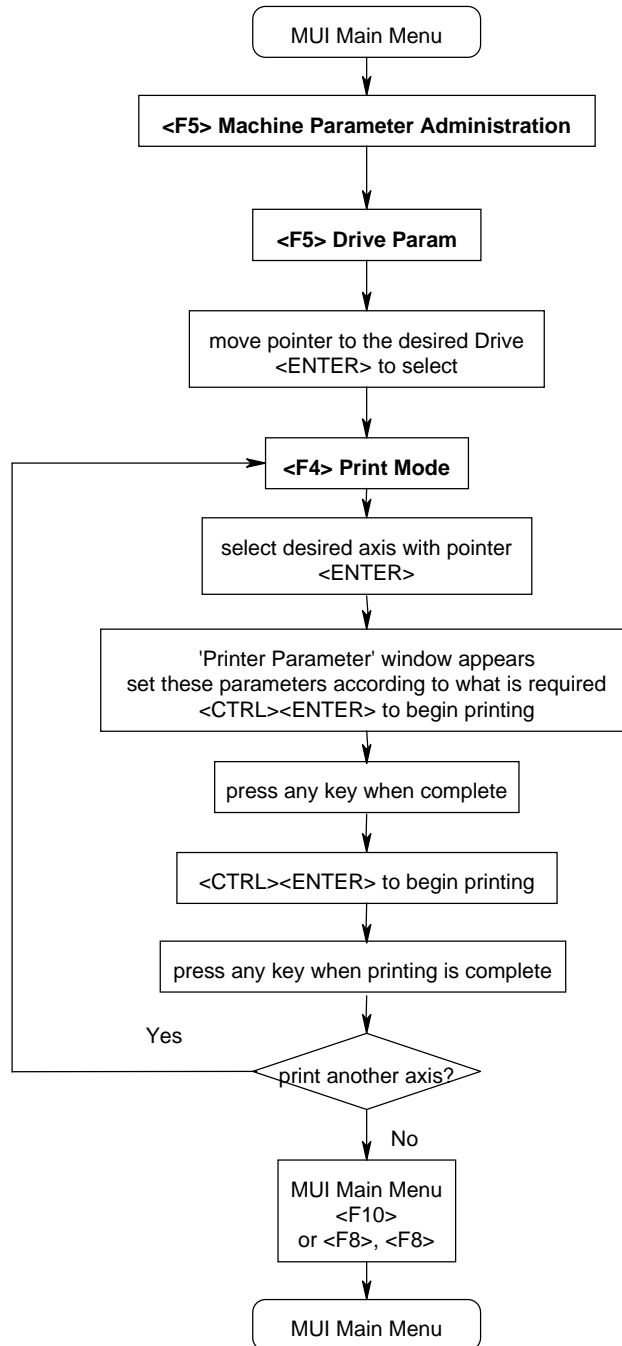


1.14 User Message - Editing

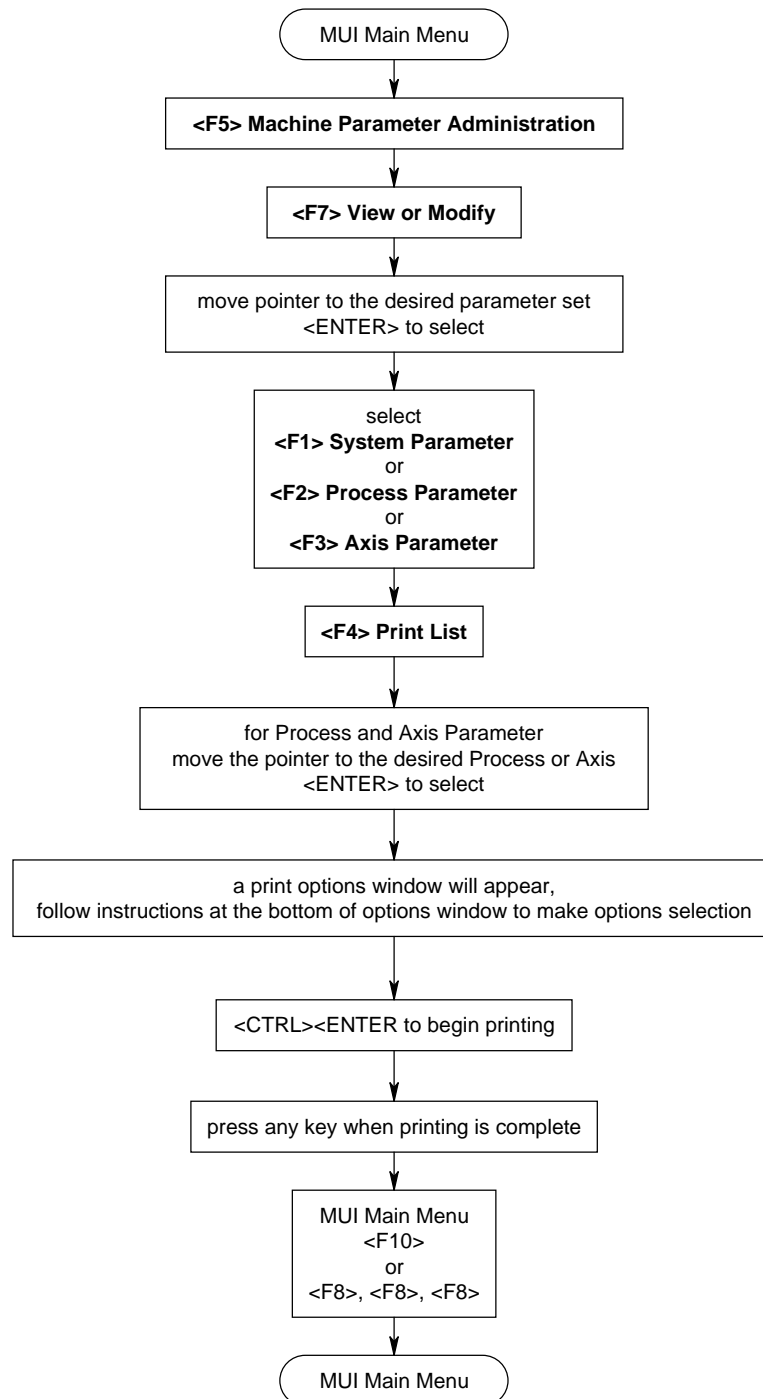
Figure 1-17 User Message - Editing -

1.15 Digital Drive Parameter (Servo, Spindle) - Print Out

Figure 1-18 Digital Drive Parameter (Servo, Spindle) - Print Out -

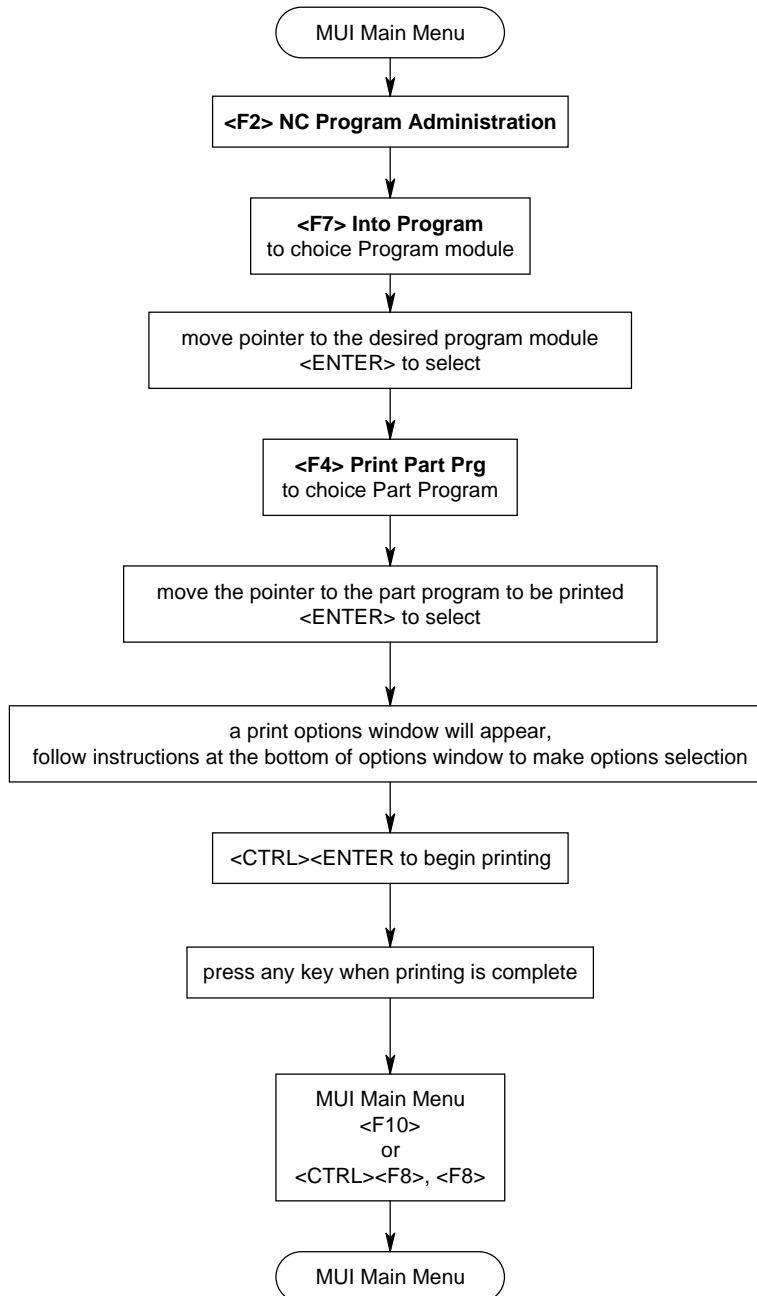


1.16 Machine Parameter (System, Process, Axis) - Print out

Figure 1-19 Machine Parameter (System, Process, Axis) - Print Out -

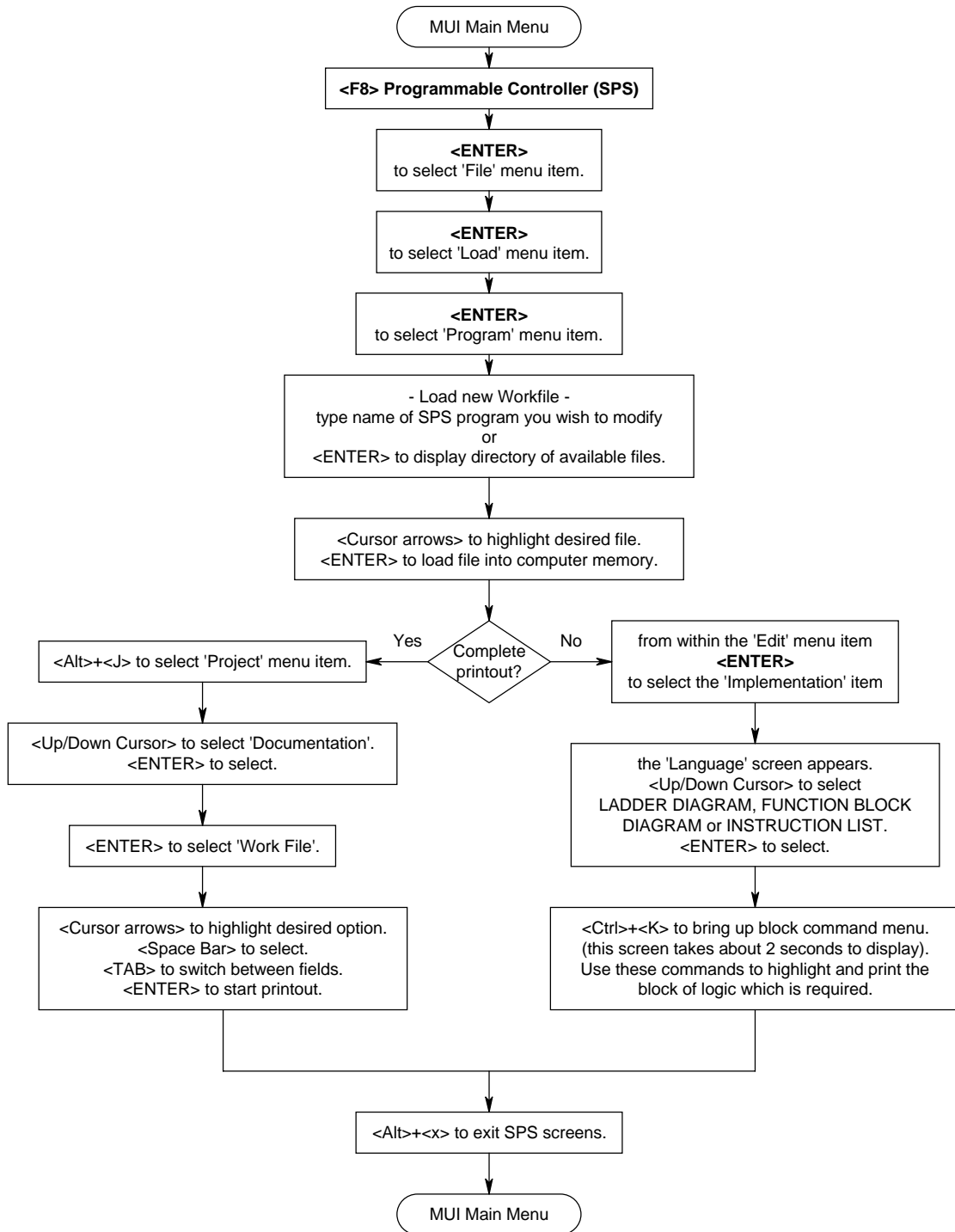
1.17 NC Program - Print Out

Figure 1-20 NC Program - Print Out -



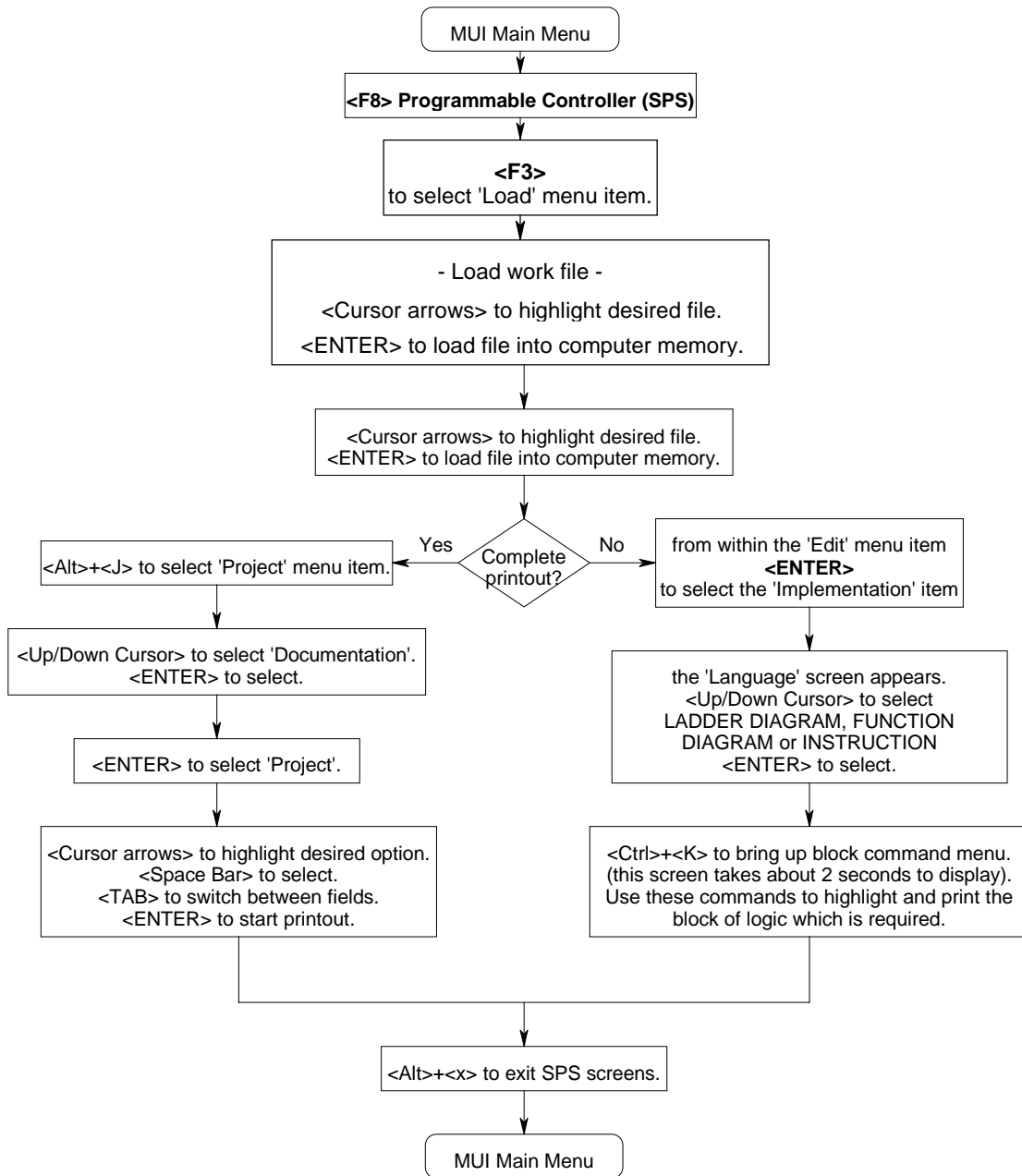
1.18 SPS Documentation - Complete and Partial Printouts ^{(1),(2),(3)}

Figure 1-21 SPS Documentation - Complete and Partial Printouts ^{(1),(2),(3)}



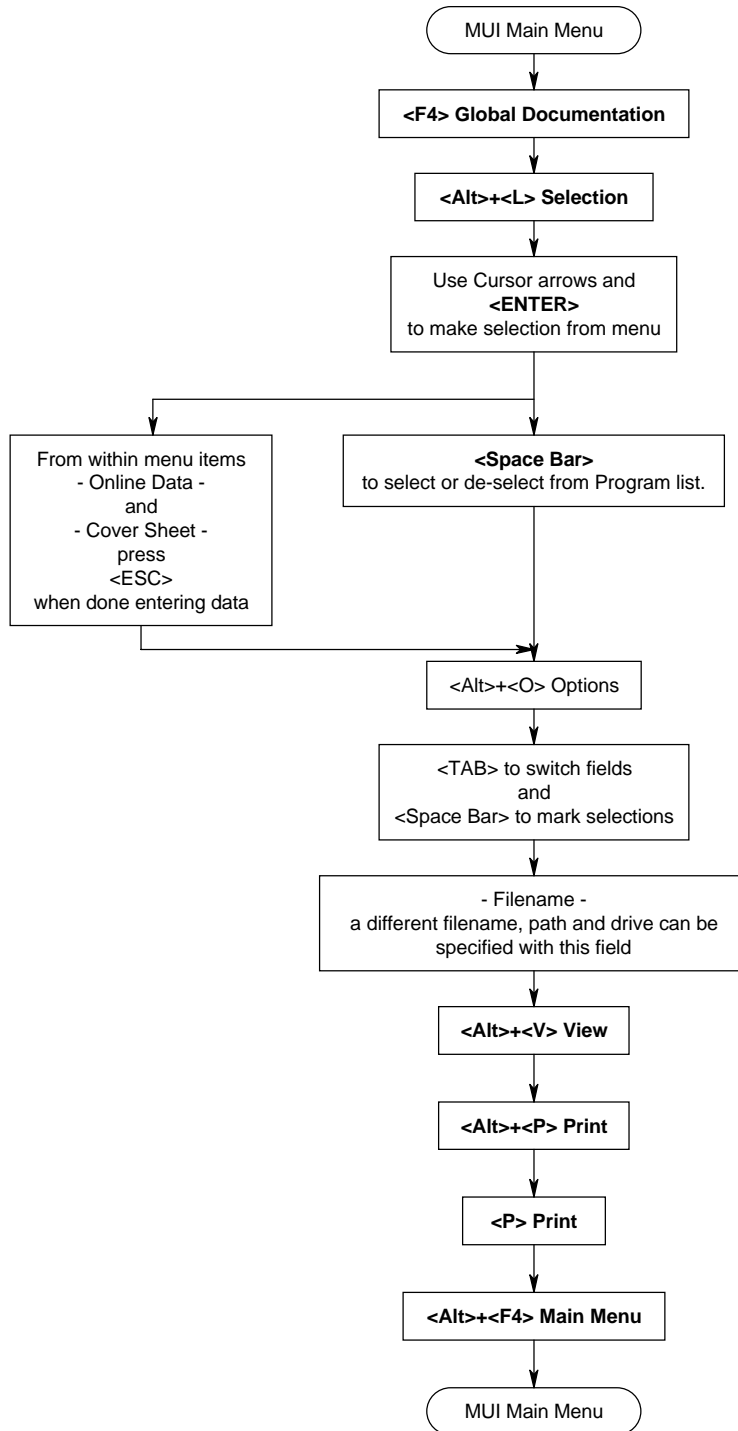
1.19 SPS Documentation - Complete and Partial Printouts

Figure 1-22 SPS Documentation - Complete and Partial Printouts ^{(4),(5)}



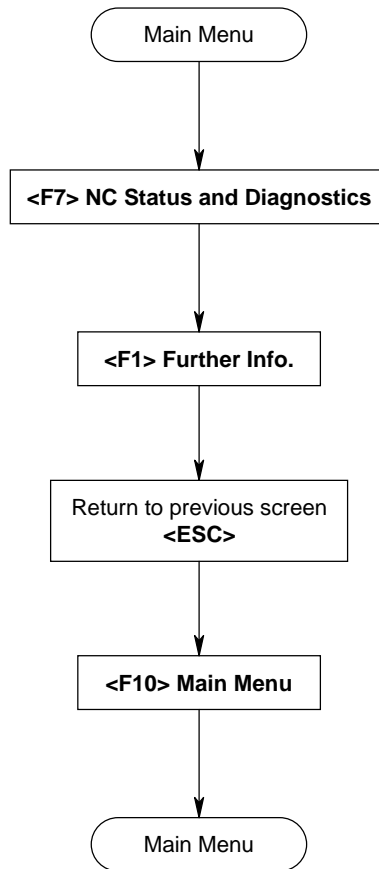
1.20 Global Documentation - Printing User Programs

Figure 1-23 Global Documentation - Printing User Programs -

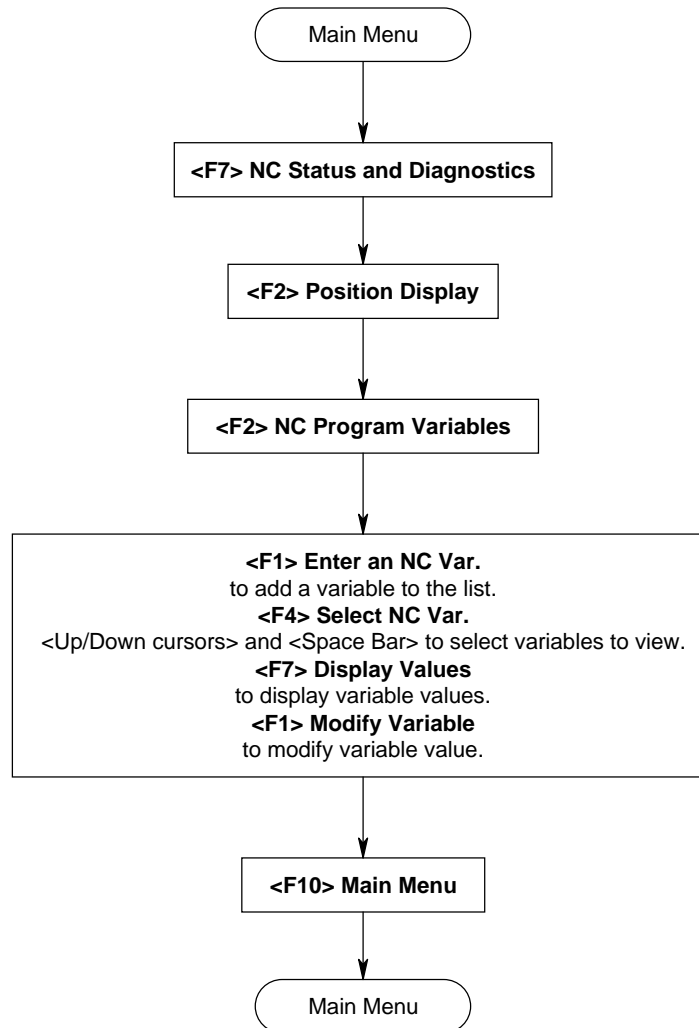


1.21 Further Information (Help) - NC Status and Diagnostics

Figure 1-24 Further Information (Help) - NC Status and Diagnostics -

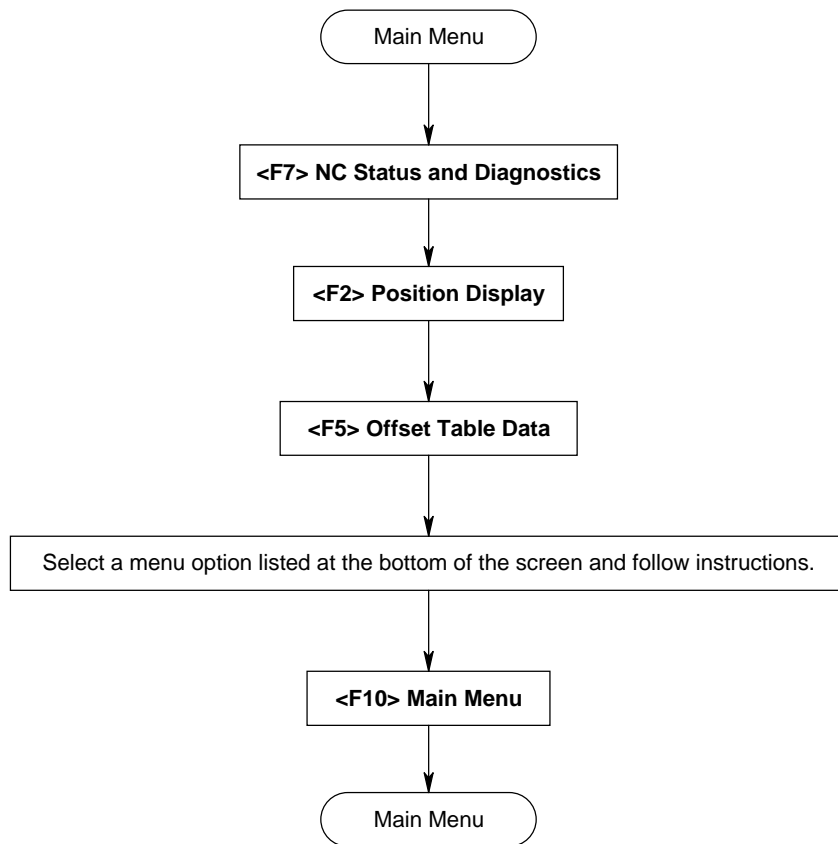


1.22 NC Program Variables

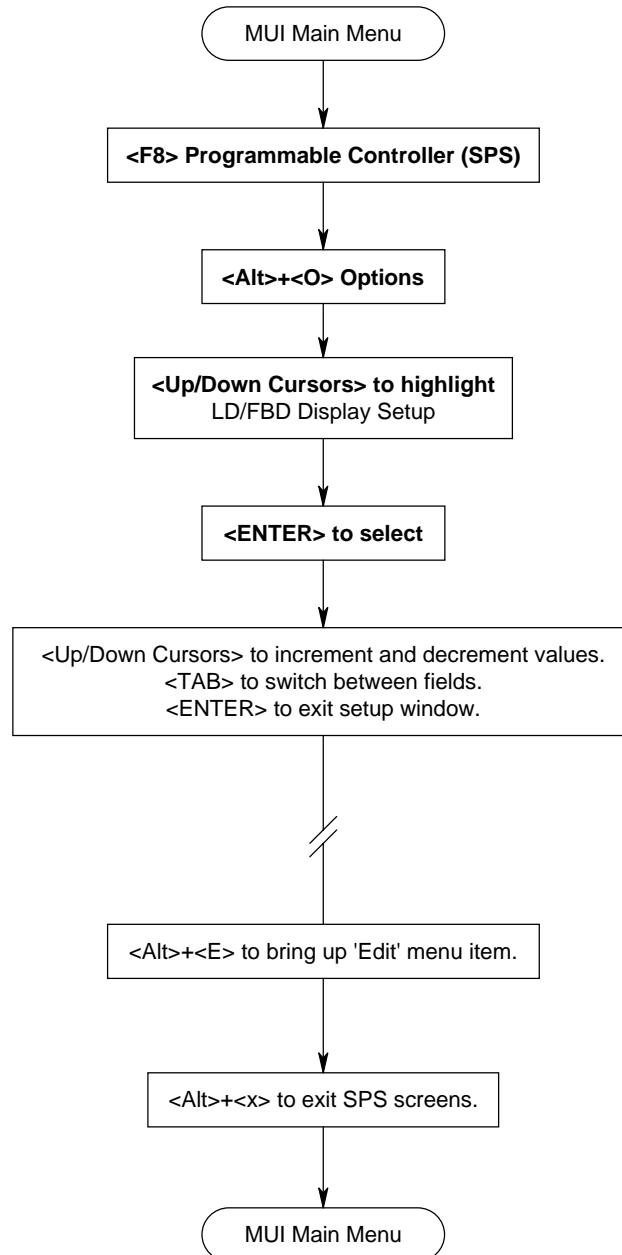
Figure 1-25 NC Program Variables

1.23 Offset Table Data

Figure 1-26 Offset Table Data

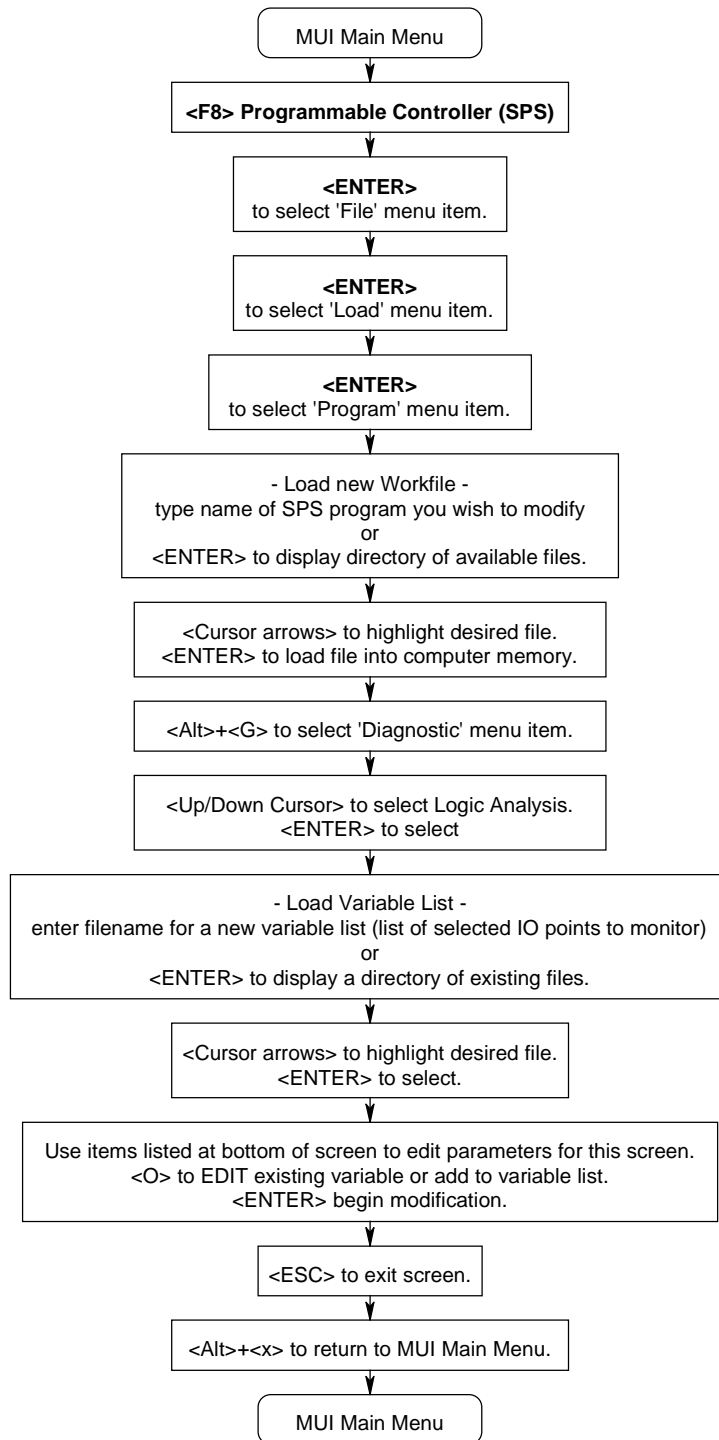


1.24 SPS Logic Display Setup

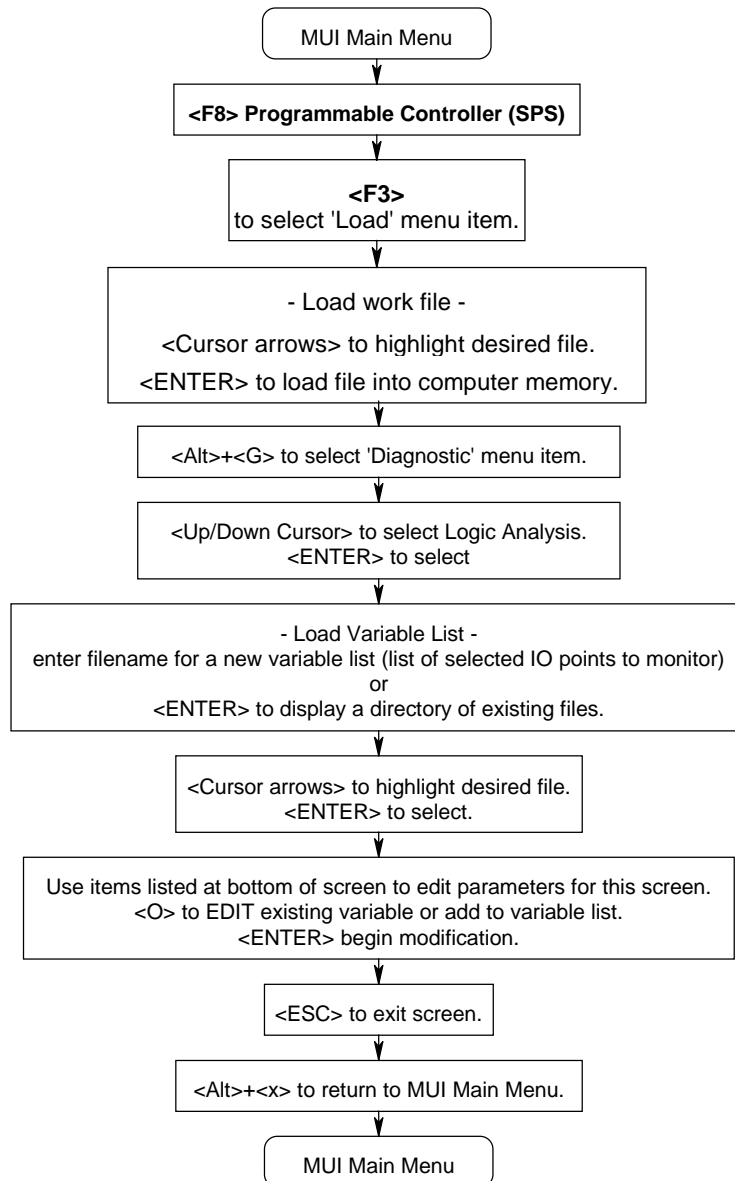
Figure 1-27 Logic Display Setup

1.25 SPS Logic Analysis ^{(1),(2),(3)}

Figure 1-28 SPS Logic Analysis ^{(1),(2),(3)}

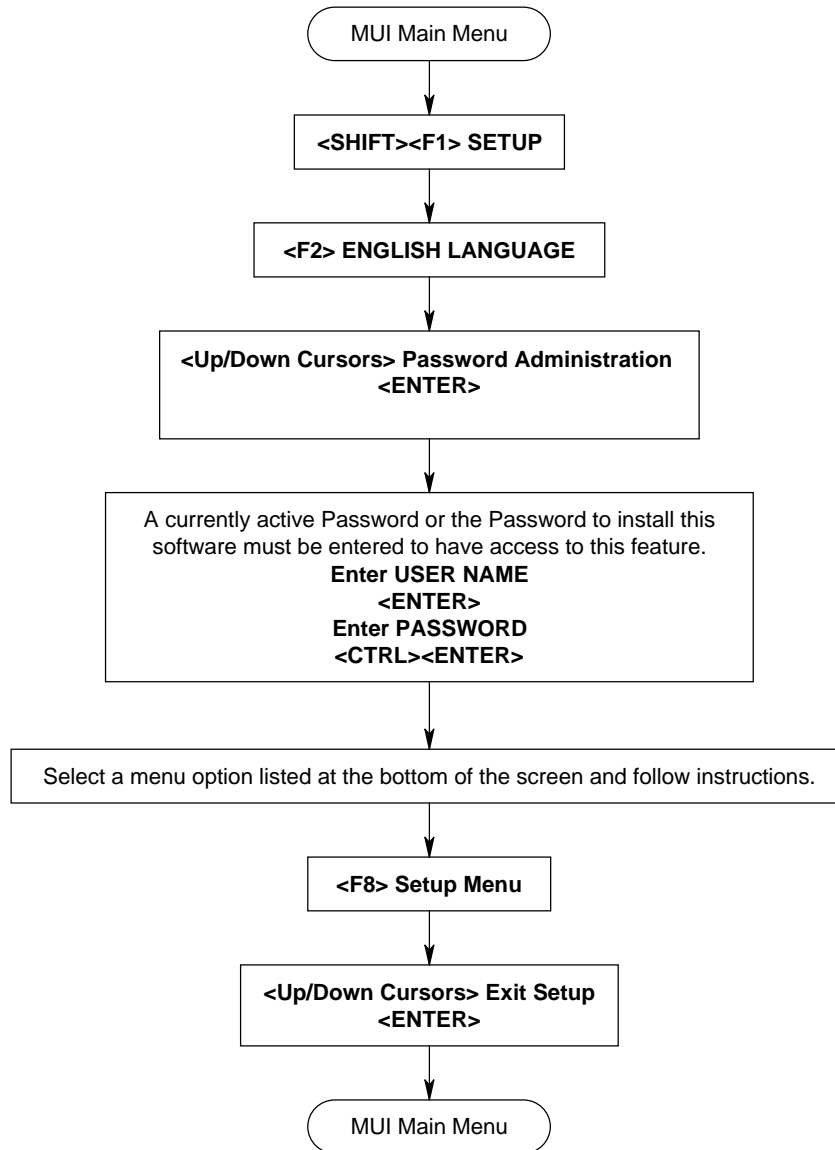


1.26 SPS Logic Analysis

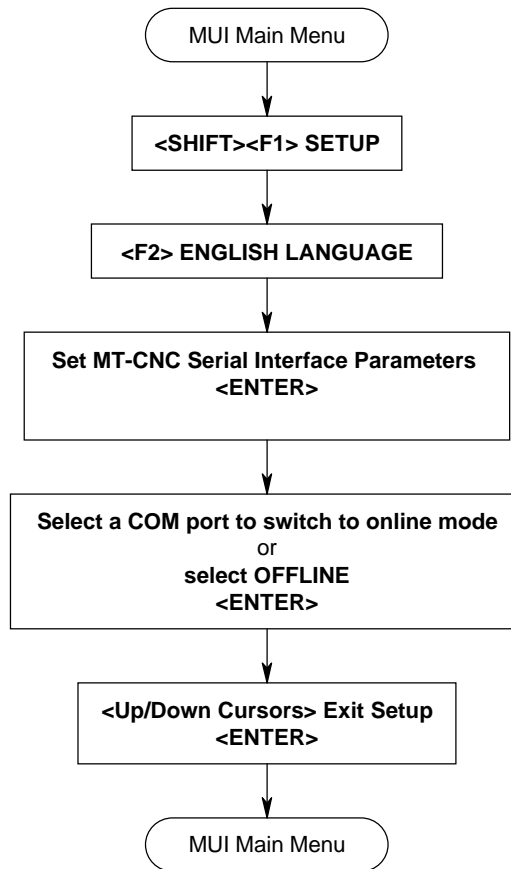
Figure 1-29 SPS Logic Analysis ^{(4),(5)}

1.27 Password Setup (for MT-CNC MUI Software)

Figure 1-30 Password Setup (for MT-CNC MUI Software)

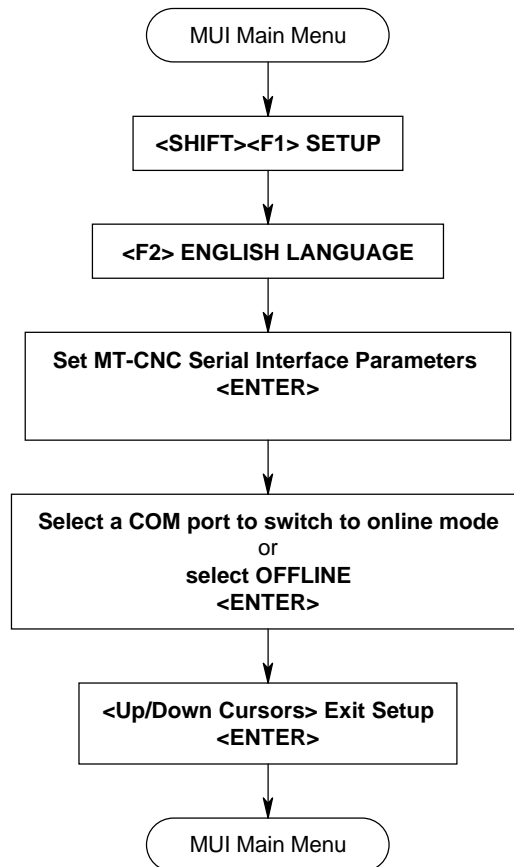


1.28 Selection of Online / Offline Mode (for MUI Software)

Figure 1-31 Selection of Online / Offline Mode (for MUI Software)

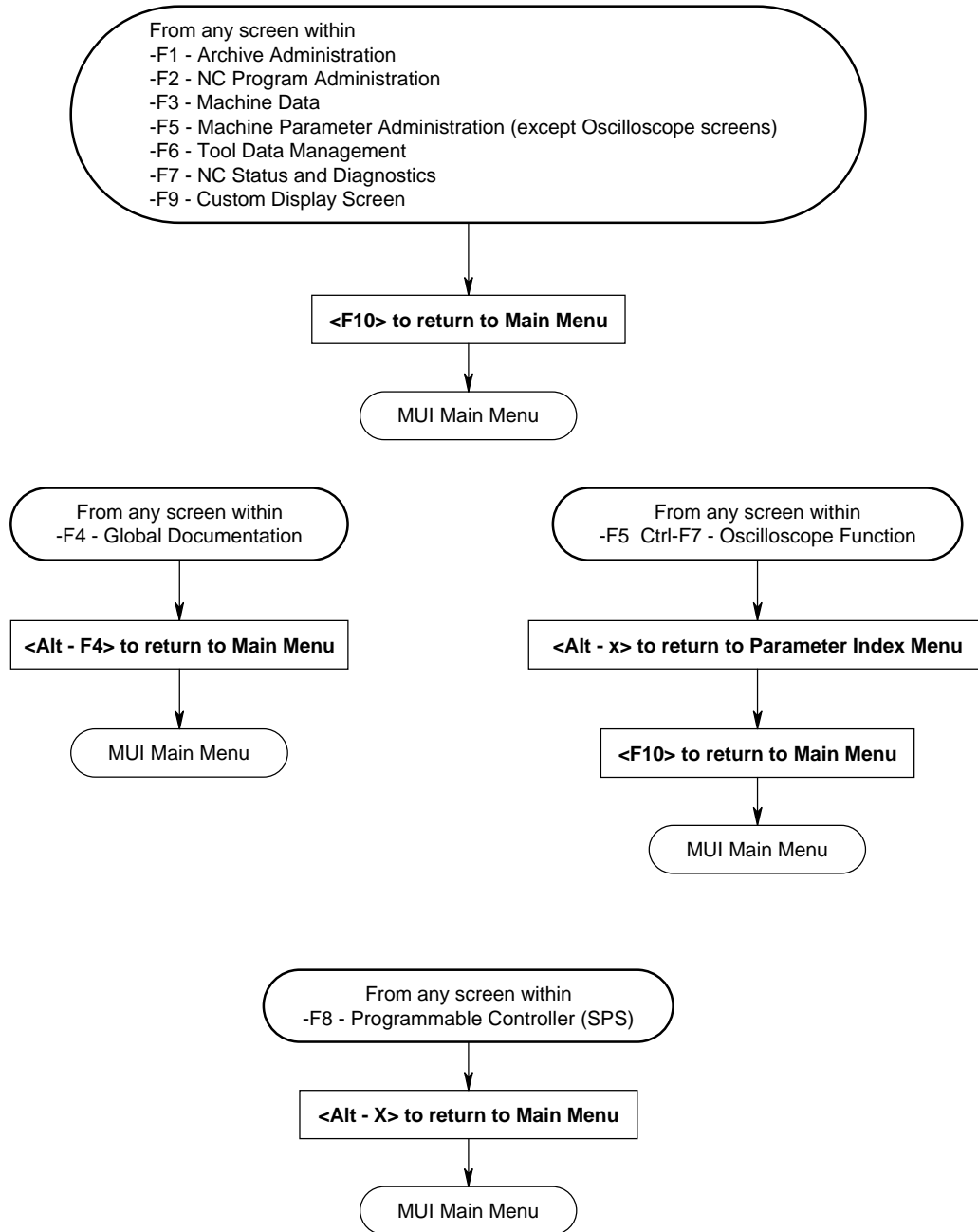
1.29 Selection of Online / Offline Mode (for MUI Software)

Figure 1-32 Selection of Online/Offline Mode (for MUI Software)



1.30 Exiting (from any screen back to the MUI Main Menu)

Figure 1-33 Exiting (from any screen back to MUI Main Menu)



1.31 Oscilloscope - Setup, Record, View, Save

Figure 1-34 Oscilloscope - Setup, Record, View, Save - (Sheet 1 of 2)

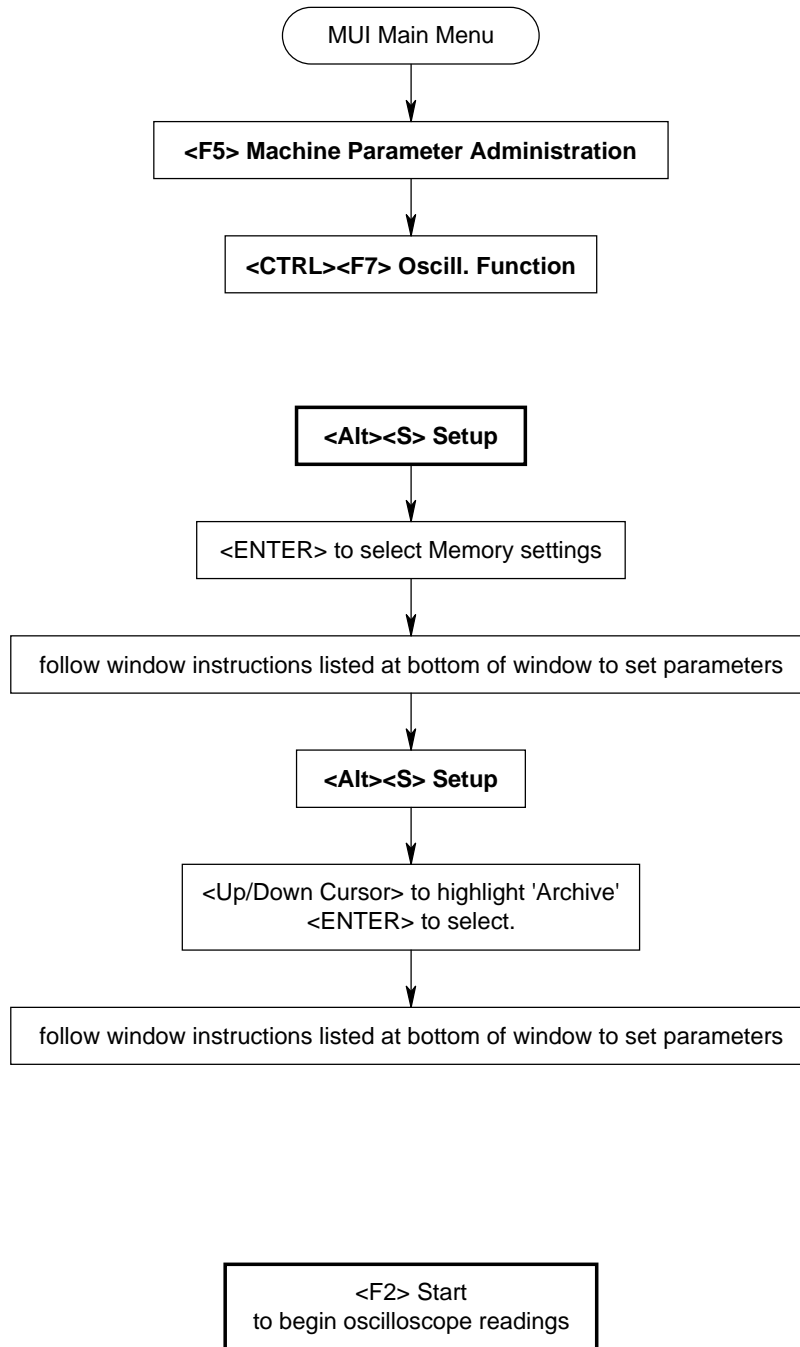
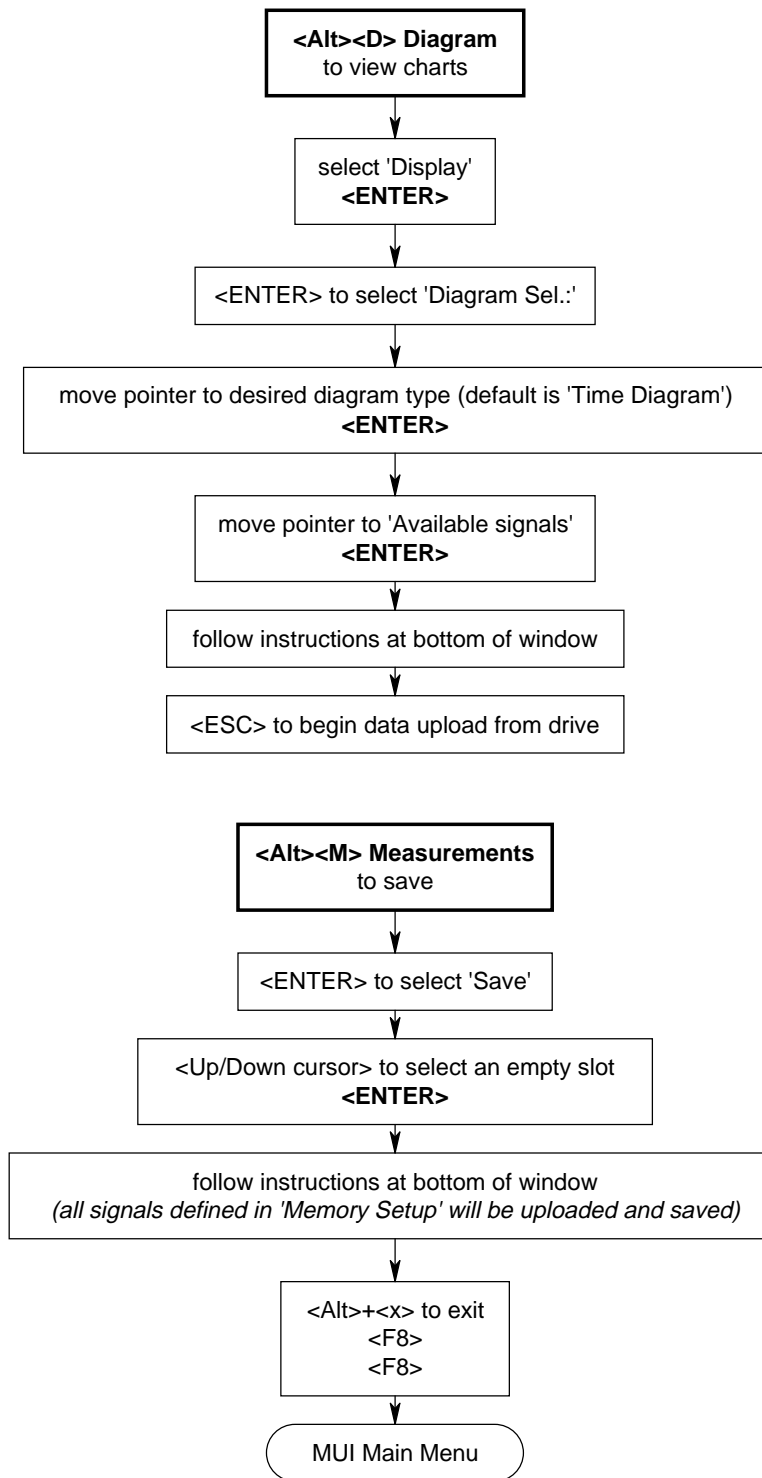
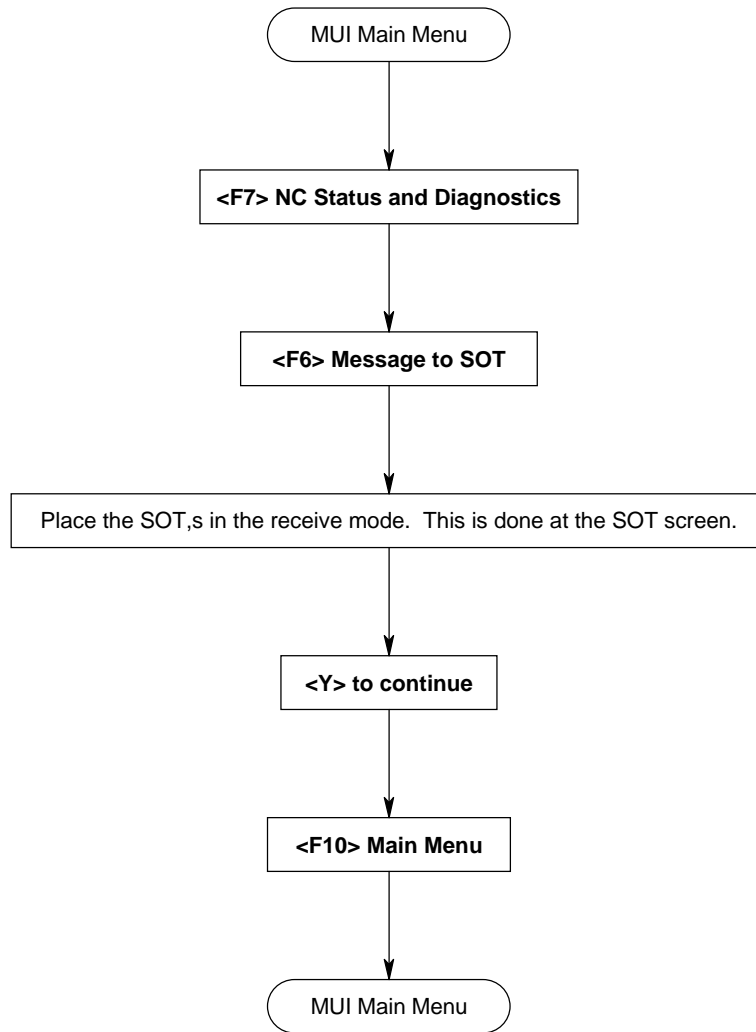


Figure 1-35 Oscilloscope - Setup, Record, View, Save - (Sheet 2 of 2)

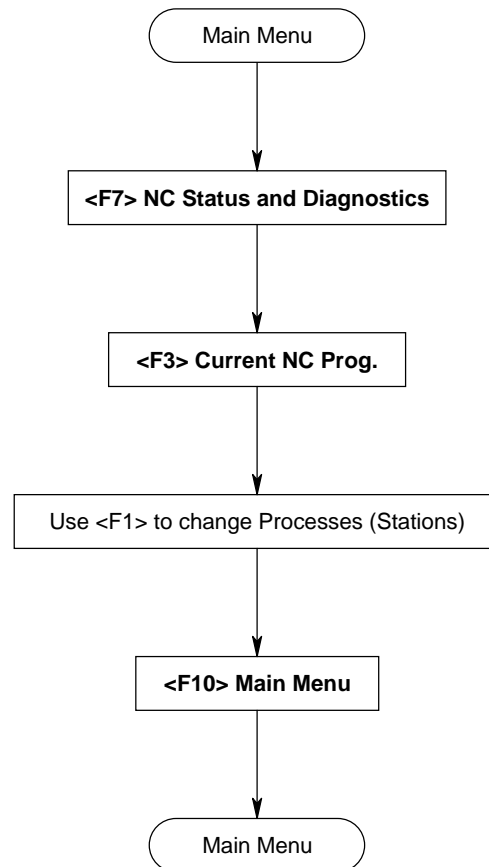


1.32 SOT Message - Downloading

Figure 1-36 SOT Message - Downloading -



1.33 Current NC Program - Viewing

Figure 1-37 Current NC Program - Viewing -

2. GUI FLOW CHARTS

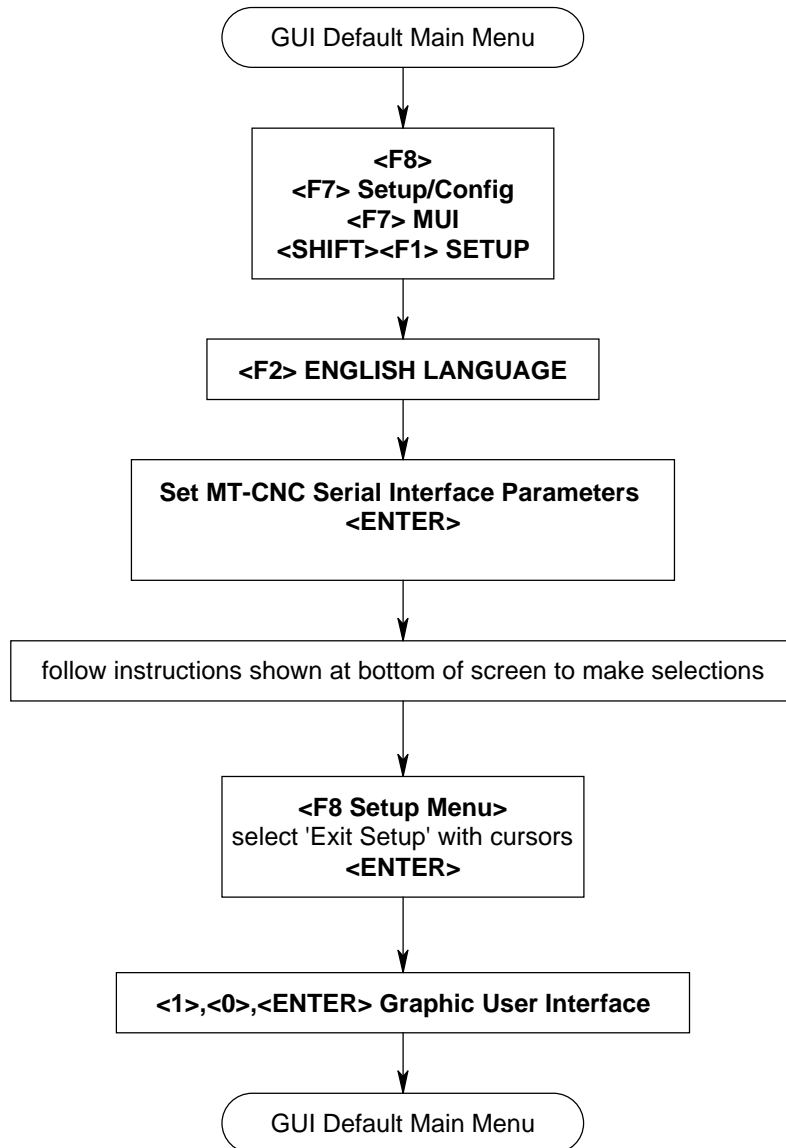
This chapter contains flowcharts to perform commonly used functions with the standard Indramat softkey menu of the MT-CNC Graphical User Interface Software. Since the user or OEM can customized the softkey levels in the GUI, the flowcharts listed here may not or only partially be compatible with the used GUI. In this case refer to the flowchart provided by the user or OEM.

2.1 Setup, GUI (Indramat default)

2.1.1 Online / Off-line Mode, GUI (Indramat default)

Figure 2-1: GUI Online/Off-line Mode

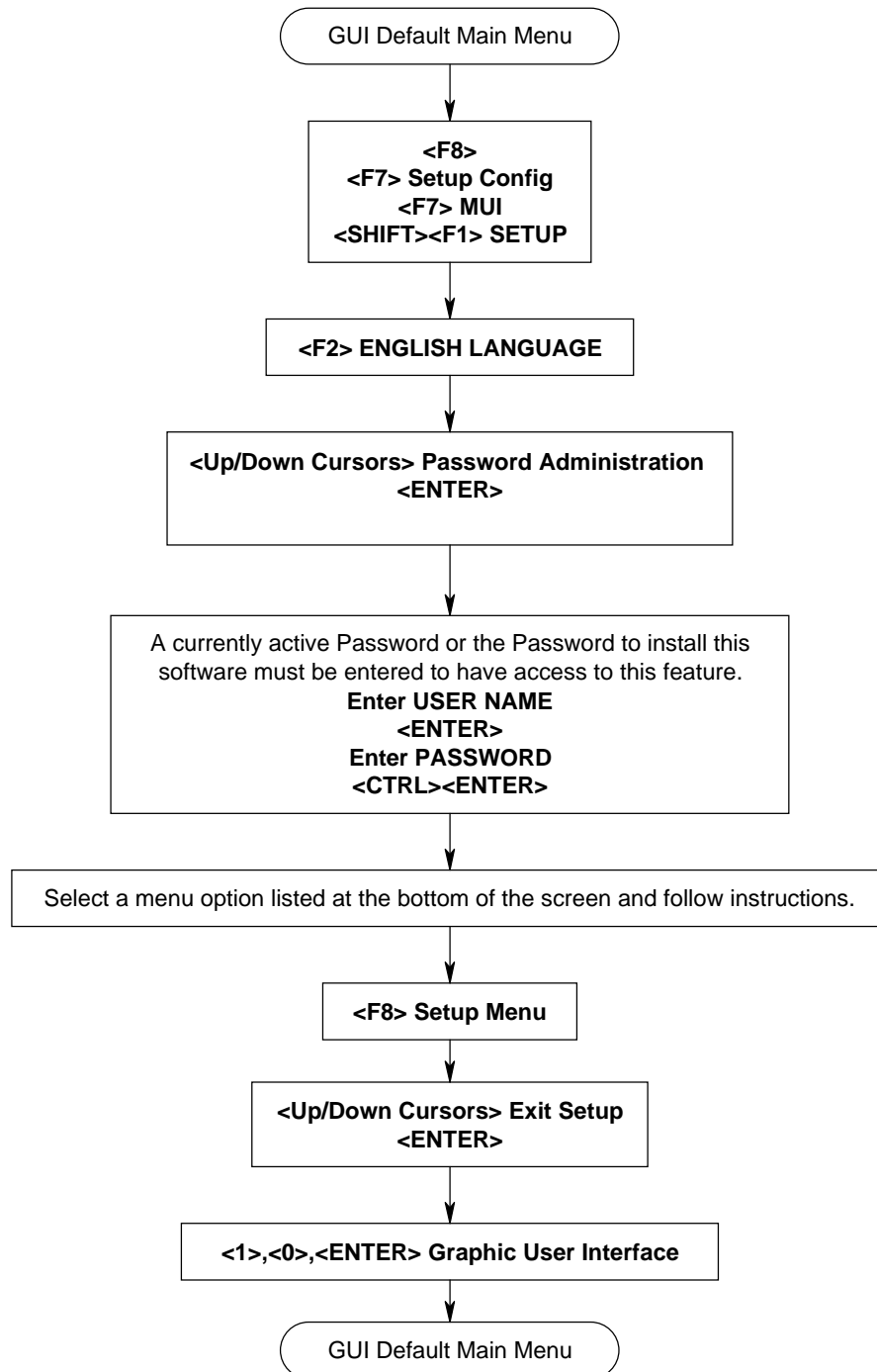
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.1.2 Password Setup, GUI (Indramat default)

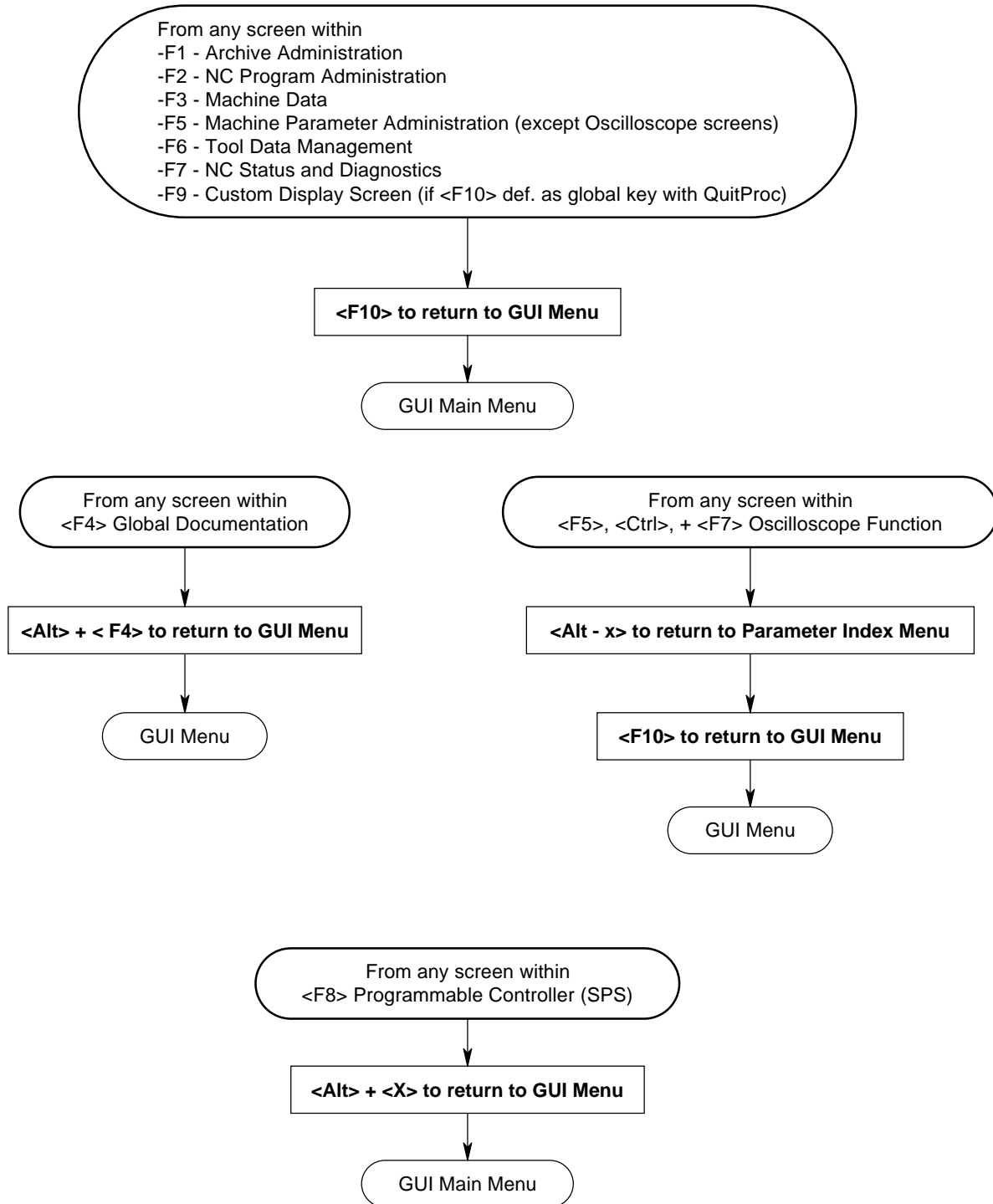
Figure 2-2 GUI Password Setup

(this chart is only valid for the default GUI screens supplied with the installation diskettes)

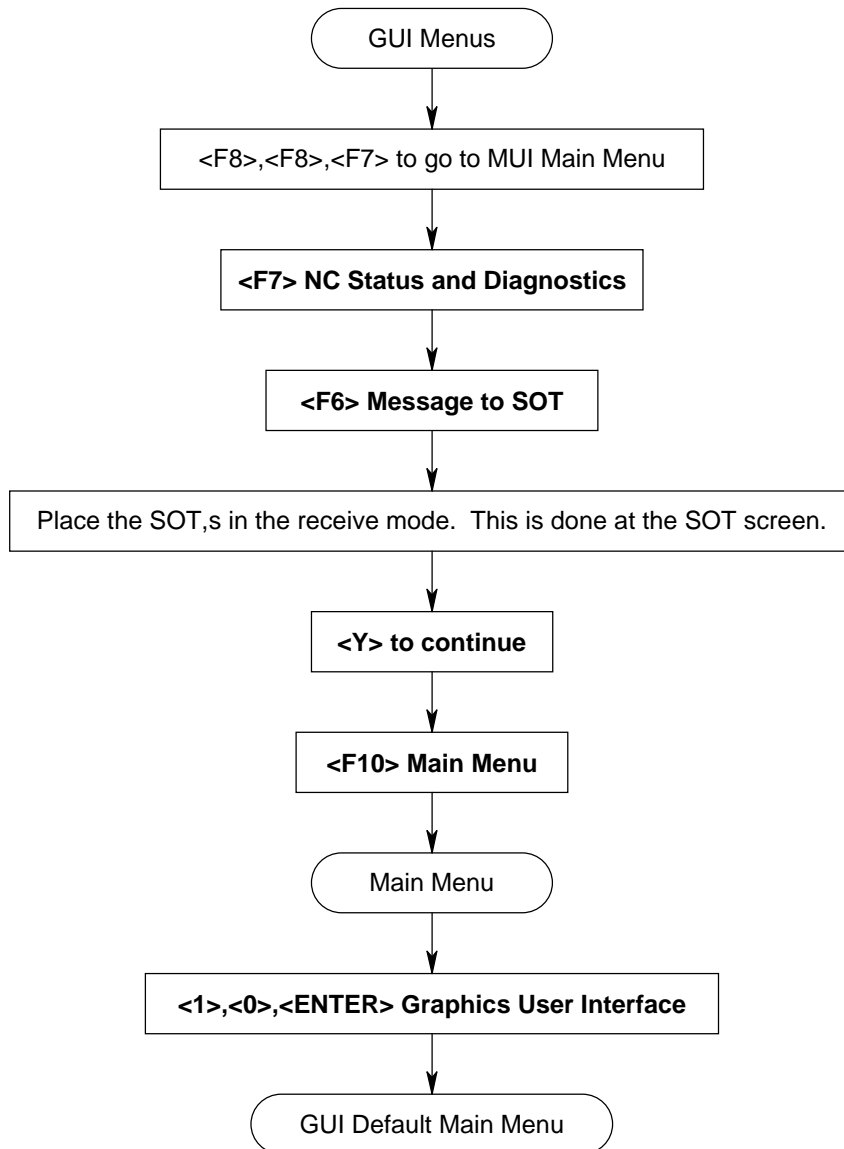


2.1.3 Exiting Menus, GUI (Indramat default)

Figure 2-3 GUI Exiting Menus - from any screen back to GUI Menu



2.1.4 Load SOT Messages, GUI (Indramat default)

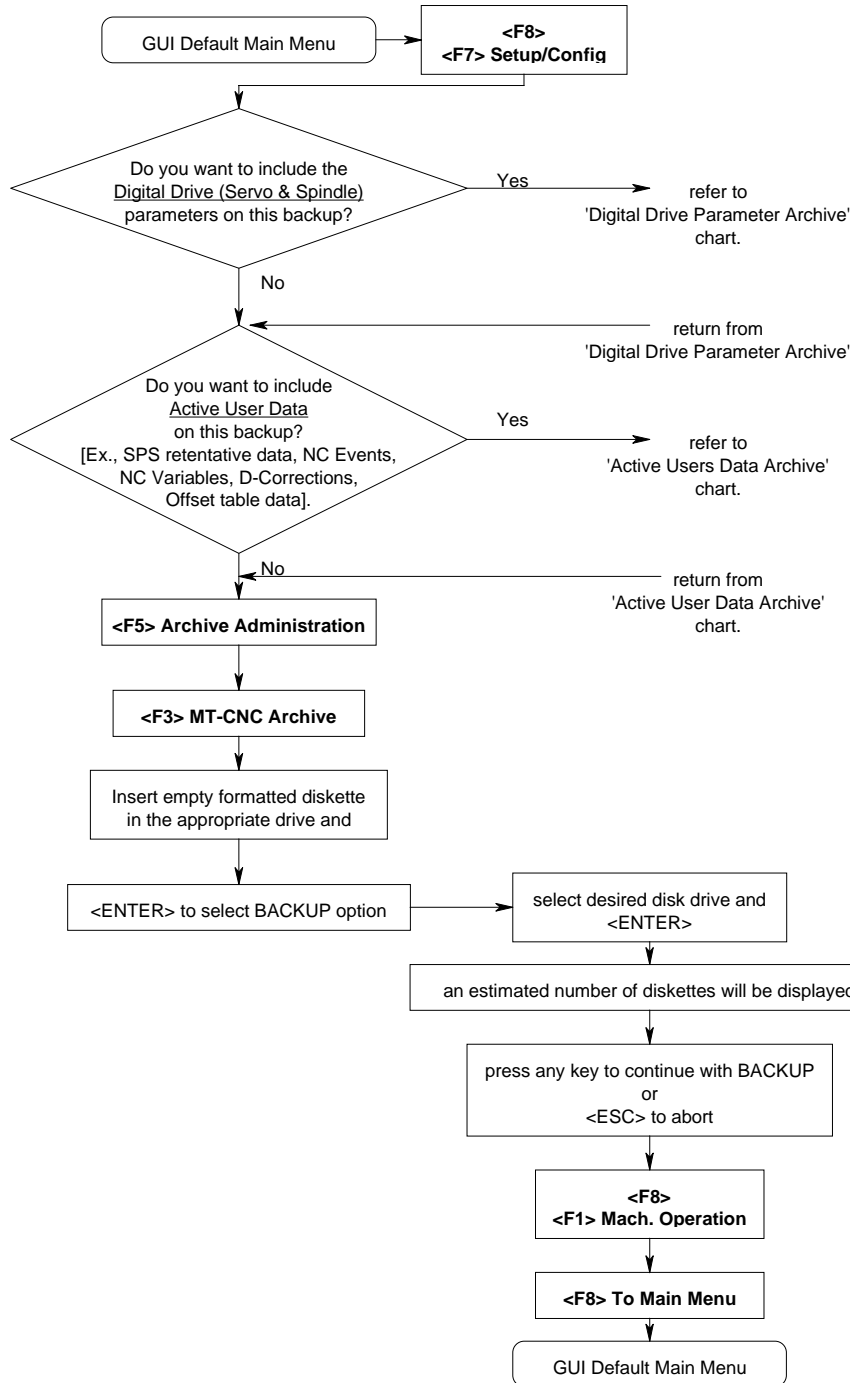
Figure 2-4 GUI Load SOT Messages

2.2 BACKUP, GUI (Indramat default)

2.2.1 Total MT-CNC User Programs & Data Backup, GUI (Indramat default)

Figure 2-5 GUI Total MT-CNC Users Programs & Data Backup

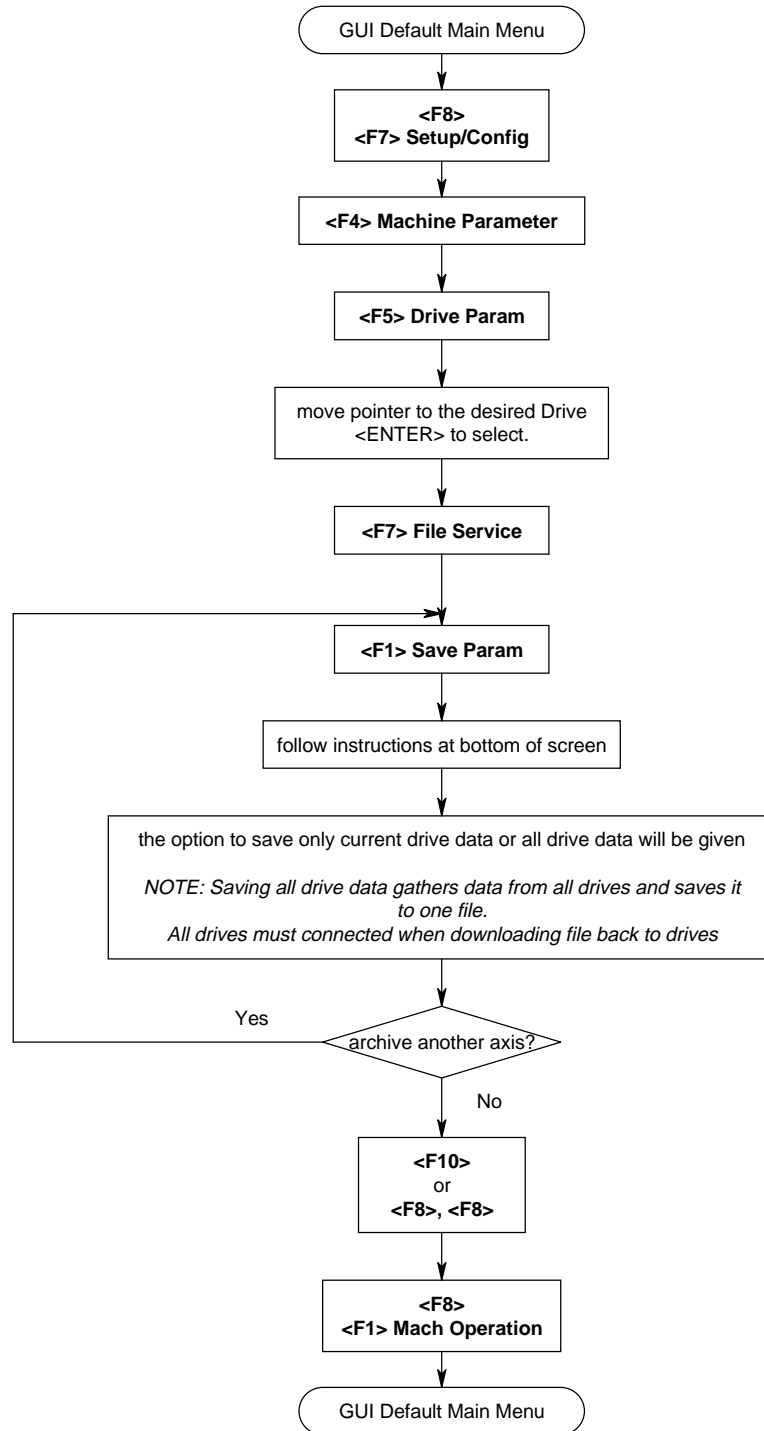
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.2.2 Digital Drive Parameter Archive, GUI (Indramat default)

Figure 2-6 GUI Digital Drive Parameter Archive

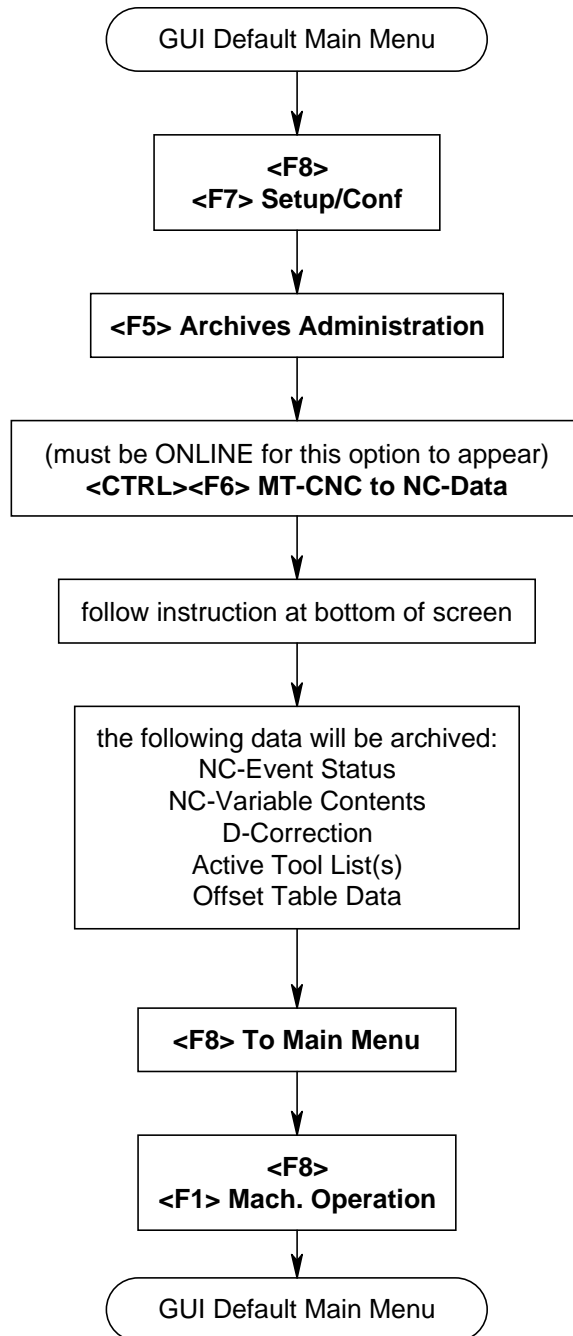
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.2.3 Active User Data Archive, GUI (Indramat default)

Figure 2-7 GUI Active User Data Archive

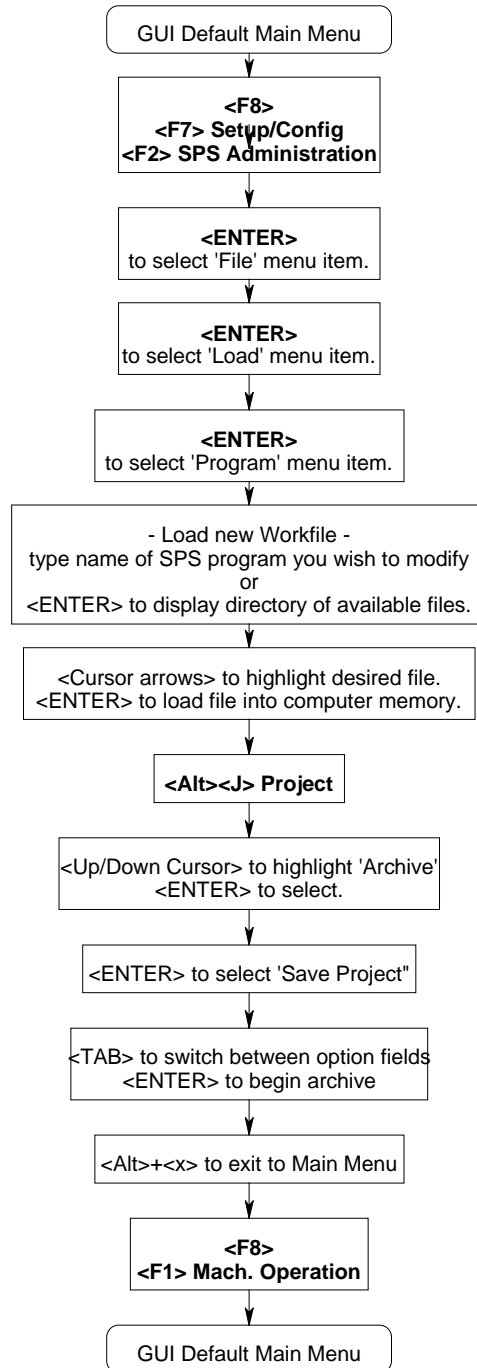
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.2.4 SPS (PLC) Project Archive, GUI ^{(1),(2),(3)} (Indramat default)

Figure 2-8 GUI SPS (PLC) Project Archive ^{(1),(2),(3)}

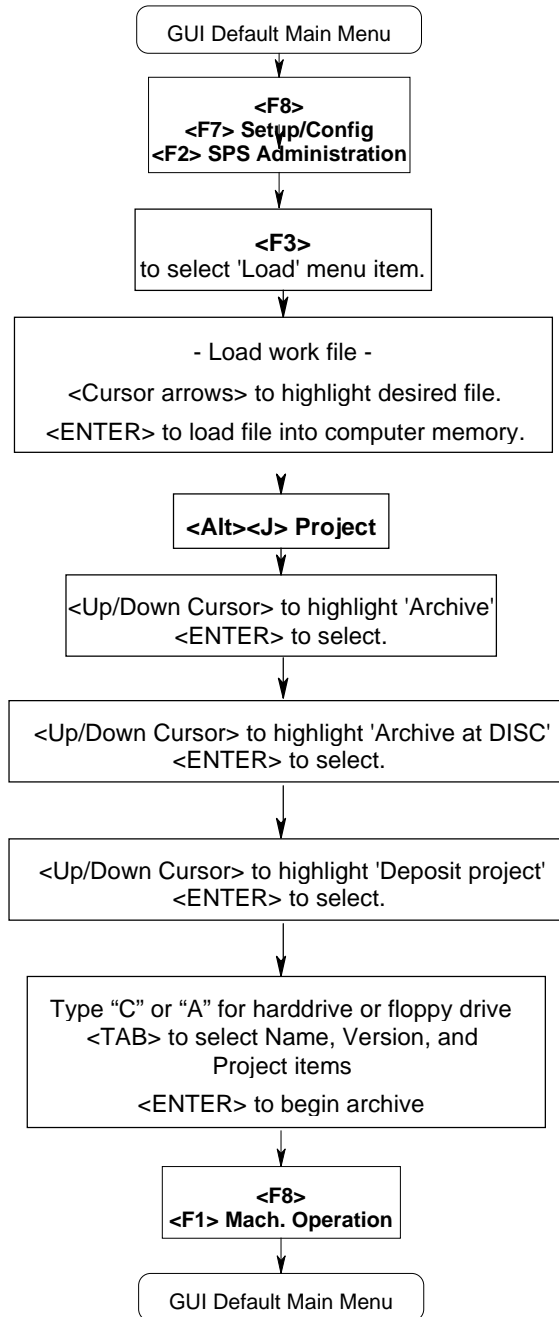
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.2.5 SPS (PLC) Project Archive, GUI ⁽⁴⁾ (Indramat default)

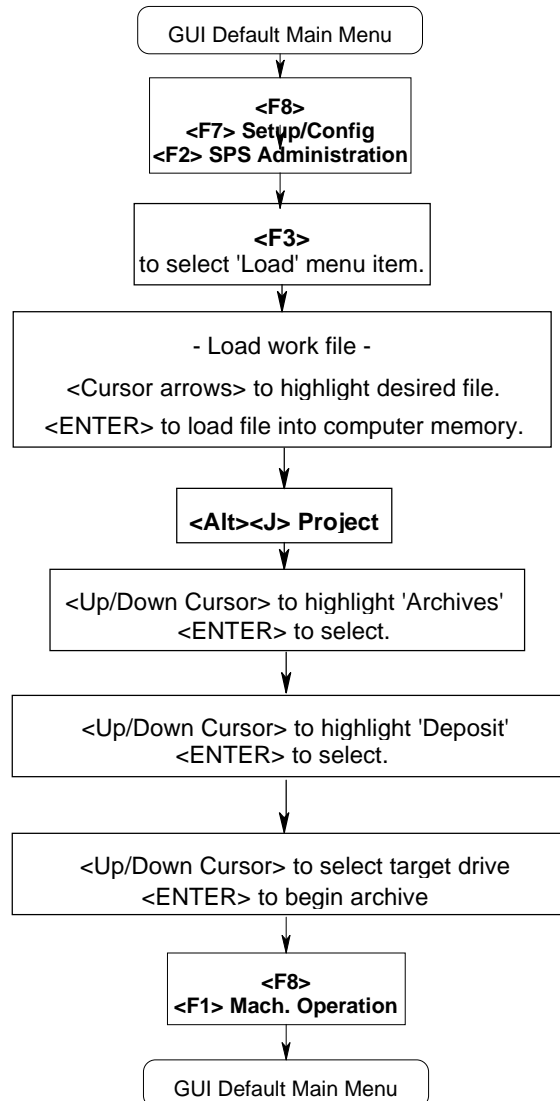
Figure 2-9 GUI SPS (PLC) Project Archive ⁽⁴⁾

(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.2.6 SPS (PLC) Project Archive, GUI ⁽⁵⁾ (Indramat default)**Figure 2-10 GUI SPS (PLC) Project Archive** ⁽⁵⁾

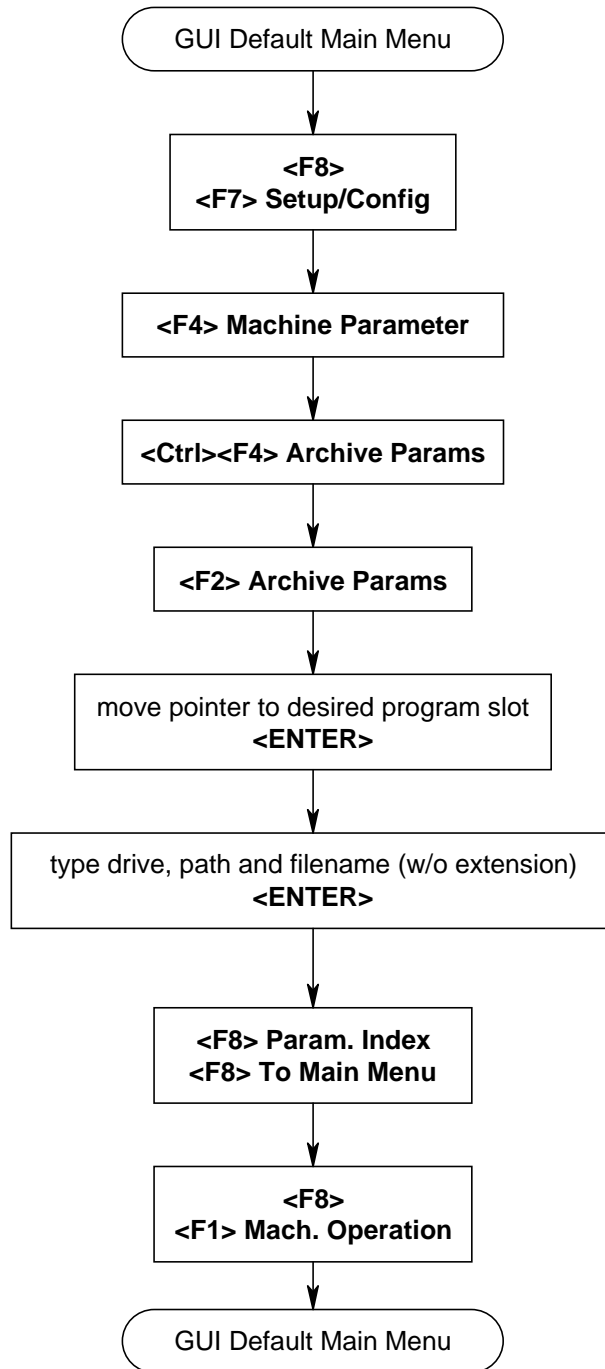
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.2.7 Machine Parameter Archive, GUI (Indramat default)

Figure 2-11 GUI Machine Parameter Archive

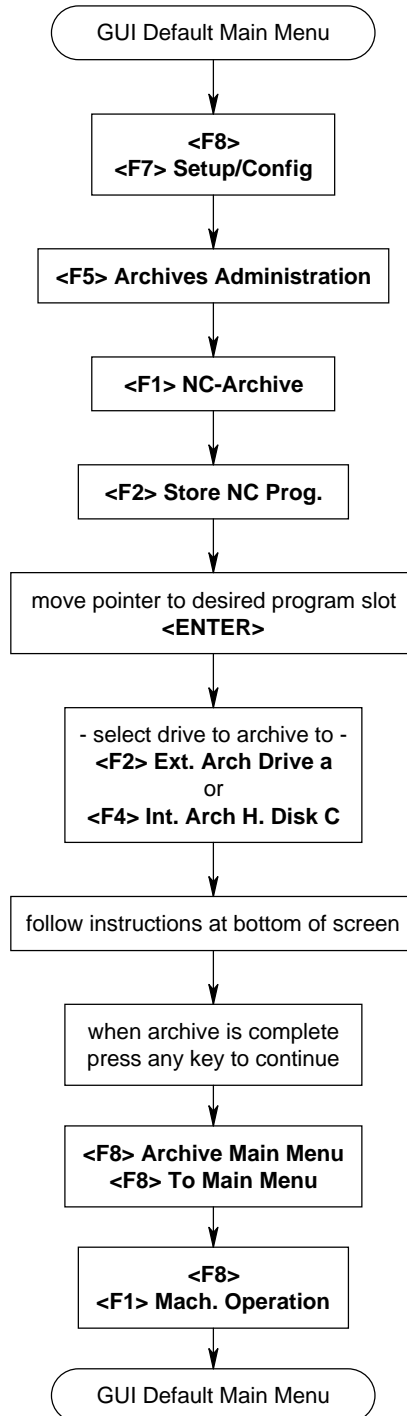
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.2.8 NC-program (w. Tool Setup List) Archive, GUI (Indramat default)

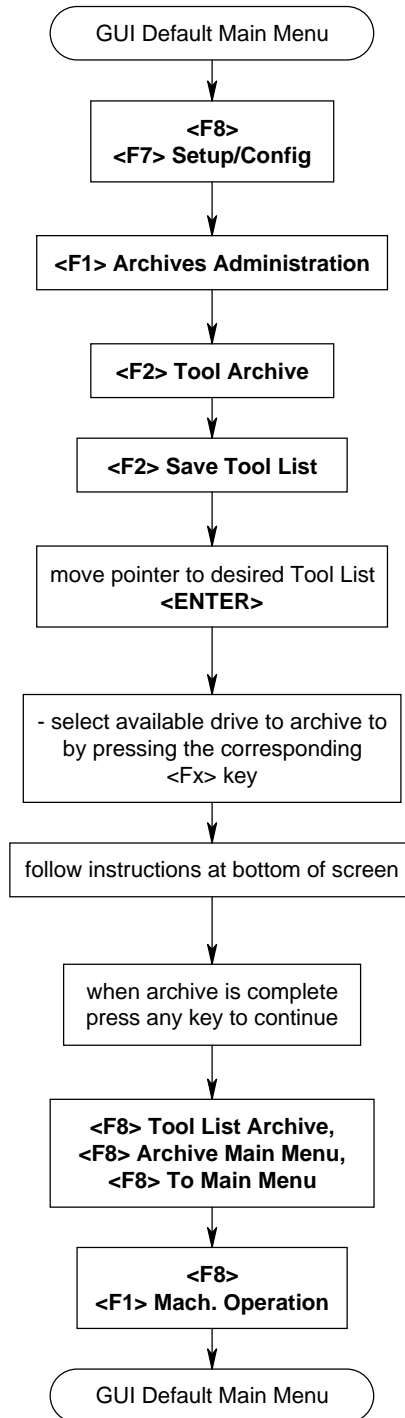
Figure 2-12 GUI Online/Off-line Mode

(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.2.9 Tool List Archive, GUI (Indramat default)

Figure 2-13 GUI Tool List Archive

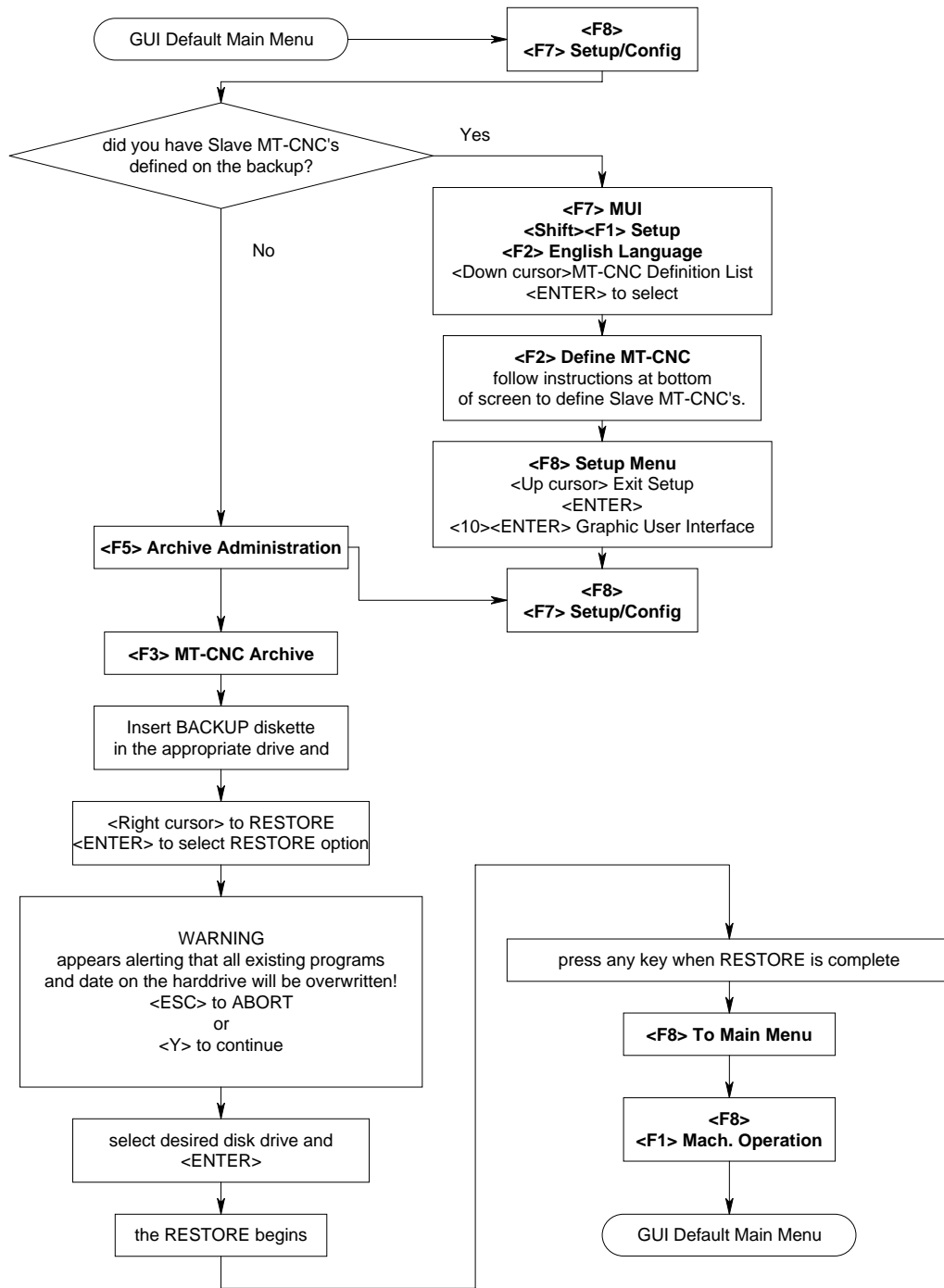


2.3 Restore, GUI (Indramat default)

2.3.1 Total MT-CNC User Programs & Data Restore, GUI (Indramat default)

Figure 2-14 GUI Online/Off-line Mode

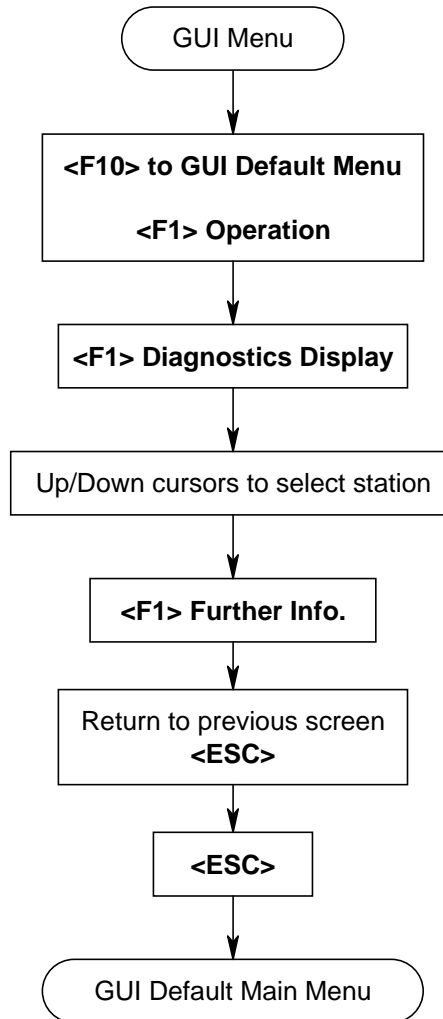
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.4 Troubleshooting and Diagnostics, GUI (Indramat default)

2.4.1 Further Information to Diagnostics, GUI (Indramat default)

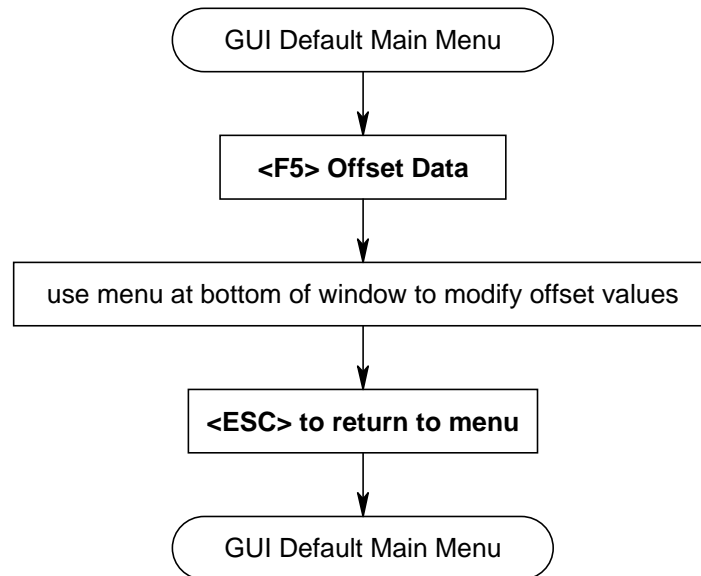
Figure 2-15 GUI Further Information to Diagnostics



2.4.2 Offset Table Data, GUI (Indramat default)

Figure 2-16 GUI Offset Table Data

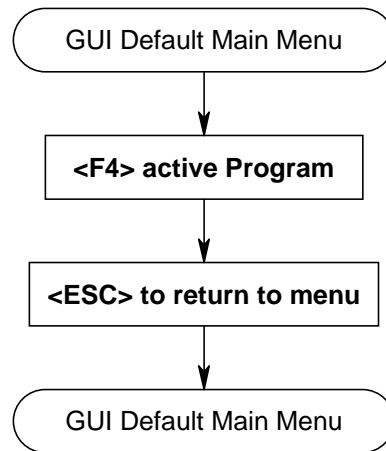
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.4.3 Active NC-Program, GUI (Indramat default)

Figure 2-17 GUI Active NC-Program Viewing

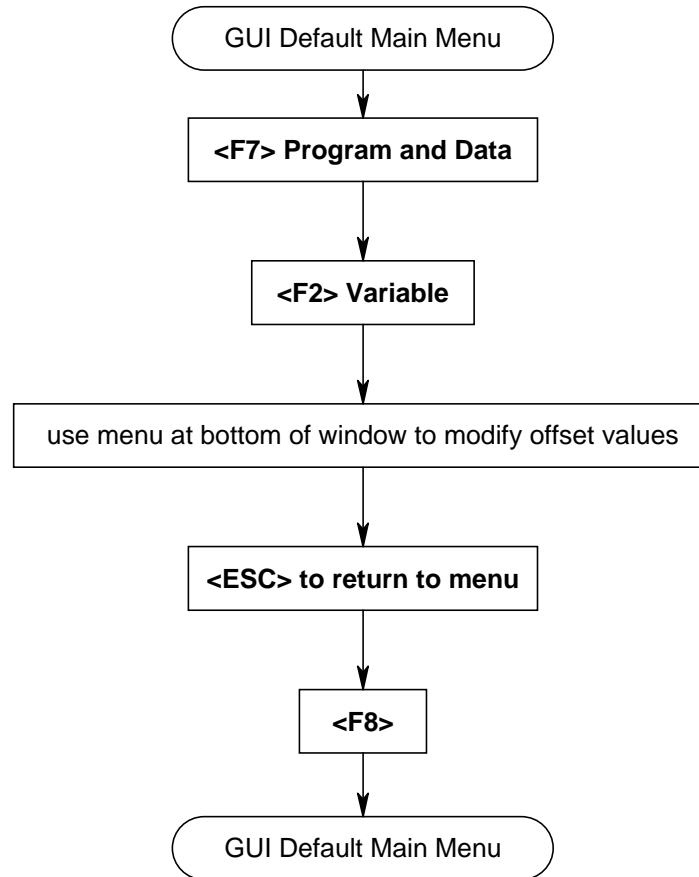
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.4.4 NC-Variables, GUI (Indramat default)

Figure 2-18 GUI NC Program Variables

(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.4.5 Oscilloscope, GUI (Indramat default)

Figure 2-19 GUI Oscilloscope (Sheet 1 of 2)

(this chart is only valid for the default GUI screens supplied with the installation diskettes)

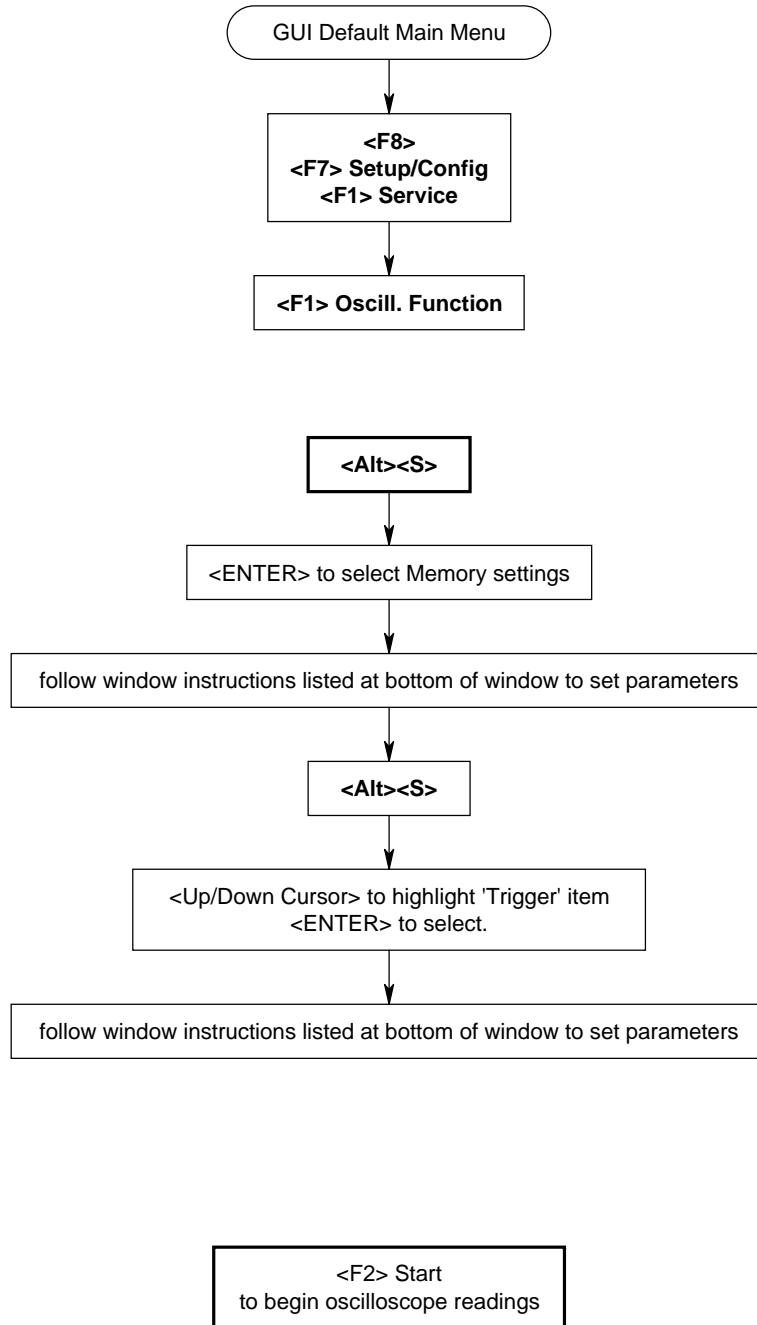
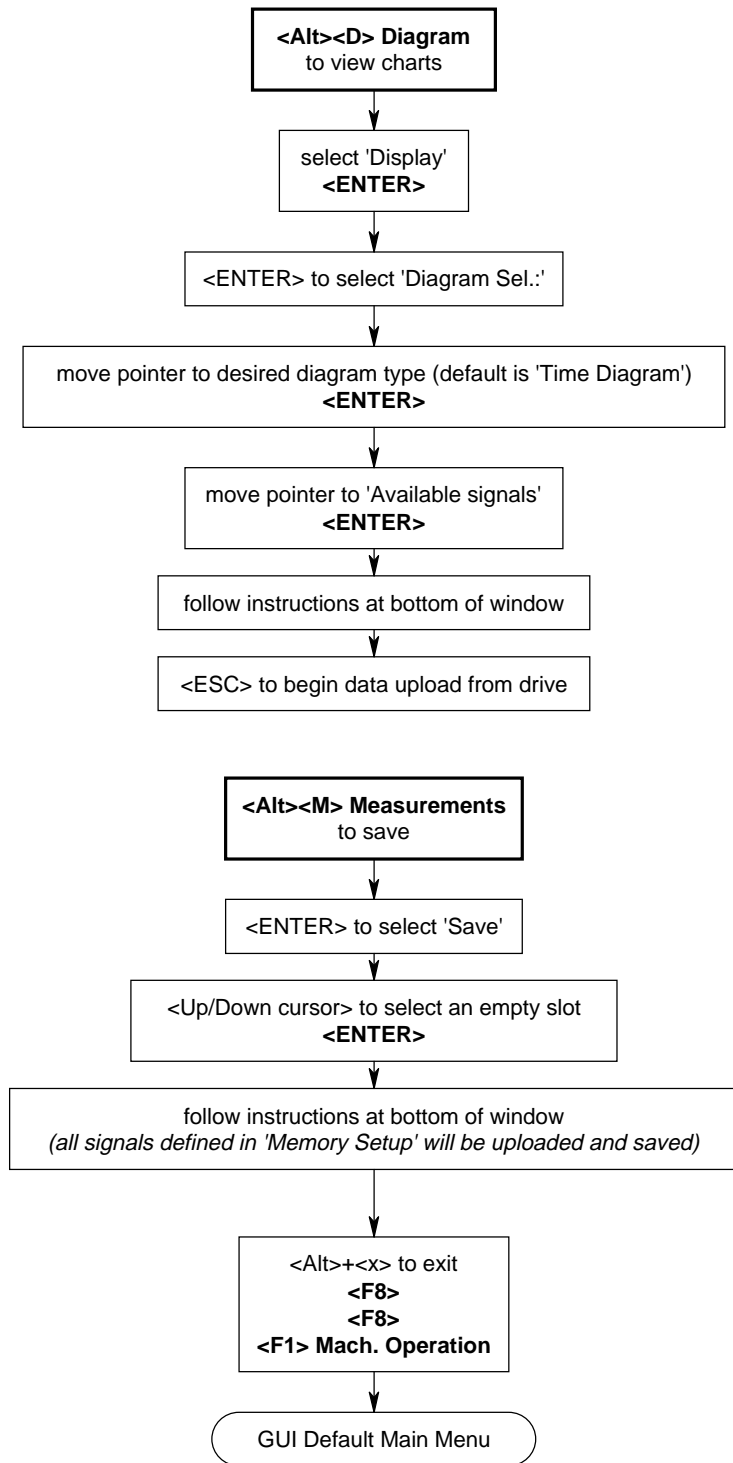


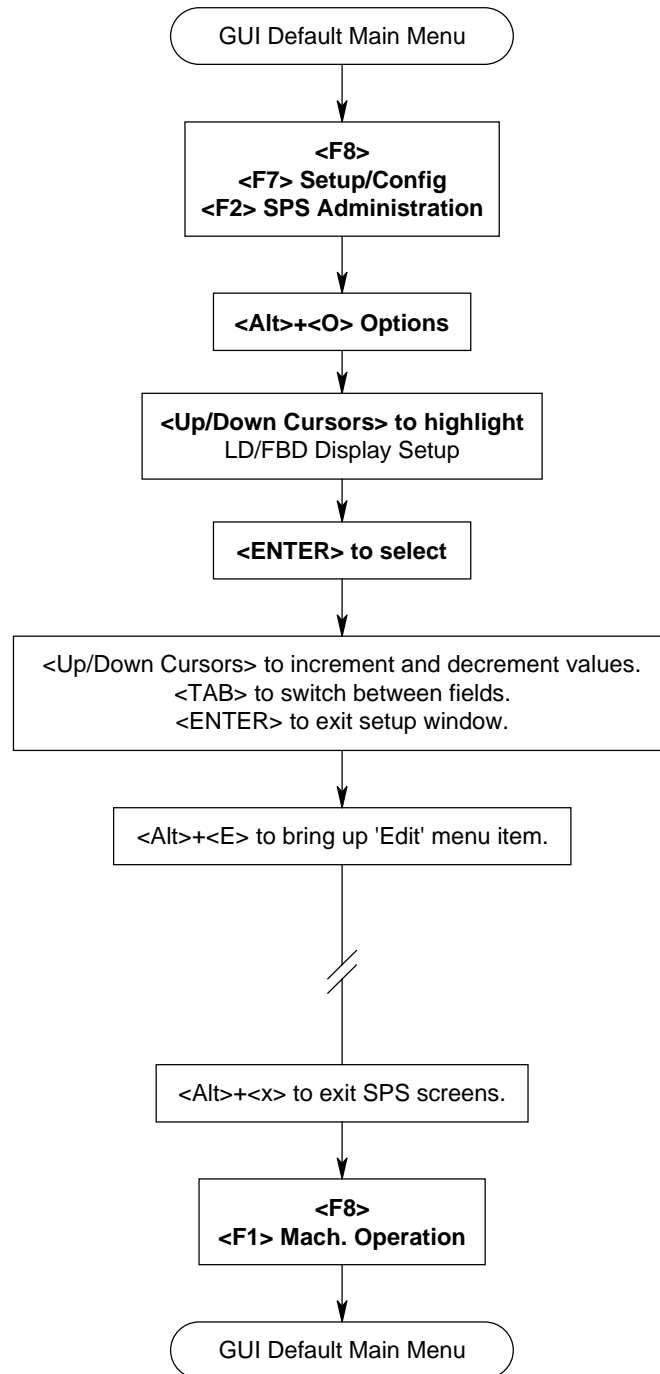
Figure 2-20 GUI Oscilloscope (Sheet 2 of 2)



2.4.6 PLC (SPS) Logic Display Setup, GUI (Indramat default)

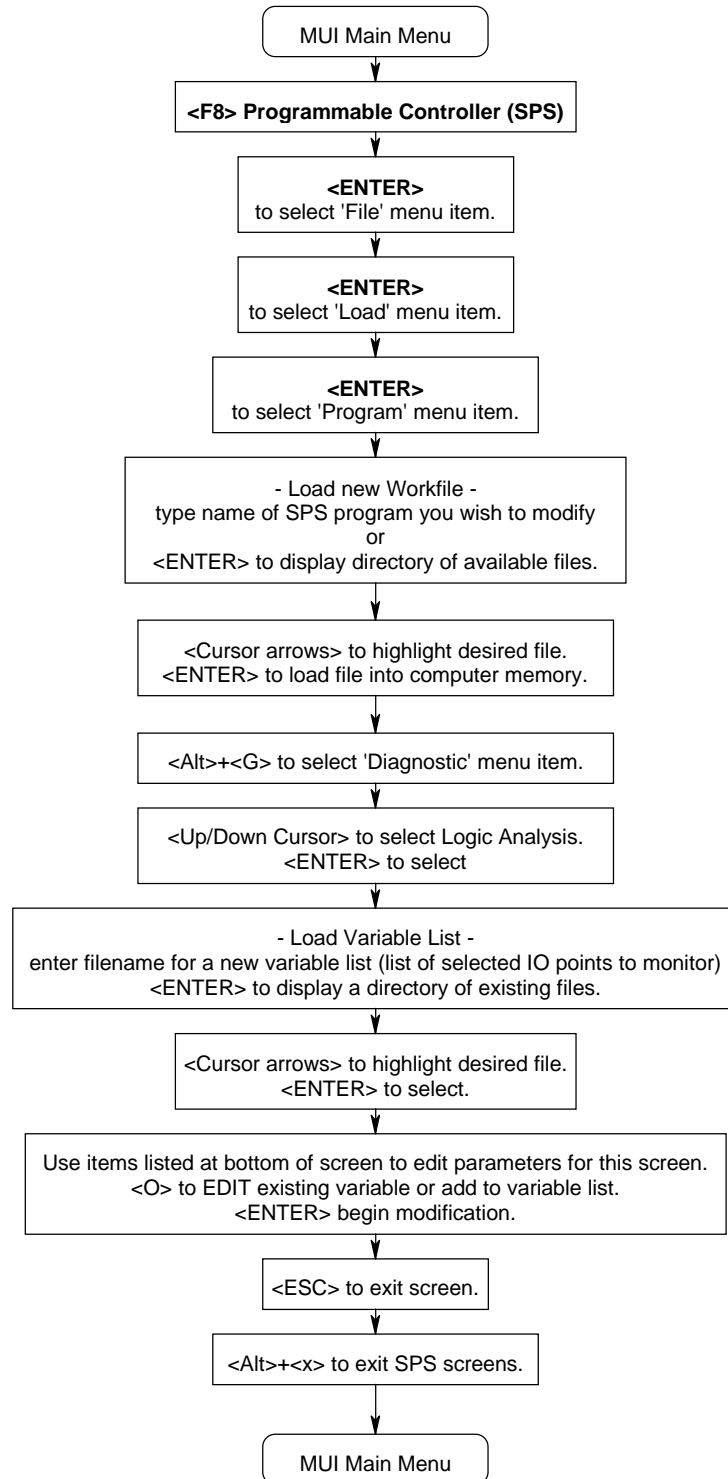
Figure 2-21 GUI SPS Logic Display Setup

(this chart is only valid for the default GUI screens supplied with the installation diskettes)



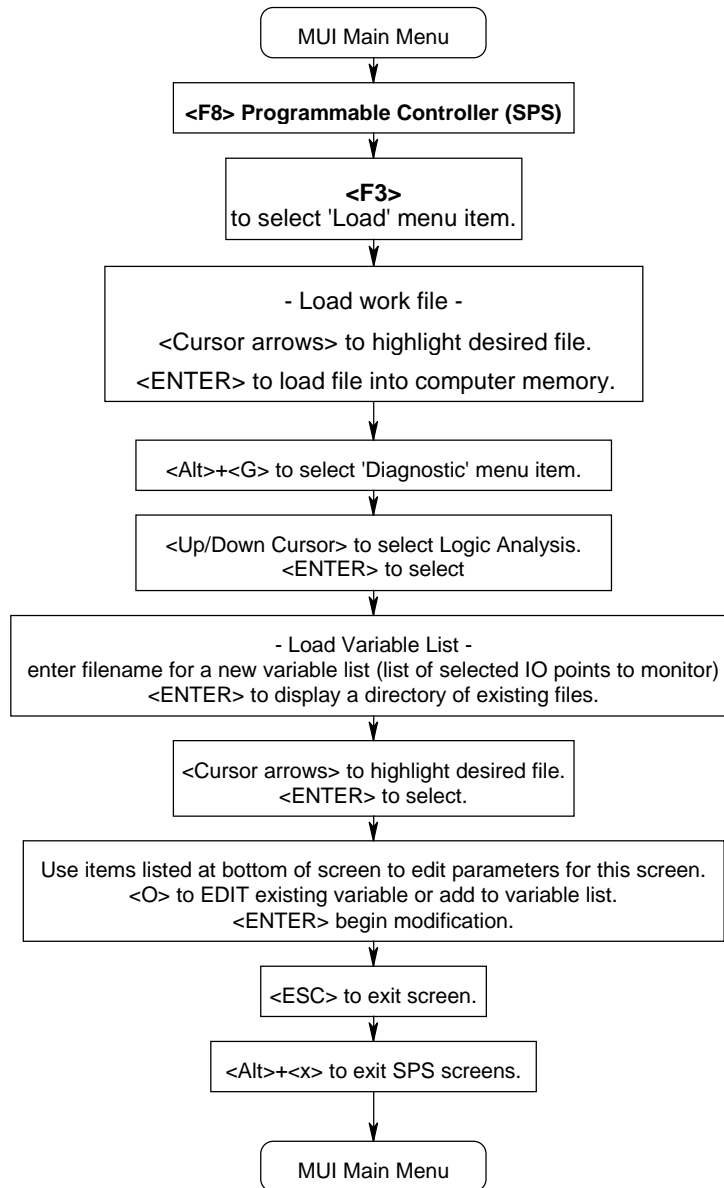
2.4.7 PLC (SPS) Logic Analysis, GUI ^{(1),(2),(3)} (Indramat default)

Figure 2-22 GUI SPS Logic Analysis ^{(1),(2),(3)}



2.4.8 PLC (SPS) Logic Analysis, GUI ^{(4),(5)} (Indramat default)

Figure 2-23 GUI SPS Logic Analysis ^{(4),(5)}



2.5 Editing, GUI (Indramat default)

2.5.1 Machine Parameter Editing, GUI (Indramat default)

Figure 2-24 GUI Machine Parameter Editing (Sheet 1 of 2)

(this chart is only valid for the default GUI screens supplied with the installation diskettes)

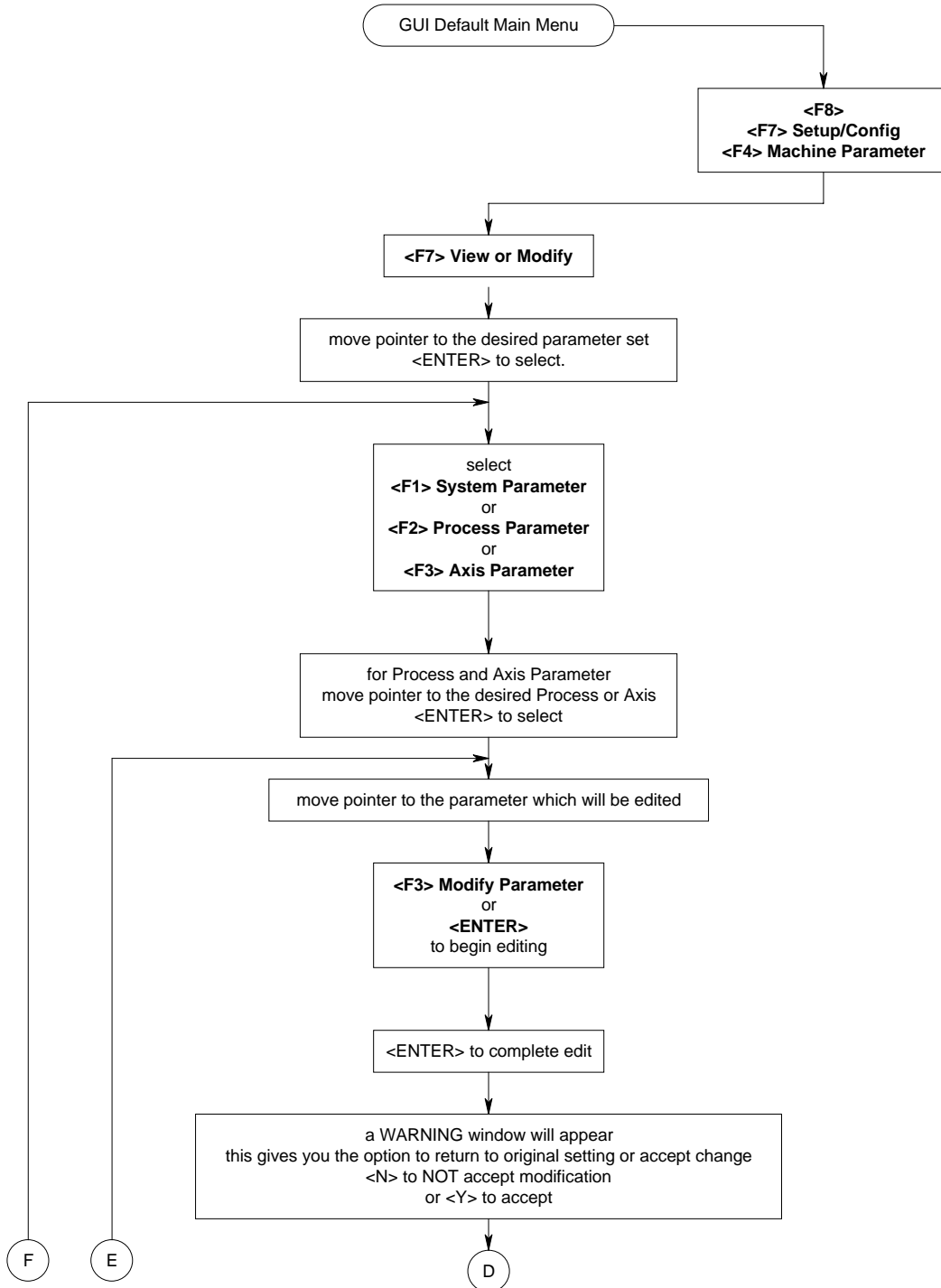
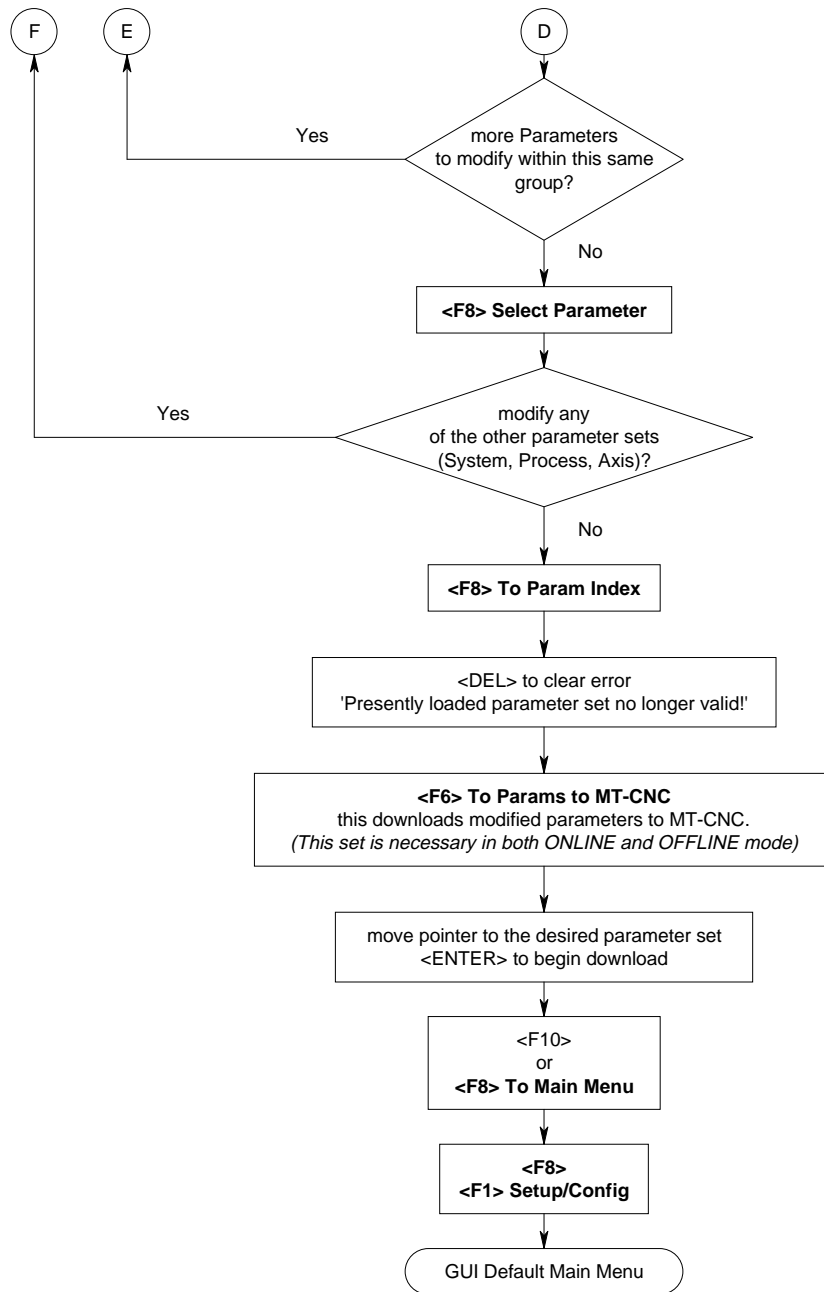


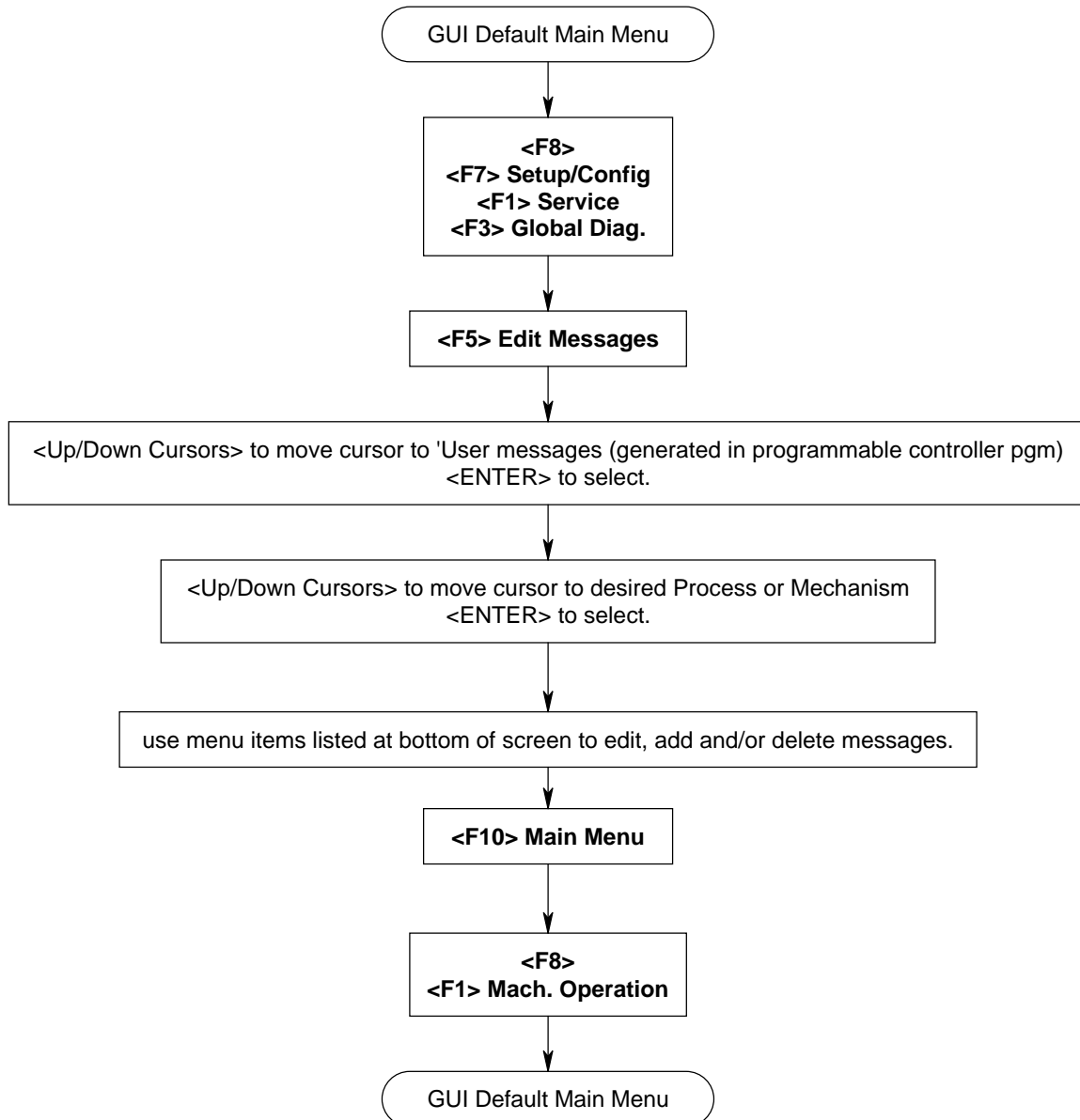
Figure 2-25 GUI Machine Parameter Editing (Sheet 2 of 2)



2.5.2 Diagnostic Messages Editing, GUI (Indramat default)

Figure 2-26 GUI Diagnostic Message Editing

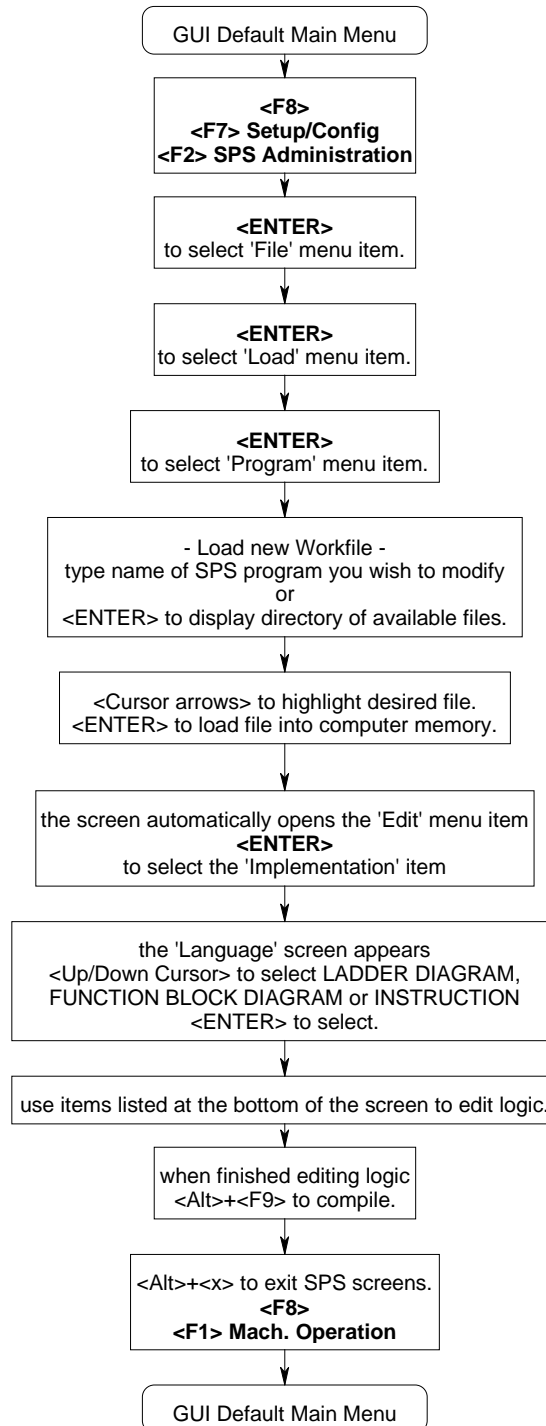
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.5.3 SPS (PLC) Program Editing, GUI ^{(1),(2),(3)} (Indramat default)

Figure 2-27 GUI SPS (PLC) Editing ^{(1),(2),(3)}

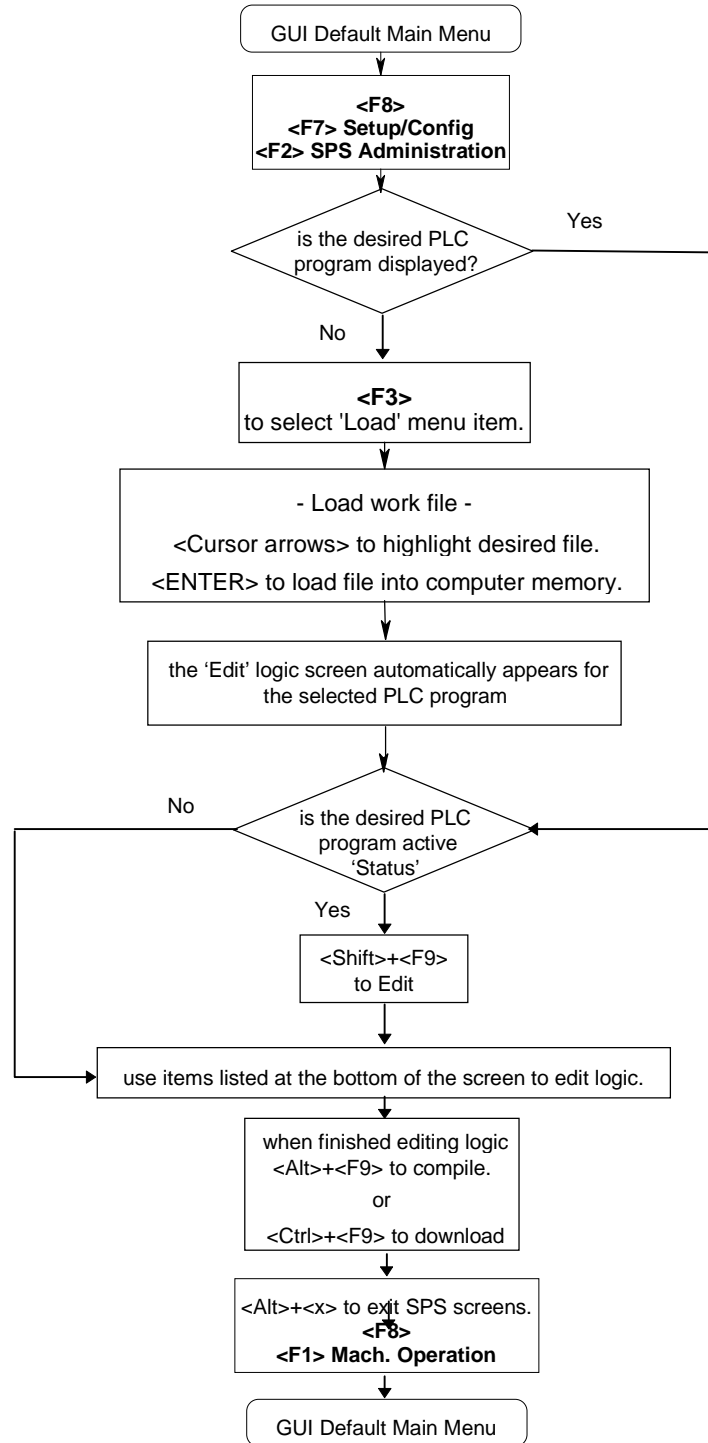
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.5.4 SPS (PLC) Program Editing, GUI ^{(4),(5)} (Indramat default)

Figure 2-28 GUI SPS (PLC) Editing ^{(4),(5)}

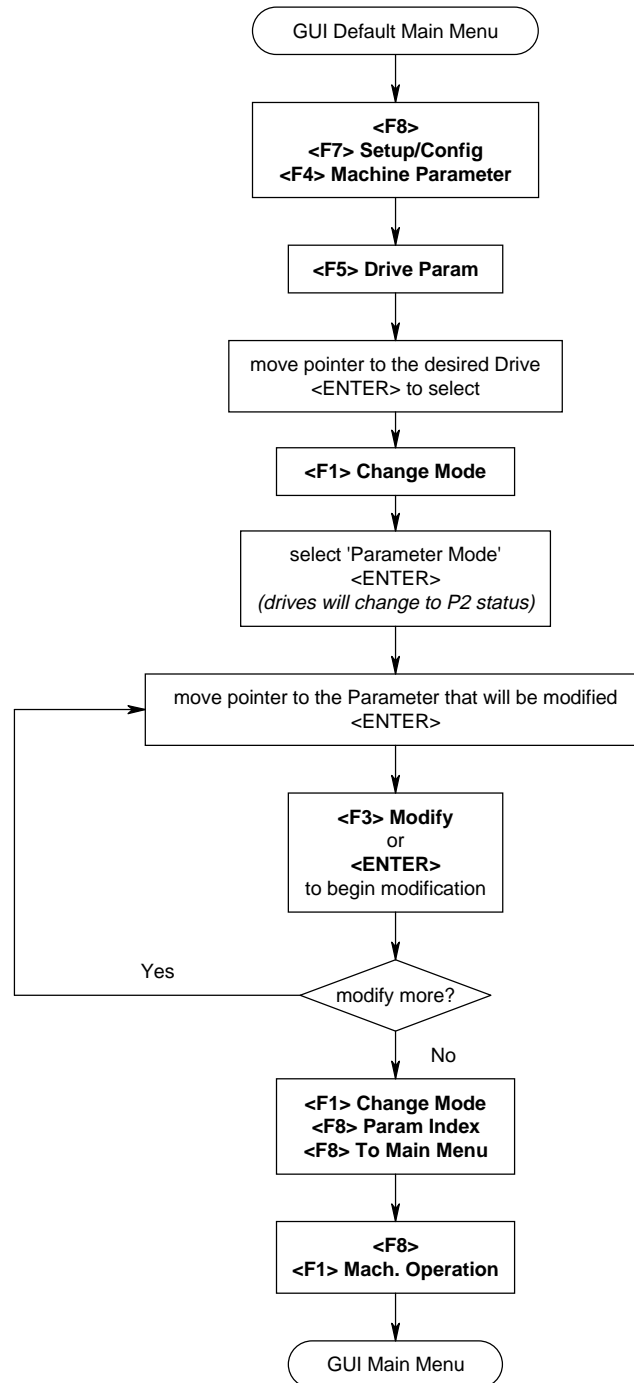
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.5.5 Digital Drive Parameter Editing, GUI (Indramat default)

Figure 2-29 GUI Digital Drive Editing

(this chart is only valid for the default GUI screens supplied with the installation diskettes)

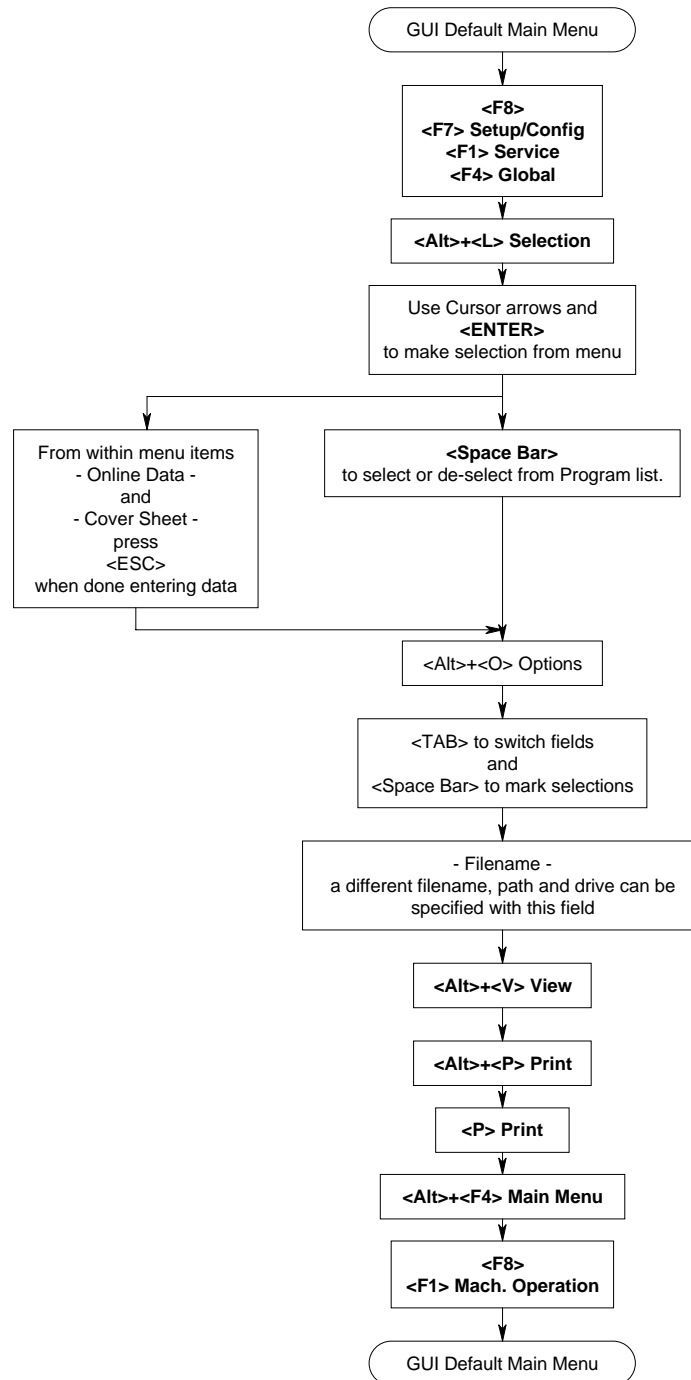


2.6 Documentation, GUI (Indramat default)

2.6.1 Global Documentation, GUI (Indramat default)

Figure 2-30 GUI Global Documentation

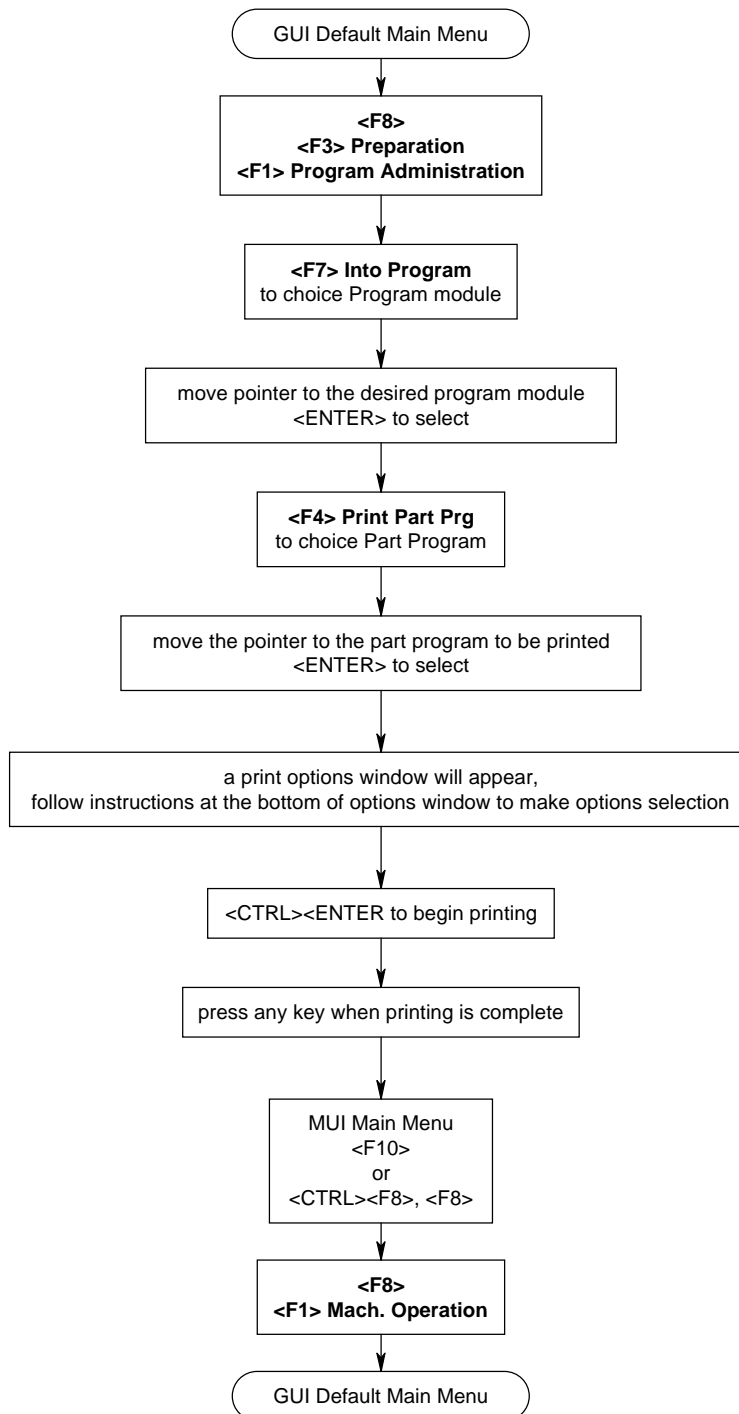
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.6.2 NC-Program Print Out, GUI (Indramat default)

Figure 2-31 GUI NC-Program Print Out

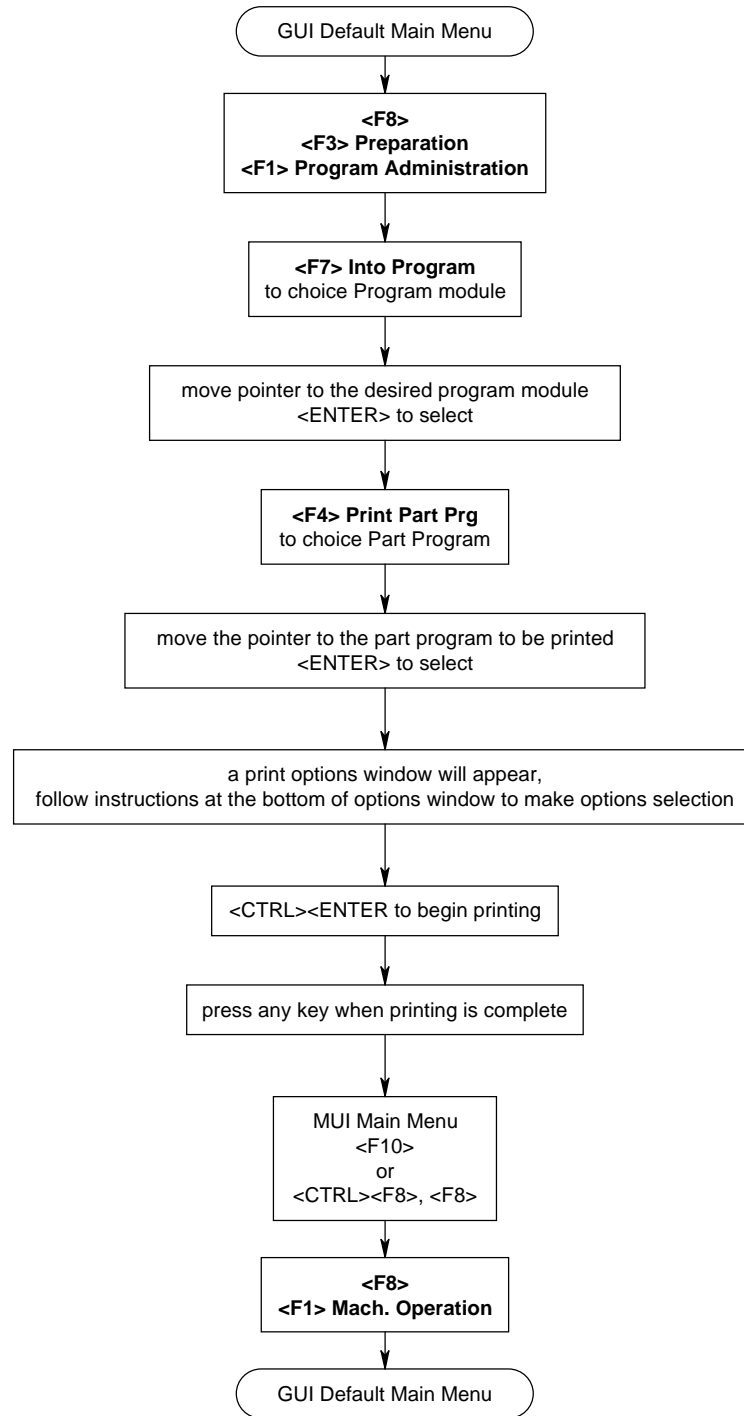
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.6.3 Machine Parameter Print Out, GUI (Indramat default)

Figure 2-32 GUI Machine Parameters Print Out

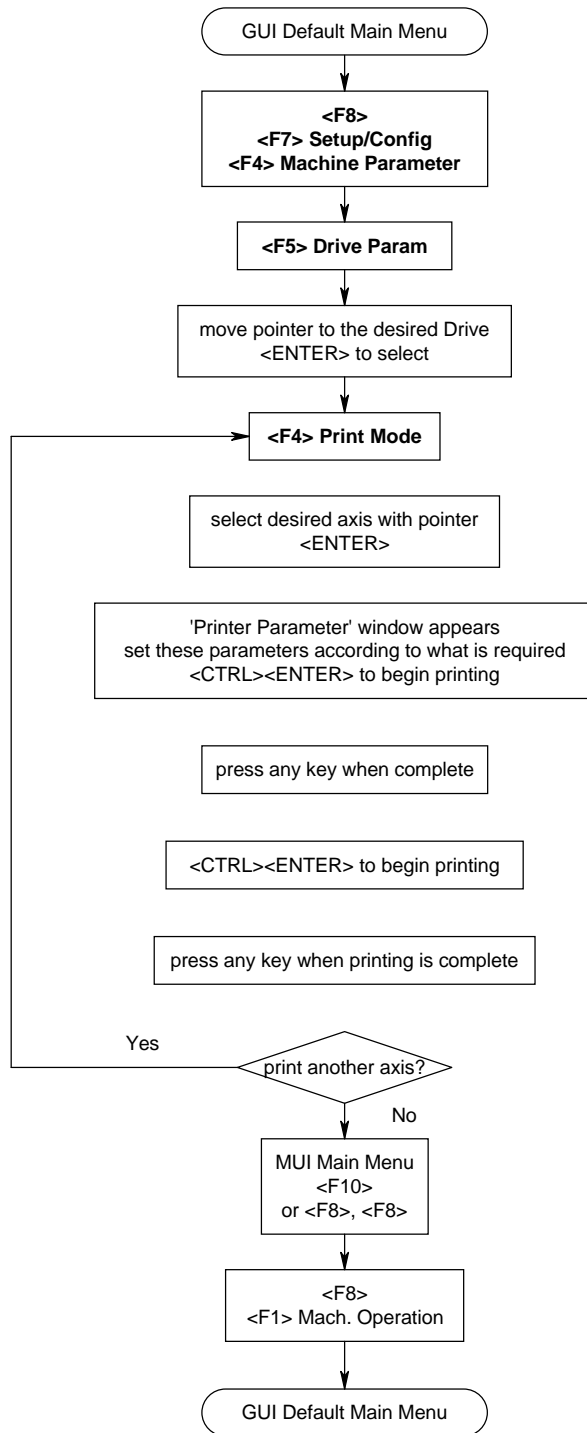
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.6.4 Digital Drive Parameter Print Out, GUI (Indramat default)

Figure 2-33 GUI Digital Drive Parameters Print Out

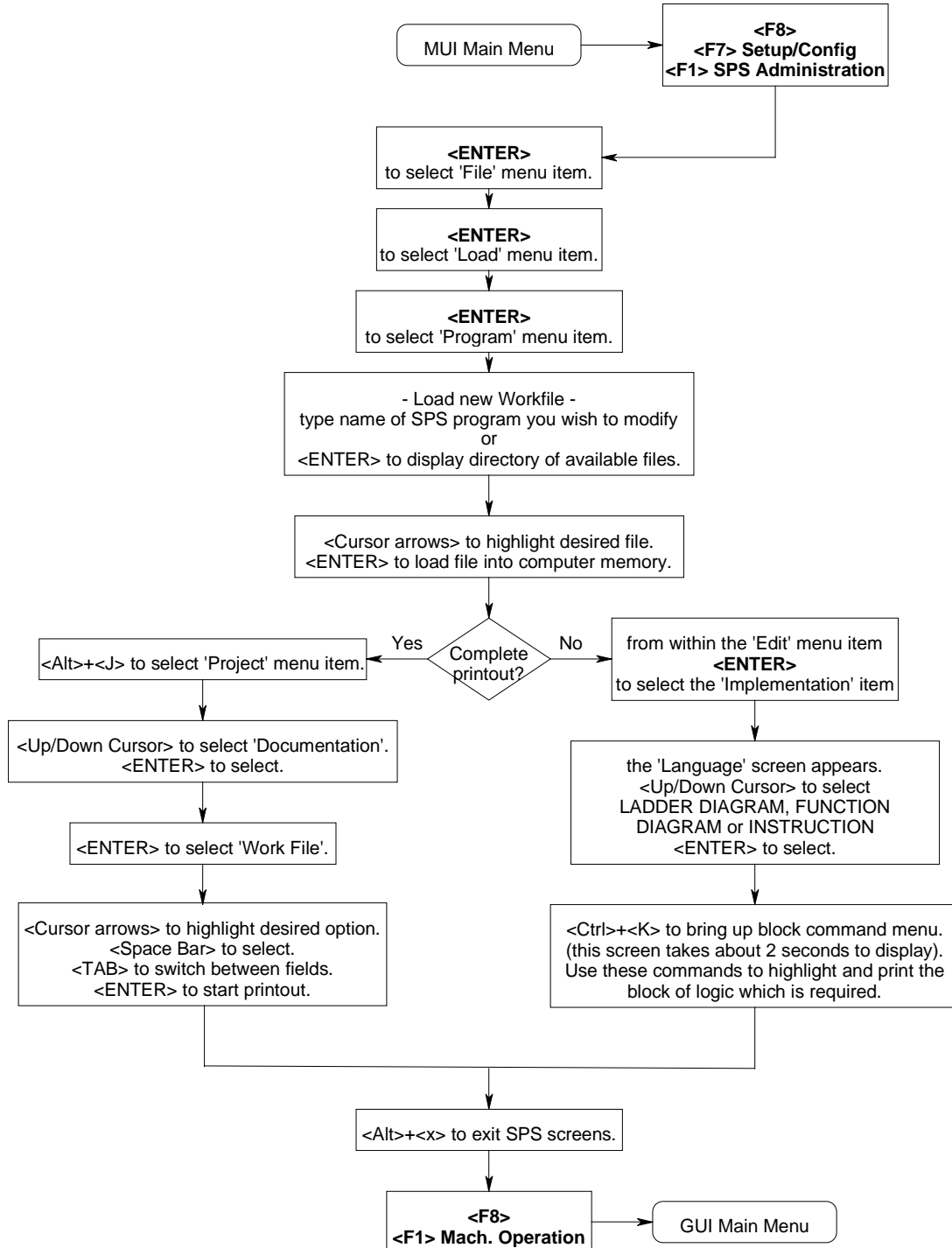
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.6.5 SPS (PLC) Documentation Print Out, GUI ^{(1),(2),(3)} (Indramat default)

Figure 2-34 GUI SPS Documentation Print Out ^{(1),(2),(3)}

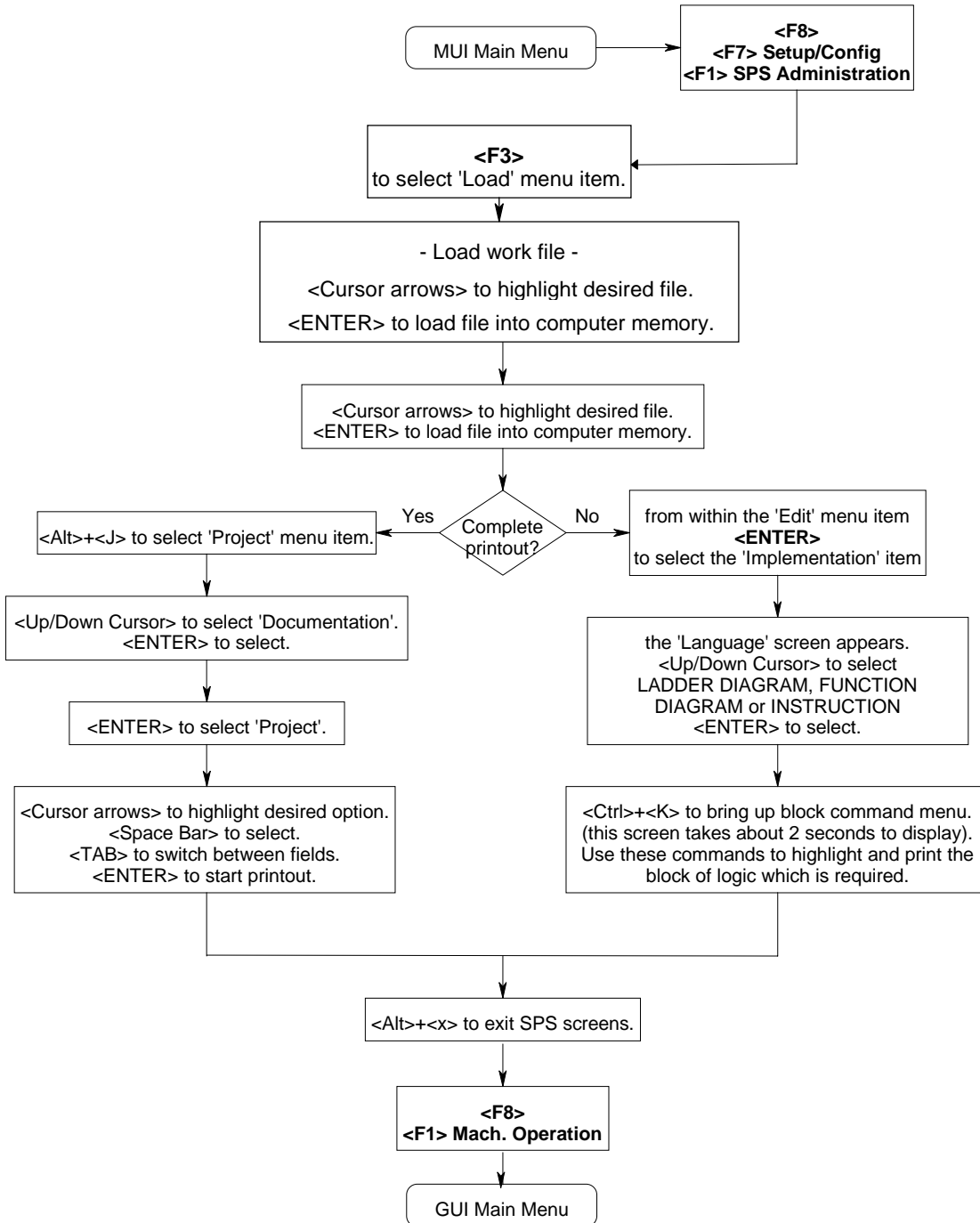
(this chart is only valid for the default GUI screens supplied with the installation diskettes)



2.6.6 SPS (PLC) Documentation Print Out, GUI ^{(4),(5)} (Indramat default)

Figure 2-35 GUI SPS Documentation Print Out ^{(4),(5)}

(this chart is only valid for the default GUI screens supplied with the installation diskettes)



3. PARAMETERS

3.1 System Parameters

- A00.000** **Management Process**
This parameter describes the Management process. This is the default parameter set and **MUST ALWAYS BE DEFINED**. This process can be used to control all other processes.
- A00.001-A00.006** **Process (1...6)**
These parameters define internal motion processes. Internal processes must have one or more axes directly controlled by the MT-CNC.
- A00.007-A00.031** **External Mechanism (7...31)**
These parameters define external mechanisms. This means the management process can access external devices via the SPS. These devices may include single axis TRANS-01 controls⁽¹⁾, slave MT-CNC controls, or other PLC interface control devices.
- A00.032-A00.51** **Axis (1...20)**
These parameters are used and type of axis connected to the to assign the number MT-CNC axis processors (APR). The axis can be defined as Linear, Rotary, Main Spindle or Turret, however upon transfer of the parameter set to the MT-CNC the correct hardware must be installed or a system fault will be initiated.
- A00.052** **Tool Management**
This parameter selects the tool management functions for a process requiring the ability to use multiple tools, monitor and/or correct for tool wear, or monitor number of cycles. If set to 'yes' further system parameters A00.053 - A00.083 appear which configure tool data sets.
- A00.087** **Global M-function Mode^{(4),(5)}**
This process parameter defines the output and acknowledge mode of all M-functions globally over all processes.
- A00.088** **Global Q-function Mode^{(4),(5)}**
This process parameter defines the output and acknowledge mode of all Q-functions globally over all processes.
- A00.089** **S-function Mode^{(4),(5)}**
This process parameter defines the output and acknowledge mode of all S-functions globally over all processes.
- A00.090** **T/E-function Mode^{(4),(5)}**
This parameter defines the output and acknowledge mode of all T-functions (Tool) and E-functions (Tool Edge) globally over all processes.

3.2 Process Parameters

xx	Can be processes (00...06).
Bxx.001	Default System of Units for Programming (G70/G71) Sets default measuring units (mm, inch) for part programs, tool offsets, and zero offsets.
Bxx.002	Programmable Decimal Places for Distances Sets number of decimal places used when programming axis.
Bxx.003	Displayed Decimal Places for Distances Sets the number of decimal places displayed on the position display.
Bxx.004	Default Interpolation Plane Sets the default reference plane G17, G18, or G19, for circular interpolation, tool data, D-corrections, and offset data.
Bxx.005	Max. Path Velocity Defines maximum path speed for all axes declared in this process.
Bxx.006	Max. Path Velocity Change W/O Ramp Defines the velocity at which the path acceleration switches to a linear ramp at the rate defined in Bxx.007.
Bxx.007	Path Acceleration Defines the path acceleration above the velocity programmed in Bxx.006.
Bxx.012	Max. Number of Offset Tables Specifies how many work offset memories for G54 through G59 are to be configured.
Bxx.013	Diameter Programming Determines whether diameter G16 or radius G15 programming is chosen for the main axis of that process.
Bxx.029	D-Correction^{(3),(4),(5)} Allows the use of D-Correction D00 - D99 in the process NC-program and provides parameters Bxx.033 - Bxx.031 for entry limit definitions.
Bxx.030	Maximum Entry for Length D Correction^{(3),(4),(5)} The maximum entry value (via MUI/GUI and SOT) for the D-correction length data L1, L2 and L3 can be set in this parameter. This parameter is only available if Bxx.029 is set to yes.
Bxx.031	Maximum Entry for Radius D Correction^{(3),(4),(5)} The maximum entry value (via MUI/GUI and SOT) for the D-correction length data L1, L2 and L3 can be set in this parameter. This parameter is only available if Bxx.029 is set to yes.
Bxx.032	Reverse Program Execution Required Determines if a reverse program needs to be complete before a forward program can be executed. This parameter is only effective if Bxx.033 is set to NO.
Bxx.033	Reverse Required Determines if a G74 is needed in the reverse section of a NC program under the .HOME label.

CAUTION: *If Bxx.032 and Bxx.033 are both set to 'NO' in a process with at least one incremental servo axis, all axes in the process are able to execute a forward program without homing!*

- Bxx.034** **Acceleration Time Constant**
Specifies a time value for the rate of acceleration starting at 0 units/S², to the rate defined in Bxx.007.
- Bxx.035** **Cartesian-Polar Coordinate Transformation**
Allows the use of 'C-axis' mode (630/631) with APRB04 and SERCOS main spindles.
- Bxx.036** **Manual Axis Jogging Causes Reset**^{(2),(3),(4),(5)}
A 'Control Reset' (resetting all active NC-commands and activating the defaults) is performed if this parameter is set to 'yes' and an axis is jogged when in manual operating mode.
- Bxx.037** **Linear Interpolation Type Default G0/G1**^{(2),(3),(4),(5)}
Default, linear interpolation.
- Bxx.038** **Tool Length Correction Default**^{(2),(3),(4),(5)}
This parameter allows to select G48 Tool length correction, positive or G49 Tool length correction, negative as the process default for modal G-code group {13}.
The preparatory function that is specified in this parameter is activated if process parameter Bxx.043 is set to 'default' and if the MT-CNC power is ON, after 'Control Reset' and at the NC-program end after the NC-command RET, BST, M02 and M30.
- Bxx.039** **Default Zero Offset (G54 - G59)**^{(2),(3),(4),(5)}
This parameter allows to select for default 'Offset Table 0' as the process default for modal G-code group {4}.
- Bxx.040** **Feed Programming Default(G94/G95)**^{(2),(3),(4),(5)}
This parameter allows to select G94 'Feed per minute' or G95 'Feed per spindle revolution' as the process default for modal G-code group {7}.
- Bxx.041** **Spindle Speed Programming Default (G96/G97)**^{(2),(3),(4),(5)}
This parameter allows to select G96 'Constant surface speed (CSS)' or G97 'Spindle speed in RPM' as the process default for modal G-code group {8}.
- Bxx.042** **Trigonometric Arguments, Unit Default (RAD, DEG)**^{(2),(3),(4),(5)}
This parameter allows to select RAD 'Radians' or DEG 'Degrees' as the process default for trigonometric NC-functions (e.g. SIN, etc.).
- Bxx.044** **Asynchronous Turret Movement**⁽³⁾
Normally, a tool storage turret is not moved asynchronously in the process. However if an application would require this, this parameter could be set to 'yes'.
- Bxx.046** **Deviating M-functions**^{(4),(5)}
Through the use of this parameter up to 40 M-functions whose mode deviates from the setting in parameter (A00.087) can be defined for this process. The mode can be defined individually in form of list entry for each of the 40 M-functions.

- Bxx.047** **Deviating Q-functions^{(4),(5)}**
 Through the use of this parameter up to 40 Q-functions whose mode deviates from the setting in parameter (A00.088) can be defined for this process. The mode can be defined individually in form of list entry for each of the 40 Q-functions.
- Bxx.048** **M-function of functional group 5^{(4),(5)}**
 This parameter allows the user to define up to 12 M-functions that should belong to the M-function group 5. Up to 12 M-functions can be assigned to a M-function group.
- Bxx.049** **M-function of functional group 6^{(4),(5)}**
 This parameter allows the user to define up to 12 M-functions that should belong to the M-function group 6. Up to 12 M-functions can be assigned to a M-function group.
- Bxx.050** **M-function of functional group 7^{(4),(5)}**
 This parameter allows the user to define up to 12 M-functions that should belong to the M-function group 7. Up to 12 M-functions can be assigned to a M-function group.
- Bxx.051** **M-function of functional group 8^{(4),(5)}**
 This parameter allows the user to define up to 12 M-functions that should belong to the M-function group 8. Up to 12 M-functions can be assigned to a M-function group.
- Bxx.052** **M-function of functional group 9^{(4),(5)}**
 This parameter allows the user to define up to 12 M-functions that should belong to the M-function group 9. Up to 12 M-functions can be assigned to a M-function group.
- Bxx.053** **M-function of functional group 10^{(4),(5)}**
 This parameter allows the user to define up to 12 M-functions that should belong to the M-function group 10. Up to 12 M-functions can be assigned to a M-function group.
- Bxx.054** **M-function of functional group 14^{(4),(5)}**
 This parameter allows the user to define up to 12 M-functions that should belong to the M-function group 14. Up to 12 M-functions can be assigned to a M-function group.
- Bxx.055** **M-function of functional group 15^{(4),(5)}**
 This parameter allows the user to define up to 12 M-functions that should belong to the M-function group 15. Up to 12 M-functions can be assigned to a M-function group.
- Bxx.056** **Rotary Axes Mode Default^{(4),(5)}**
 The process default rotary mode G-code can be defined in this process parameter:
- G36 Shortest distance, with modulo calculation
 - G37 Positive direction, with modulo calculation
 - G38 Negative direction, with modulo calculation

3.3 Axis Parameters

3.3.1 Linear and Rotary Axes

*A = Available for Analog Servo Drives only.

*D = Available for Digital Servo Drives only.

xx Can be processes (00...06).

Cxx.000 **Associated Processes**
 Assigns the axis to one or more (up to 4) processes.

- Cxx.001 Axis Designation 1**
Assigns a programming name to the axis. Axis designation 1 represents primary axis name used when programming in NC-programs.
- Cxx.003 Position Feedback Location *A**
Defines the location of the position feedback device (encoder). This entry will determine whether Cxx.004 or Cxx.005 will be used. (Motor, Gearbox Output, Linear Encoder)
- Cxx.004 Incremental Encoder Resolution *A**
Linear Axis:
Defines the number of lines per revolution of the rotary encoder connected to this axis. This parameter is available when Cxx.003 is set to (Motor or Gearbox Output).
Rotary Axis:
Defines the number of lines per revolution of the rotary encoder connected to this axis.
- Cxx.005 Linear Encoder Resolution *A**
Defines the resolution of the linear scale connected to this axis. This parameter is available only when Cxx.003 is set to (Linear Encoder).
- Cxx.006 Ballscrew lead *A (linear axis)**
Defines the value for the lead of the ballscrew. This is the distance traveled per revolution of the ballscrew.
Programming Units / Table Revolution (rotary axis)
Defines the number of graduations (units) per revolution of the moved rotary table. Typically this parameter is set to 360 degrees/revolution.
- Cxx.007 Gearbox Revolutions, Input *A**
Specifies the value of input revolutions of a gearbox. This value MUST be a whole number.
- Cxx.008 Gearbox Revolutions, Output *A**
Specifies the value of output revolutions of a gearbox. This value MUST be a whole number.
- Cxx.009 Motor Speed at 9 Volts Command *A**
Calibrates the D/A converter in the analog APR module for motor speed at 9 volts command.
- Cxx.010 Kv Factor *A**
Is used to adjust the response of the servo axis. A low Kv Factor will cause the axis to react sluggish, and a high Kv Factor will cause the axis to oscillate.
- Cxx.011 Positive/Negative Software Travel Limits**
Cxx.012
Linear Axis:
Become active only after reference (home) has been achieved either by single axis homing or via G74 in the NC-program. In addition, axis parameter Cxx.085 allows to specify when software travel limits are activated.^{(3),(4),(5)} Once established, the limits will not allow travel outside the positions entered. Before axis is homed, it can be jogged (+) or (-) the distance between the positive and negative software travel limits.
Rotary Axis:
- A non modulo (non endless rotary axis) can be defined by setting the positive and negative software travel limits to different values.
 - A modulo (endless rotary axis) can be defined by setting the positive and negative software travel limits to the same value or to '0'.
- Cxx.013 Reference Position *A**

Specifies the value displayed and used by the NC program as the machine coordinate reference. After homing the axis, this value is loaded into the actual position register.

Cxx.014 Position Polarity *A

Designates the polarity (sign) of the coordinate system or the axis. A value of '0' signifies a normal coordinate polarity, while a 1 signifies a reverse polarity.

Cxx.015 Homing Direction *A

Specifies the direction the axis will search for the home switch. If set to '0' the axis will search in same direction as Cxx.014, if set to '1' the axis will search in the opposite direction.

Cxx.016 Maximum Feedrate

Sets the limited velocity of the axis. This is the velocity that the axis will try to achieve when a G0 is programmed without the F-word.

Cxx.017 Maximum Feedrate Change W/O RampLinear Axis:

Defines the velocity at which the acceleration switches from the Kv factor to a linear ramp defined in Cxx.018.

Rotary Axis:

For a rotary table axis, this parameter should be set close to zero so that the Maximum Acceleration Rate (Cxx.018) will adjust the smooth operation of the tables rotation.

Cxx.018 Maximum Acceleration Rate

Defines the acceleration rate ramp that is activated when above the feedrate defined in Cxx.017.

Cxx.019 Homing Speed *A

Specifies the speed the axis will home. This value will be used when a G74 is programmed without the F-word, or the axis is sent home manually using the AXC.HOME gateway signal.

Cxx.020 Jogging Speed

Specifies the speed the axis will jog using the AXC.JGPOS or AXC.JGNEG gateway signals.

Cxx.021 Jogging Speed - Rapid

Specifies the speed the axis will jog when using the AXC.JGPOS or AXC.JGNEG Gateway signals, along with the PXC.RAPID Gateway signal. This speed is only available after the axis has been homed.

Cxx.022 Parametric Jogging Distance

Defines the incremental jogging distance when using the Parametric jogging mode, via the Gateway signals Pxx.JOGMx. The axis will move this distance while the AxxC.JGPOS or AxxC.JGNEG Gateway signals are activated in manual mode.

Cxx.023 In Position Window

Sets the maximum position loop deviation allowed. If axis is not within this value at block end, the block will not be completed, and NO ERROR WILL BE PRESENT.

- Cxx.024 Monitoring Window *A**
Defines the percentage of position error allowed before a fault is issued. A value of zero disables the monitoring window.
- Cxx.025 Waypoints 0 - 7**
Cxx.032 Are used to set actual axis positions to be monitored by the SPS program. If the absolute actual position is equal to or greater than the value set in one of the parameters 25 - 32, the associated SPS signal AxxS.WPn (n = 0 - 7) will become '1'.
- CAUTION:** *Waypoints are valid only after reference has been established for incremental axis.*
- Cxx.033 Home Limit Switch Input Address⁽¹⁾ *A**
This address eliminates the need to write the logic for the home switch in the SPS program, as well as providing a fast input for this switch. This parameter input can not be connected to a DEA input.
(XX,YY,ZZ) : XX → RECO Bit Number
YY → RECO Input Slot Number
ZZ → RECO Number
- Cxx.034 Automatic Backlash Compensation *A**
Defines the amount of mechanical backlash, between the motor and load during reversal relative to the position data, that should be compensated.
- Cxx.040 Position Feedback Direction *A**
Allows the user to reverse the counting direction of the position feedback device. A value of "0" signifies a normal counting direction. A value of "1" signifies a reverse counting direction, which is needed when the feedback and motor are separated by a single stage gearbox.
- Cxx.041 Tach Voltage per 1000 RPM *A**
Defines the tachometer output voltage from the motor to the servo drive.
- Cxx.042 Voltage Current Ratio *A**
Defines the scaling of the current output (MA) from the servo drive. This value can be obtained from the drive personality module.
- Cxx.043 Current -Torque Constant *A**
Defines the scaling of the drive current (MA) to motor torque. This value can be obtained from the motor nameplate.
- Cxx.044 Reduced Torque at Positive Stop**
Defines the amount of torque reduction when the axis has reached the hard stop during a feed to positive stop move (G75). The value is a percentage of nominal drive torque for analog drives.
- NOTE:** *If using DDS digital servo drives the Bipolar Torque Limit (S-0-0092) is preset at 400%, therefor the value must be divide by 4 before being entered into parameter CXX.044*
- Cxx.045 Maximum Feed to Positive Stop**
Defines the maximum feedrate for an axis during a feed to positive stop function.
- Cxx.053 Axis Defined in Right Hand Coordinate System as**
Defines the axis for interpolation purposes. Even though the axis will be called the value in Cxx.001, the NC program must know which axis to use in a particular application. (X, Y, Z= Main Axis; U, V, W= Parallel to main axis; A, B, C= Rotary around main axis)
- Cxx.060 Position Feedback Location *D**

Defines whether the motor or external feedback device is used for position control. The scaling parameters must be entered in the digital drive itself.

- Cxx.075** **Axis Designation 2**^{(2),(3),(4),(5)}
 Defines the symbolic identifier of the fictitious axis in the transformation functions 'facing' (G31) and 'cylinder machining' (G32).
- Cxx.076** **Direction for Polar Coord. Transformation**^{(2),(3),(4),(5)} *D
 Defines the axis direction in rotary mode 'C-Axis' while operation with Polar Coordinate Transformation is performed.
 0 direction in the positive range
 1 direction in the negative range
- Cxx.077** **Axis Droop Compensation** [linear axis]^{(2),(3),(4),(5)}
 Axis droop compensation can be used for error compensation with drooping axes or in oblique-angled coordinate systems. Setting this parameter to 'yes' activates Axis Droop Comp. parameters Cxx.078 thru Cxx.083.
- Cxx.078** **Number of associated axis (droop comp.)** [linear axis]^{(2),(3),(4),(5)}
 Specifies the axis number of the axis that is to be compensated. This parameter is present only if Cxx.077 is set to 'yes'.
- Cxx.079** **Number of Droop Compensation Points** [linear axis]^{(2),(3),(4),(5)}
 Defines the size of the compensation table. This parameter is present only if Cxx.077 is set to 'yes'.
- Cxx.080** **Distance Between Droop Compensation Points** [linear axis]^{(2),(3),(4),(5)}
 Defines the distance between two neighboring table values. The distance between droop compensation points is constant for the entire compensation range. This parameter is present only if Cxx.077 is set to 'yes'.
- Cxx.081** **Start Position of Droop Compensation** [linear axis]^{(2),(3),(4),(5)}
 Defines the distance between the first compensation point and machine zero. This parameter is present only if Cxx.077 is set to 'yes'.
- Cxx.082** **Unit of Droop Compensation Values** [linear axis]^{(2),(3),(4),(5)}
 Defines the unit of the compensation values in the table. This parameter is present only if Cxx.077 is set to 'yes'.
- Cxx.083** **List of Droop Compensation Values** [linear axis]^{(2),(3),(4),(5)}
 Enables the compensation values to be entered and modified in the table. This parameter is present only if Cxx.077 is set to 'yes'.
- Cxx.085** **Software Travel Limits Immediately Active**^{(3),(4),(5)}
 If this parameter is set to 'yes', the NC activates the software travel limits immediately after the control voltage has been switched on.

NOTE: *This parameter should only be set to 'yes' if absolute feedback motors are being used. Incremental feedback motors must be homed first to validate the software travel limits.*

Cxx.086 Window for Re-positioning and NC-block reset^{(3),(4),(5)}
 Enables the user to define for each feed axis (X,Y,Z,U,V,W,A,B,C) when the axis is considered as adjusted in the process of re-positioning or NC-block restart.

3.3.2 Main Spindle Axis Parameters

***A = Available for Analog Main Spindle Drives only.**

***D = Available for Digital Main Spindle Drives only.**

***O = Available for Open Loop Main Spindle Drives only.**

- Cxx.000 Associated Processes**
 Assigns the axis to one or more processes. The first process named will be the default process.
- Cxx.001 Axis Designation**
 Assigns a programming name for the spindle axis.
- Cxx.002 Mode of Operation *A *O**
 Selects the type of spindle that will be connected to the axis. The MUI selects relevant parameters needed for a particular mode. (Open Loop, Positioning)
- Cxx.004 Spindle Incremental Encoder Resolution *A**
 Defines the number of lines per revolution of the rotary encoder on the spindle motor.
- Cxx.009 Motor Speed at 9 Volts Command *A *O**
 Calibrates the D/A converter in the analog APR module for motor speed at 9 volts command. On Indramat TDAs, KDAs, RACs, this value is found in drive parameters A01 and A02.
- Cxx.010 Kv Factor *A**
 Defines the position loop gain for the spindle when the position loop control mode is active. This parameter is only active when the speed is over the value set in Cxx.035.
- Cxx.020 Jogging Speed⁽¹⁾**
Cxx.057 Jogging Speed^{(2),(3),(4),(5)}
 Specifies the speed at which the spindle will jog when using the AXC.JGPOS or AXC.JGNEG (SPS) Gateway signals.
- Cxx.021 Jogging Speed - Rapid⁽¹⁾**
Cxx.058 Jogging Speed - Rapid^{(2),(3),(4),(5)}
 Specifies the speed at which the spindle will jog when using the AXC.JGPOS or AXC.JGNEG (SPS) Gateway signals along with the AxC.RAPID Gateway signal.
- Cxx.022 Parametric Spindle Jogging Distance⁽¹⁾**
Cxx.059 Parametric Spindle Jogging Distance^{(2),(3),(4),(5)}
 Defines the incremental jogging distance when using the Parametric jogging mode via the PxxC.JOGMn Gateway signals.
- Cxx.024 Monitoring Window *A**
 Defines the percentage of position error allowed before a fault is issued. A value of zero disables the monitoring window.
- CAUTION:** *Disabling the monitoring window is not recommended. Damage to the machine and/or personnel may occur.*
- Cxx.035 Minimum Open-Loop Speed *A**

Specifies minimum speed that the spindle will be able to run in open loop mode. This value also specifies the changeover speed for the normal and low speed Kv-Factors Cxx.010. When the speed falls below the 'minimum open loop speed' parameter, the MT-CNC switches over to position control when a speed is specified.

- Cxx.036 Kv Factor for Low Speed Operation *A**
Defines the Kv factor position loop gain for the spindle when the position loop control mode is active. This parameter is only active when the speed is only under the value set in Cxx.035.
- Cxx.037 Motor Direction *A*0**
Defines the motor direction when the jog commands (+ / -) or M Functions (M03, M04, M19) are used. A value of "0" denotes a clockwise rotation of the motor shaft when looking at the front of the spindle motor, (M03) command.
- Cxx.038 Zero Point Offset *A**
Defines the reference position shift. The reference position can be shifted to another physical position.
- Cxx.039 Max. Operating Speed of Incr. Encoder *A**
Defines the manufacture specified maximum operating speed of the spindle encoder. If commanded spindle speed exceeds this value, the orient position of the spindle is lost and the position loop is opened.
- Cxx.040 Position Feedback Direction^{(2),(3),(4),(5)} *A**
Defines the counting direction of the spindle position feedback device. A value of "0" = (normal) mounted directly to motor, "1" = (reversed) mounted through a single stage geartrain.
- Cxx.041 Tach Voltage per 1000 RPM *A**
Only used when an Indramat servo drive is used as a spindle. It defines the tachometer voltage output from the motor to the servo drive.
- Cxx.047 Spindle Gearbox Revolutions - Input *A**
Specifies the value of input revolutions of a gearbox. This value MUST be a whole number.
- Cxx.048 Spindle Gearbox Revolutions - Output *A**
Specifies the value of output revolutions of a gearbox. This value MUST be a whole number.
- Cxx.049 Maximum Programmable Spindle Speed**
Specifies the maximum spindle speed allowed to be programmed.
- Cxx.050 Maximum Spindle Acceleration Rate**
Specifies a value that the spindle can be accelerated from standstill to max. velocity and back to standstill in the time required without overloading the spindle drive.
- Cxx.051 Orient Speed *A**
If the spindle is stopped and a M19 is programmed the axis will orient at this value. If the spindle is moving and a M19 is programmed, it will orient using the last programmed S word.
- Cxx.052 In Position Window**
Specifies the angle in degrees that a spindle will be considered "in position" by the MT-CNC. If the spindle is not within this position window at block end, the block will not be completed and NO ERROR WILL BE PRESENT.
- Cxx.056 Axis Number of Assigned Rotary Axis^{(2),(3),(4),(5)} *D**

Indramat's digital SERCOS main spindle drives can be used in 'C-axis mode' applications. If high power and torque is required in spindle mode, a separate geared (belts) spindle drive and motor is used in this mode. A geared (gear box) servo drive to the spindle shaft is than used during 'C-axis mode'.

- Cxx.060** **Position Feedback Location (digital drives)** ^{(2),(3),(4),(5)} *D
Defines whether the motor or external feedback device is used for position control. The scaling feedback parameters must be entered in the digital SERCOS drive.
- Cxx.061** **Number of Spindle Gear Ranges** ^{(2),(3),(4),(5)} *D
Defines the maximum number of gear ranges that can be selected for this axis in main spindle mode. The auxiliary M-functions M40, Mx41, Mx42, Mx43 and Mx44 (x=, ,1,2,3) can be used to select a different gear range for the main spindle mode.
- Cxx.062, Cxx.065** **Min. Spindle Speed, Gear Range (1...4)** ^{(2),(3),(4),(5)} *D
Cxx.068, Cxx.071
Cxx.063, Cxx.066 **Max. Spindle Speed, Gear Range (1...4)** ^{(2),(3),(4),(5)} *D
Cxx.069, Cxx.072
The NC employs the maximum and minimum spindle speed of the individual gear ranges as a database for automatic gear shifting and for the speed monitoring function. The minimum and maximum spindle speed of gear range (2...4) parameters are only displayed if a value >1 has been entered in (Cxx061).
- Cxx.064, Cxx.067** **Maximum Spindle Acceleration of Gear Range (1...4)** ^{(2),(3),(4),(5)} *D
Cxx.070, Cxx.073
In each gear range, the NC takes the maximum acceleration of the currently active gear range into account. The maximum spindle acceleration of gear range (2...4) parameters are only displayed if a value >1 has been entered in (Cxx061).
- Cxx.074** **Automatic Spindle Gear Range Selection** ^{(2),(3),(4),(5)} *D
Enables the NC to automatically select the required rear range of the 'Sx' spindle (x=, ,1,2,3) according to the specified speed values.
- Cxx.084** **Spindle Acceleration, Position Mode** ^{(3),(4),(5)}
The NC uses this parameter when it uses the spindle in position control mode. This is the case in rigid tapping (G63 and G64), and in primary spindle synchronization.

3.3.3 Digital Comb. Turret Axis / Main Spindle, C-Axis Parameters *D

- Cxx.000** **Associated Processes**
Assigns the turret to one or more processes. The first process named will be the default process.

- Cxx.001** **Axis Designation⁽¹⁾**
Cxx.054 **Axis Designation 1^{(2),(3),(4),(5)}**
 Assigns a programming name for the spindle axis. Axis designation 1 represents primary axis name used when programming in NC-programs.
- Cxx.006** **Programming Units / Table Revolution, Rotary Mode**
 Defines the number of graduations (units) per revolution of the moved turret. Typically this parameter is set to 360 degrees/revolution.
- Cxx.007** **Gearbox Revolutions, Input⁽¹⁾**
Gearbox Input Revolutions, Rotary Mode^{(2),(3),(4),(5)}
 Specifies the value of input revolutions of a gearbox. This value MUST be a whole number.
- Cxx.008** **Gearbox Revolutions, Output⁽¹⁾**
Gearbox Output Revolutions, Rotary Mode^{(2),(3),(4),(5)}
 Specifies the value of output revolutions of a gearbox. This value MUST be a whole number.
- Cxx.013** **Reference Position⁽¹⁾**
Reference Position, Rotary Mode^{(2),(3),(4),(5)}
 Specifies the value displayed and used by the NC program as the machine coordinate reference. After homing the turret, this value is loaded into the actual position register.
- Cxx.014** **Position Polarity (0/1)⁽¹⁾**
Position Polarity (0/1), Rotary Mode^{(2),(3),(4),(5)}
 Designates the polarity (sign) of the turret. A value of '0' signifies a normal coordinate polarity, while a 1 signifies a reverse polarity.
- Cxx.016** **Maximum Feedrate⁽¹⁾**
Maximum Feedrate, Rotary Mode^{(2),(3),(4),(5)}
 Sets the limited velocity of the turret. This is the velocity that the turret will try to achieve when a G0 is programmed without the F-word.
- Cxx.017** **Maximum Feedrate Change W/O Ramp⁽¹⁾**
Maximum Feedrate Change W/O Ramp, Rotary Mode^{(2),(3),(4),(5)}
 For a turret axis, this parameter should be set close to zero so that the Maximum Acceleration Rate (Cxx.018) will adjust the smooth operation of the turret's rotation.
- Cxx.018** **Maximum Acceleration Rate⁽¹⁾**
Maximum Acceleration Rate, Rotary Mode^{(2),(3),(4),(5)}
 This parameter is only active when the turret is positioning the tool, not when the spindle is turning the tool. The maximum spindle acceleration rate is programmed with (Cxx.050).
- Cxx.019** **Homing Speed⁽¹⁾**
Homing Speed, Rotary Mode^{(2),(3),(4),(5)}
 Specifies the speed the turret will home. This value will be used when a G74 is programmed without the F-word, or the axis is sent home manually using the AXC.HOME gateway signal.
- Cxx.020** **Jogging Speed⁽¹⁾**
Jogging Speed, Rotary Mode^{(2),(3),(4),(5)}
 Specifies the speed at which the spindle will jog when using the AXC.JGPOS or AXC.JGNEG (SPS) Gateway signals. This parameter is only used in spindle mode, because the turret can only be jogged from pocket to pocket using the parametric jog mode.
- Cxx.021** **Jogging Speed - Rapid⁽¹⁾**
Jogging Speed - Rapid, Rotary Mode^{(2),(3),(4),(5)}

Specifies the speed at which the spindle will jog when using the AXC.JGPOS or AXC.JGNEG (SPS) Gateway signals along with the AxC.RAPID Gateway signal. This parameter is only used in spindle mode, because the turret can only be jogged from pocket to pocket using the parametric jog mode.

- Cxx.022** **Parametric Jogging Distance⁽¹⁾**
Parametric Jogging Distance, Rotary Mode^{(2),(3),(4),(5)}
Defines the incremental jogging distance when using the Parametric jogging mode via the PxxC.JOGMn Gateway signals. The value entered in this parameter should be the distance between turret positions or magazine pockets.
- Cxx.023** **In Position Window⁽¹⁾**
In Position Window, Rotary Mode^{(2),(3),(4),(5)}
Sets the maximum position loop deviation allowed. If axis is not within this value at block end, the block will not be completed, and NO ERROR WILL BE PRESENT.
- Cxx.025** **Waypoints 0 - 7^{(2),(3),(4),(5)}**
Cxx.032 Are used to set actual axis positions to be monitored by the SPS program. If the absolute actual position is equal to or greater than the value set in one of the parameters 25 - 32, the associated SPS signal AxxS.WPn (n = 0 - 7) will become '1'.
- Cxx.037** **Motor Direction (0/1)⁽¹⁾**
Motor Direction, Main Spindle Mode^{(2),(3),(4),(5)}
Defines the motor direction when the jog commands (+ / -) or M Functions (M03, M04, M19) are used. A value of "0" denotes a clockwise rotation of the motor shaft when looking at the front of the spindle motor, (M03) command.
- Cxx.038** **Zero Point Offset⁽¹⁾**
Reference Offset, Main Spindle Mode^{(2),(3),(4),(5)}
Defines the reference position shift. The reference position can be shifted to another physical position.
- Cxx.047** **Gearbox Revolutions, Input⁽¹⁾**
Gearbox Input Revolutions, Main Spdl. Mode^{(2),(3),(4),(5)}
Specifies the value of input revolutions of a gearbox. This value MUST be a whole number.
- Cxx.048** **Gearbox Revolutions, Output⁽¹⁾**
Gearbox Output Revolutions, Main Spdl. Mode^{(2),(3),(4),(5)}
Specifies the value of output revolutions of a gearbox. This value MUST be a whole number.
- Cxx.049** **Maximum Programmable Spindle Speed⁽¹⁾**
Maximum Programmable Spindle Speed, Main Spdl. Mode^{(2),(3),(4),(5)}
Specifies the maximum spindle speed allowed to be programmed.
- Cxx.050** **Maximum Spindle Acceleration Rate⁽¹⁾**
Maximum Spindle Acceleration, Main Spindle Mode^{(2),(3),(4),(5)}
Specifies a value that the spindle can be accelerated from standstill to max. velocity and back to standstill in the time required without overloading the spindle drive.

- Cxx.051 Orient Speed, Main Spindle Mode**
If the spindle is stopped and a M19 is programmed the axis will orient at this value. If the spindle is moving and a M19 is programmed, it will orient using the last programmed S word.
- Cxx.052 In Position Window⁽¹⁾**
In Position Window, Main Spindle Mode^{(2),(3),(4),(5)}
Sets the maximum position loop deviation allowed. If axis is not within this value at block end, the block will not be completed, and NO ERROR WILL BE PRESENT.
- Cxx.053 Rotary Axis Defined in COORD. System as...^{(2),(3),(4),(5)}**
Defines the axis for interpolation purposes. Even though the axis will be called the value in Cxx.001, the NC program must know which axis to use in a particular application. (X, Y, Z= Main Axis; U, V, W= Parallel to main axis; A, B, C= Rotary around main axis)
- Cxx.055 Parameter Set in Drive Used for Rotary Mode^{(2),(3),(4),(5)}**
Defines the parameter set in the spindle drive that the NC activates in rotary axis mode. The default parameter set used by the NC is set '0'.
- Cxx.057 Jogging Speed, Main Spindle Mode^{(2),(3),(4),(5)}**
Specifies the speed at which the spindle will jog when using the AXC.JGPOS or AXC.JGNEG (SPS) Gateway signals.
- Cxx.058 Jogging Speed - Rapid, Main Spindle Mode^{(2),(3),(4),(5)}**
Specifies the speed at which the spindle will jog when using the AXC.JGPOS or AXC.JGNEG (SPS) Gateway signals along with the AxC.RAPID Gateway signal.
- Cxx.059 Jogging Spindle Position, Main Spindle Mode^{(2),(3),(4),(5)}**
Defines the incremental jogging distance when using the Parametric jogging mode via the PxxC.JOGMn Gateway signals.
- Cxx.060 Position Feedback Location (digital drives)^{(2),(3),(4),(5)}**
Defines whether the motor or external feedback device is used for position control. The scaling feedback parameters must be entered in the digital SERCOS drive.
- Cxx.061 Number of Spindle Gear Ranges^{(2),(3),(4),(5)}**
Defines the maximum number of gear ranges that can be selected for this axis in main spindle mode. The auxiliary M-functions M40, Mx41, Mx42, Mx43 and Mx44 (x=, ,1,2,3) can be used to select a different gear range for the main spindle mode.
- Cxx.062, Cxx.065 Min. Spindle Speed, Gear Range (1...4)^{(2),(3),(4),(5)}**
Cxx.068, Cxx.071
Cxx.063, Cxx.066 Max. Spindle Speed, Gear Range (1...4)^{(2),(3),(4),(5)}
Cxx.069, Cxx.072
The NC employs the maximum and minimum spindle speed of the individual gear ranges as a database for automatic gear shifting and for the speed monitoring function. The minimum and maximum spindle speed of gear range (2...4) parameters are only displayed if a value >1 has been entered in (Cxx061).

- Cxx.064, Cxx.067** **Maximum Spindle Acceleration of Gear Range (1...4)**^{(2),(3),(4),(5)}
Cxx.070, Cxx.073
- In each gear range, the NC takes the maximum acceleration of the currently active gear range into account. The maximum spindle acceleration of gear range (2...4) parameters are only displayed if a value >1 has been entered in (Cxx061).
- Cxx.074** **Automatic Spindle Gear Range Selection**^{(2),(3),(4),(5)}
Enables the NC to automatically select the required rear range of the 'Sx' spindle (x=, ,1,2,3) according to the specified speed values.
- Cxx.075** **Axis Designation 2, Rotary Axis Mode**^{(2),(3),(4),(5)}
Defines the symbolic identifier of the fictitious axis in the transformation functions 'facing' (G31) and 'cylinder machining' (G32).
- Cxx.076** **Direction for Polar Coord. Transformation**^{(2),(3),(4),(5)}
Defines the axis direction in rotary mode 'C-Axis' while operation with Polar Coordinate Transformation is performed.
0 direction in the positive range
1 direction in the negative range
- Cxx.084** **Spindle Acceleration, Position Mode**^{(3),(4),(5)}
The NC uses this parameter when it uses the spindle in position control mode. This is the case in rigid tapping (G63 and G64), and in primary spindle synchronization.
- Cxx.086** **Window for Re-positioning and NC-block reset**^{(3),(4),(5)}
Enables the user to define for each feed axis (X,Y,Z,U,V,W,A,B,C) when the axis is considered as adjusted in the process of re-positioning or NC-block restart.

4. NC Programming

4.1 NC Program Structure

- The NC program must be generated as a clean ASCII file (no control characters e.g., character format, etc.).
- The NC program can consist of a maximum of 2000 lines⁽¹⁾.
- The NC program can consist of a maximum of 10000 lines^{(4),(5)}.
- NC program 0 (zero) for each process is used for NC cycles subroutines.
- An NC program package contains all of the necessary part and/or process NC programs and tool setup lists for each specified process. NC programs 1...99 and NC cycles (program 0) can be contained for each process.
- Each NC program (1...99) may contain an advance and reverse program and is assigned to one of the available processes (0...6).
- Up to 12 NC program package slots are immediately available in the MUI. All packages can be archived on either the Hard Disk, Floppy Disk, or a Host computer.^{(1),(2),(3)}
- Up to 99 NC program package slots are immediately available in the MUI. All packages can be archived on either the Hard Disk, Floppy Disk, or a Host computer.^{(1),(2),(3)}

4.2 NC Memory

The MT-CNC has 2 NC memories to store user data. Each can store the data of one NC program package. They are referred to as NC-memory A and NC-memory B. The user can activate either memory with functions of the MUI software or via the programmable controller program.

4.3 Advance Program

The advance program starts with the label with the NC-block number "N0000", usually contains the label ".START", and ends with the NC-block that contains the NC-command BST (branch to .START and stop) or the NC-command RET, program end with reset.

4.4 Reverse Program

The reverse program starts at the NC-block that contains the label ".HOME" and ends with NC-command BST, usually to the label that marks the beginning of the advance program ".START". The reverse program can be programmed in program '0' or '99', so it can be used as a subroutine.

A reverse program may also be started at labels which have been specified as reverse vectors. Reverse vectors are programmed in the advance program (see Reverse Vector) and are executed first when starting a reverse program. The CNC will continue program execution at the label of the last active reverse vector.

4.5 Part Program

NC programs of process 0 are referred to as Part Programs or Management Programs. Special management NC-commands can be programmed in process 0 only. In addition to the special management NC-commands, the part program has all of the capabilities of a process program.

4.6 Process Program

NC programs of process (1...6) are referred to as Process Programs. Process Programs contain the actual motion or machining program for the process.

4.7 NC-Block Elements

An NC-block is composed of the following elements.

- block number
- branch label
- NC-command (s)
 - command word
 - command operand (s)
- remark
- hint
- comment in source program.

4.7.1 NC-Block Numbers

N0000 NC-Block Number

Syntax: Nxxxx :x = 0000...9999

Each NC-block begins with the letter N. The block numbers are automatically generated by the MUI when entered, and regenerated when a block is deleted.

4.7.2 NC-block which can be Skipped

Blocks in a part program which are not to be processed each time the program is executed must be identified by a '/' at the beginning of the NC-block. These NC blocks are only processed when the user activates the skip function via the Gateway signal (PxC.BLSKP).

4.7.3 NC-Labels

.LABEL Branch (Jump) Label

Syntax: .xxxxxx :x = Alphanumeric characters

Identifies an NC-block that is used as the target NC-block for a program branch via a branch NC-command or a call of a subroutine. Label begins with a "." and consists of at least one and up to 6 non case sensitive characters. The branch to address (destination label) can be in the same NC-program or located in program '99' or '0'.

NOTE: *Standard NC cycles start with the label .*xxxxx.*

4.7.4 NC Message for MUI

A NC-block may contain information which will be displayed in menu item 7 of the MUI. The message is displayed after the NC-block with the message is executed. The message can consist of a maximum of 48 characters and is displayed until another message is activated or the NC program start is initiated for the process. The message must be enclosed in square brackets [...].

4.7.5 NC Message for SOT

A NC-block may contain information which will be displayed on the SOT. The message is displayed after the execution of the NC-block in which it is programmed. The message can consist of a maximum of 32 characters and is displayed until another message is activated or the NC program start is initiated for the process. The message must be enclosed in parentheses (...).

4.7.6 NC-Block Comment

An NC-block may contain a comment. The comment's beginning is marked with a semicolon ";". All characters following the semicolon to the block end, are interpreted as a comment. Comments are only present in the source program, and are not downloaded to the controller memory.

4.8 G-Codes

- *D = G-Code is activated at MT-CNC power on (Default) and is automatically activated at NC program end (BST,RET) or control reset.
- *R = G-Code remains active until opposing G-Code is programmed, NC program end (BST,RET), or control reset.
- *B = NC-block active
- *P = Parameter influenced
- [...] = G-Codes which cancel the active G-Code.

The preparatory functions are organized in different groups. Modal active G-codes influence G-codes of the same group only. The following groups for preparatory functions exist for the MT-CNC:

Preparatory function (G-code)	Group	Active	Description
G00, G01, G02, G03	1	modal	Type of Interpolation
G17, G18, G19, G20	2	modal	Plane selection
G40, G41, G42	3	modal	Tool path compensation
G52 . . G59	4	modal	Workpiece zero points and offsets
G15, G16	5	modal	Diameter programming
G90, G91	6	modal	Input dimensioning style, absolute/incremental
G65, G94, G95	7	modal	Feed programming
G96, G97, G66	8	modal	Spindle speed programming
G70, G71	9	modal	Measuring Units (inch/metric)
G43, G44	10	modal	Transition element of tool path correction
G61, G62	11	modal	Block transition selection (with/without lag finishing)
G98, G99	12	modal	Feed definition for tool path correction
G47, G48, G49	13	modal	Tool length correction
G08, G09	14	modal	Contouring mode (optimize feedrate changes)
G06, G07	15	modal	Interpolation method
G04	16	block	Dwell time
G33	16	block	Thread cutting
G50, G51	16	block	Programmable zero offsets
G63, G64	16	block	Rigid tapping
G74	16	block	Home axes

Preparatory function (G-code)	Group	Active	Description
G75, G76	16	block	Feed to positive stop
G77	16	block	NC-block restart / re-positioning
G92	16	block	Spindle speed limit
G93	16	block	Input feedrate as inverse time value
G30, G31, G32	17	modal	Cartesian->polar coordinate transformation
G72, G73	18	modal	Mirror Axes
G78, G79	19	modal	Scaling

Note: *It is possible to program more than one G-code in the same NC-block, provided they are NOT of the same group and are compatible with each other. Leading zeros are omitted, i.e., G00 may be programmed as G0 or G00.*

NC-block active G-codes can be read into a NC-variable only if programmed in the NC-block together with the block active G-code. A value of '-1' will be read otherwise.

G0 Linear Interpolation with Rapid Traverse*R [G1,G2,G3]

Syntax:

G00

All axes programmed in the NC-block with a G0 perform a linear interpolated move using maximum velocity Bxx.005, if no feedrate is programmed. The feedrate remains active until another feedrate is programmed.

Note: *In software releases 15Vxx and later the Gateway signal (PxC.ROVRD) controls the velocity override when G00 is programmed for linear and rotary axis.*

G1 Linear Interpolation*R [G0,G2,G3]

Syntax:

G01 F value

Defines a linear interpolated move using a programmed feedrate. All programmed axes start and complete their movements simultaneously.

**G2 Circular Interpolation (CW/CCW)*R [G0,G1]
 G3**

Syntax:

G02 <endpoint><interpolation

parameters [I,J,K] or <radius [R]>

The coordinates following the G2/G3 define the end positions of the main axes (X,Y,Z), and the interpolation parameters (I,J,K) specify the center of the circle in the active plane. The circular movement to the arc endpoint is performed clockwise if G2 is active and counterclockwise if G3 is active.

Circles may also be programmed by specifying the end point and radius value following the letter 'R'.

G4 Dwell Time*B

Syntax:

G04 F<time in seconds>

Specifies a programmed dwell time using an F word. The value following the letter F specifies the time duration in seconds. (0.01...600 seconds). The F-word programmed together with 'G04' can be programmed with three places to the left of the decimal point and two places to the right. Only 'M', 'S', and 'Q' functions can be programmed in the same NC-block.

G6 Positioning with Minimized Lag*R[G7]

Syntax:

G06

Activates an algorithm for positioning with minimized lag (following error). Path velocity is reduced to 0 before NC-block is completed. A following error only occurs within the '2ms' limits of the interpolation clock.

- G7 Positioning with Lag*D[G6]**
Syntax: **G07**
Cancels the minimized lag interpolation G6. (BST,RET). NC-block transitions which are not tangential will be rounded.
- G8 Contouring Mode (Acceleration)*R[G9]**
Syntax: **G08**
The path velocity is not reduced to 0 before the execution of the next NC-block. The velocity is reduced to a value that is below the value in parameter Cxx.017. This G-Code is used to adjust the final velocity at the end of the NC-block to ensure that the transition to the next NC-block occurs at the highest possible velocity.
- G9 Contouring Mode (Deceleration)*D[G8]**
Syntax: **G09**
Reduces the velocity at the NC-block end to a value that would allow the axis to be stopped with the maximum feedrate change, specified on parameter Cxx.017.
- G15 Radius Programming*D[G16]**
Syntax: **G15**
For NC programs that involve turning applications, the programmed values will represent the radius of the part.
- G16 Diameter Programming*R[G15]**
Syntax: **G16**
For NC programs that involve turning applications, the programmed values will represent the diameter of the part.
- G17 XY Plane *P**
Syntax: **G17**
The Z-axis is perpendicular to the XY plane. Process parameter Bxx.004 specifies the default plane activated on MT-CNC power on.
- G18 ZX Plane*P**
Syntax: **G18**
The Y-axis is perpendicular to the ZX plane. Process parameter Bxx.004 specifies the default plane activated on MT-CNC power on.
- G19 YZ Plane*P**
Syntax: **G19**
The X-axis is perpendicular to the YZ plane. Process parameter Bxx.004 specifies the default plane activated on MT-CNC power on.
- G20 Free Plane Selection^{(4),(5)}**
Syntax: **G20**
Allows the user to select the axes that build the coordinate system. The user must sequentially specify the axes that will have the axis meaning X, Y and Z. The first and second axis define the active plane and the third axis defines the axis perpendicular to the plane. This allows rotary axis and secondary axis to be considered in circular interpolation, tool path compensation, path velocity calculation, constant surface speed (CSS) and diameter programming.

- G30 Coordinate Transformation, Cancel*D*R[G31]^{(2),(3),(4),(5)}**
Syntax: **G30**
Cancels an active coordinate transformation (G31). The fictive Cartesian coordinate system is canceled, and the coordinate system which is defined in the process parameter Bxx.004 is set.. The fictive axes are canceled and therefore must not be programmed any more.
- G31 Coordinate Transformation, On*R[G30]^{(2),(3),(4),(5)}**
Syntax: **G31**
Enables a fictive Cartesian coordinate system. The defined fictive linear axes are used in the interpolation instead of the assigned real main axes. The path feedrate with the transformation function must be specific, as with milling, as a relative speed between the tool and the workpiece using the F-value. The programmed path feedrate is reduced in such a way that the maximum rpm of the rotary axis is not exceeded. This is especially true with movement near the center of rotation.
- G32 Cylinder Surface Interpolation^{(4),(5)}**
The NC generates straight lines and circles on a cylinder surface as defined in the NC-program when using Cylinder Interpolation (G32). The contour that should be generated on the cylinder surface can be programmed in the plane of the rolled out cylinder surface. The associated rotary axis receives the functionality of a linear main axis providing the user with all functions of a linear main axis during the cylinder interpolation.
- G33 Thread Cutting*B**
Syntax: **G33<end point><lead><starting angle>**
Is used to machine cylindrical, face and taper threads with a constant pitch. The thread lead is entered in addresses 'T', 'J' and 'K' however, no more than one interpolation parameter can be programmed in a single thread NC-block. The interpolation parameters 'T', 'J' and 'K' are programmed as unsigned incremental values. The interpolation parameters 'T', 'J' and 'K' are assigned to axes 'X', 'Y' and 'Z'. The thread starting angle may be programmed as a value from 0 to 360 degrees. in address 'P'.
- G36 Shortest Distance, with Modulo Calculation^{(4),(5)}**
When the axis is programmed as modulo (endless rotating) rotary axis, the axis will move the shortest distance to the programmed value regardless of direction. The selected rotary mode is valid for all rotary axes in a process. Independent of the selected mode the NC considers rotary axes as always like wound up linear axes when 'G91' is active. The process default rotary mode G-code can be defined in process parameter (Bxx.056 Rotary Axes Mode Default).
- G37 Positive Direction, with Modulo Calculation^{(4),(5)}**
When the axis is programmed as modulo (endless rotating) rotary axis, the axis will only move in the positive direction to the programmed value. The selected rotary mode is valid for all rotary axes in a process. Independent of the selected mode the NC considers rotary axes as always like wound up linear axes when 'G91' is active. The process default rotary mode G-code can be defined in process parameter (Bxx.056 Rotary Axes Mode Default).
- G38 Negative Direction, with Modulo Calculation^{(4),(5)}**
When the axis is programmed as modulo (endless rotating) rotary axis, the axis will only move in the negative direction to the programmed value. The selected rotary mode is valid for all rotary axes in a process. Independent of the selected mode the NC considers rotary axes as always like wound up linear axes when 'G91' is active. The process default rotary mode G-code can be defined in process parameter (Bxx.056 Rotary Axes Mode Default).

- G40 Tool Path Compensation, Cancel *D[G41,G42]**
Syntax: **G40**
Cancels and deactivates the tool path compensation. The tool center moves to the programmed path if no tool path compensation is active.
- G41 Tool Path Comp. Left of Workpiece Cont.*R[G40,G42]**
Syntax: **G41**
Activates tool path compensation to the left of the workpiece. The axis moves on a path which is parallel to the contour and which is offset from the contour by the tool radius value.
- G42 Tool Path Comp. Right of Workpiece Cont.*R[G40,G42]**
Syntax: **G42**
Activates tool path compensation to the right of the workpiece. The axis moves on a path which is parallel to the contour and which is offset from the contour by the tool radius value.
- G43 Contour Transition, Arc*D[G44]**
Syntax: **G43**
An arc is inserted as the contour transition element in the outside corners if G43 is programmed and (G41/G42) is activated.
- G44 Contour Transition, Chamfer*R[G43]**
Syntax: **G44**
A chamfer is inserted as the contour transition element in the outside corners with a transition angle greater than 90° if G44 is programmed and (G41/G42) is activated. With outside corners having a transition angle greater than or equal to 90°, the corrected transition point is defined as the intersection of the lines parallel to the contour.
- G47 Tool Length Correction, Cancel*D[G48,G49]**
Syntax: **G47**
If active no tool length correction is performed. The programmed positions for the tool main axis refer to the spindle flange.
- G48 Tool Length Correction, Positive *R[G47,G49]**
Syntax: **G48**
CNC compensates the tool lengths L1, L2 (depending on the tool correction type) in positive direction of the main axes that form the active plane and the length L3 in the positive direction of the tool axis.
- G49 Tool Length Correction, Negative *R[G47,G48]**
Syntax: **G49**
CNC compensates the tool lengths L1, L2 (depending on the tool correction type) in negative direction of the main axes that form the active plane and the length L3 in the negative direction of the tool axis.
- G50 Absolute Zero Point Offset*B[G52,G53]**
Syntax: **G50 <axis>**
Shifts the Cutting Zero Point (absolute) relative to the last selected Workpiece Zero Point. The offset values must be specified with the corresponding axis designation in the NC-block together with G50.

G51 Increment Zero Offset *B[G52,G53]

Syntax: **G51 <axis>**

Shifts the Cutting Zero Point (incremental) relative to the previous Cutting Zero Point. The offset values must be defined with an axis designations and programmed in the same NC-block together with G51.

G52 Programmed Workpiece Zero Point*R[G53]

Syntax: **G52 <axis>**

Assigns the specified axis coordinate value (s) to the current axis position (s). This defines the new Workpiece Zero Point relative to the current axis position (s).

G53 Machine Zero Point *D[G50,G51,G52,G54-G59]

Syntax: **G53**

Moves the Workpiece Zero Point to the Machine Zero Point, all active Zero Point Offsets and Workpiece Zero Points will be canceled.

G54...G59 Workpiece Zero Point *R[G52,G53]

Syntax: **G54-G59**

Activates the coordinate offsets for one of up to six defined Workpiece Zero Points. The coordinate offset **G59** values (retentive) can be entered via the MUI or SOT.

G61 Block Transition without Lag Finishing *R[G62]

Syntax: **G61**

The NC-block transition is executed by the CNC if the **actual position** of the programmed axes have reached their programmed end position. The allowed position deviation for block transition is programmed in parameter Cxx.023 (In position window) for each axis.

G62 Block Transition with Lag Present *D[G61]

Syntax: **G62**

The NC-block transition is executed by the CNC if the **command values** of the programmed axes have reached their programmed end position. The CNC does not wait for all axes to reach their exact actual end position to execute a block transition.

G63 Rigid Tapping with Spindle Off at NC-Block End *B

Syntax: **G63 <end point> <feed per spindle revolution[F]>**

Selects an interpolating movement between the programmed main spindle and the main servo axes (X,Y,Z). At NC block end the spindle and feed axis decelerate and stop together. A spindle with a closed position loop is required.

G64 Rigid Tapping with Spindle On at NC Block End*B

Syntax: **G64 <end point> <feed per spindle revolution[F]>**

Selects an interpolating movement between the programmed main spindle and the main servo axes (X,Y,Z). At NC-block end the spindle/feed axis interpolation is deactivated at the beginning of the deceleration phase. A spindle with a closed position loop is required.

G65 Floating Tapping*B

Syntax: **G65 <feed per spindle revolution[F]>**

The feed axis follows the spindle with following error. This following error is not compensated for when changing spindle direction. This function can only be used with a floating tap. A spindle with a closed position loop is required (tension/compression).

G66 Constant Wheel Surface Speed^{(4),(5)}

This grinding specific function keeps the wheel's surface speed (programmed in [m/s] or [feet/sec]) constant, even though the wheel diameter defined in the tool data changes. The spindle speed driving the wheel is immediately adjusted when the user modifies the data that specifies the wheel diameter in the Active Tool data set via the MUI or SPS program.

G70 Inch Programming Input *P

Syntax: **G70**

The CNC converts the dimensions following the G70 to (mm) if the basic unit parameter Bxx.001 is (mm). The dimensions are not converted if the basic unit is (inch).

G71 Metric (mm) programming Input *P

Syntax: **G71**

The CNC converts the dimensions following the G71 to (inch) if the basic unit parameter Bxx.001 is (inch). The dimensions are not converted if the basic unit is (mm).

G72 Mirror Coordinate Axes^{(2),(3),(4),(5)}

G73 Syntax: G72, G73

The programmable mirror function allows the mirroring of any coordinate axes within the NC-program. The mirror function applied to a coordinate axis causes the original contour to be machined mirrored in the same size to the other side of the mirror axis. Preparatory functions in the NC-program are used to activate and cancel the mirror function. The mirror function can be activated via G73 and remains modal active until it is canceled via G72 or at the NC-program end (RET, M2, M30 or BST). G72 sets all mirror axes into their base position.

G74 Homing Cycle

Syntax: **G74 <axis name> <coordinate value> <feed>**

All axes programmed in the NC-block will simultaneously (not interpolating) move to their home position. If axis is being homed for the first time the feedrate is the value programmed in parameter Cxx.019 (analog) and S-0-0041 (digital). If axis has been homed before the CNC uses the currently active feedrate. the coordinate values of the programmed axes in a 'G74 NC-block' must be '0'.

G75 Feed to Positive Stop*R[G76]

Syntax: **G75 <axis name> <coordinate value> <feed>**

During a programmed axis movement a positive stop is encountered. The analog drives can reach their max. torque, and the digital drives are limited to 150% of their nominal torque when against a the positive stop. The axis parameters Cxx.044 and Cxx.045 are used with this function. Parameters P-7-3577 and P-7-3578 can be used directly in the NC program to limit the torque TO and torque AT positive stop.

G76 Feed to Positive Stop , Cancel*D[G75]

Syntax: **G76**

Cancels the feed to positive stop mode of all process axes. The current position values of the axes are then used as their position command value.

G77 NC-Block Restart/Re-Positioning^{(2),(3),(4),(5)}

Syntax: **G77 <axis name> <coordinate value[0]>**

G77 directs the CNC to establish for the programmed axis the original state. The sequence and the velocity that is used to establish the state prior to the NC-program interruption is contained in the NC-subroutine 'REPOS (similar to the homing NC-subroutine) that can be executed in the operating mode automatic, semi automatic and manual.

```
;NC-cycle for Re-positioning and Re-start  
;  
N0035                            .REPOS
```

N0036 location	G77 W0 ;position tool turret (servo axis) on last selected
N0037	G77 S1 0 S2 0 F5000 ;position spindle S1 and spindle S2 to their old position
N0038	MRY ;wait for tool turret positioning complete
N0039	G77 V0 U0 F2000 ;position secondary axes
N0040	RTS

G78 Scaling ^{(2),(3),(4),(5)}

G79 The scaling factor can be used to modify dimensions to be moved of all machine axes via programmable scaling factors. The activation and deactivation of the scaling function can be programmed via G-functions in the NC-program.

Scaling can be switched on via G79 and remains modal active until canceled via G78 or at the NC-program (RET, M2, M30 or BST) or via Control Reset. G78 cancels the scaling of all axes.

Syntax: **G78 ;Scaling, OFF (all axes)**
G79 <Axis designation><scaling factor> ;Scaling, ON

The following values are converted during scaling:

- Axis coordinates
- Interpolation parameter (I, J, K)
- Radius (R)
- Programmed zero offsets G50 and G51
- Thread lead (pitch, I, J, K)
- Nominal radius (RX, RY, and RZ)

The zero point offsets G54 ... G59 and the adjustable offset are not being scaled. The programmable zero points G50 and G51 are being scaled when programmed after selecting the scaling function.

G90 Input Data as Absolute Dimensions*D[G91]

Syntax: **G90**

Specifies that all dimensions which follow are absolute coordinate values in the actual zero point system.

G91 Input Data as Incremental Values *R[G90]

Syntax: **G91**

Defines all following dimensions as the difference between the current NC-block start and the end position. The distance does not reference programmed distances to the currently active zero point.

G92 Spindle Speed Limit *B

Syntax: **G92 S<upper spindle speed limit>**

Can be used in conjunction with G96 (constant surface speed) to limit the maximum spindle speed for the selected spindle (NC-command SPF).

G93 Input Feedrate as Inverse Time Value *B

Syntax: **G93 F<time in seconds>**

Defines the F-word value as time (in seconds) in which the programmed movement must be performed. The CNC calculates the appropriate path velocity for the movement.

G94 Input Feedrate in Inches (mm) per Minute *D

Syntax: **G94**

Defines the value programmed in the F-word as “Feed per Minute”. Units per revolution are defined in the axis parameter Cxx.006 if the axis is specified as rotary axis. Depending on the settings in the process process parameters (Bxx.041 Spindle Speed Programming Default) ‘G94’ may be the power-on default.

G95 Input Feedrate in Inches (mm) per Spindle Rev.

Syntax: **G95 F<feed per revolution>**

The path velocity depends on the actual spindle speed of the selected spindle (NC-command SPF). The commanded spindle speed is used if no actual spindle speed (feedback) is used.

G96 Constant Surface Speed *R[G97]

Syntax: **G96 S<constant surface speed in m/min>**

The speed of the selected spindle is inversely proportional to the coordinate value of the main axis X in the machine coordinate system. The spindle speed will increase if the X coordinate value decreases. Spindle speed is limited to the value in parameter Cxx.049. Depending on the process parameter (Bxx.041 Spindle Speed Programming Default) 'G96' may be the power-on default.

G97 Spindle Speed in RPM *D[G96]

Syntax: **G97**

The S-word value of the selected spindle is interpreted as "revolutions per minute". Depending on the process parameter (Bxx.041 Spindle Speed Programming Default) 'G97' may be the power-on default.

G98 Constant Feed on Tool Center Line *D[G99]

Syntax: **G98**

Works in conjunction with the active tool path compensation (G41,G42). There is no path velocity correction in the case of arc contour elements. The programmed path velocity is calculated for the tool center and not the actual arc contour at the tool edge. In the case of convex arcs (outside circle) this results in a reduction of the path feedrate at the contour; with concave arcs (inside circle) it results in an increase.

G99 Constant Feed at the Contour

Syntax: **G99**

Works in conjunction with the active tool path compensation (G41,G42). There is a path velocity correction in the case of arc contour elements. The programmed path velocity is equal to the contour velocity and not the tool center velocity. In the case of convex arcs (outside circle) this results in a reduction of the path feedrate at the contour; with concave arcs (inside circle) it results in an increase.

4.9 Auxiliary Functions

15VRS and Below

Auxiliary NC-functions are passed on by the CNC to the SPS at the end of NC-block execution. The CNC stops NC-block execution until the auxiliary NC-function is acknowledged by the SPS:

NOTE: *The status of auxiliary functions is set to "1" in the SPS at the end of NC block execution. The CNC stops execution for the process until the SPS acknowledges (resets) the auxiliary function.*

16VRS and above

The following describes how the user can define auxiliary M and Q-function output behavior to the SPS:

- At the end of motion (NC-block)
- At the beginning of motion (NC-block)
- No output

The following describes how the user can define auxiliary M and Q-function acknowledgment from the SPS:

- At the end of motion (NC-block)
- At the beginning of motion (NC-block)
- At a later time (quick auxiliary functions) in the System and Process parameters

Auxiliary M and Q-functions can be programmed with an optional appended 'Q' (for quick i.e.,MQ806) as quick auxiliary functions if defined in the System and Process parameters as acknowledged 'at a later time'. Quick aux. functions can be checked later in the NC-program for acknowledgment by appending the letter 'W' (for wait, i.e., MW806). The NC stops NC-program execution until the 'quick' aux. function is acknowledged and continues if already acknowledged.

Spindles can be asynchronously accelerated/decelerated while continuing NC-block execution if the OEM defined acknowledgment of the spindle control M-functions (Mj03, Mj04 and Mj05; j=(1,2,3) as 'at a later time' and the end user programs them with appended 'Q' (MQj03, MQj04 and MQj05).

Types of auxiliary NC-functions available

- S-functions: (S0...S9999) for each process^{(1),(2),(3)}
- S-functions: (S0.00...S99999.99) for each process^{(4),(5)}
- M-functions: (M0...M399) for each process
- M-functions: (M0... M999)^{(4),(5)} for each process
- Q-functions: (Q0...Q9999) for each process
- T-functions: (T0 to T9999999)^{(4),(5)}
- E-functions (E0...E9)^{(4),(5)}

S Auxiliary S-function

Syntax: Sx :x = 0...9999

Defined as auxiliary NC-function if **NO** main spindle is specified for that process. The processes that control a main spindle are specified in the spindle axis parameter Cxx.000. Only one S-function per NC-block, and integer numbers only.

M Auxiliary M-function

Syntax: Mx :x = 0...399

M-function	Group	Active	User Specific	Description
M000, M001, M002, M030	1	modal	yes	Program control
M3 M4 M5 M13 M14 M103 M104 M105 M113 M114	2	modal	no	Spindle 1 (S, S1) commands
M203 M204 M205 M213 M214	3	modal	no	Spindle 2 (S2) commands
M303 M304 M305 M313 M314	4	modal	no	Spindle 3 (S3) commands
M7 M8 M9 M107 M108 M109	5	modal	yes	coolant spindle 1 (S, S1)
M207 M208 M209	6	modal	yes	coolant spindle 2 (S2)
M307 M308 M309	7	modal	yes	coolant spindle 3 (S3)
M10 M11 M110 M111	8	modal	yes	clp/uclp. spindle 1 (S, S1)
M210 M211	9	modal	yes	clp/uclp. spindle 2 (S2)
M310 M311	10	modal	yes	clp/uclp. spindle 3 (S3)
M40 M41 M42 M43 M44 M140 M141 M142 M143 M144 M145	11	modal	no if Cxx.74 = yes	gear range selection spindle 1 (S, S1)

M-function	Group	Active	User Specific	Description
M240 M241 M242 M243 M244 M245	12	modal	no if Cxx.74 = yes	gear range selection spindle 2 (S2)
M340 M341 M342 M343 M344 M345	13	modal	no if Cxx.74 = yes	gear range selection spindle 3 (S3)
M46 M47	14	modal	yes	spindle (1-3) override
M48 M49	15	modal	yes	feedrate override
M19 M119 M219 M319	16	block	no	position spindle 1, 2, 3 (S, S1), (S2), (S3)
All unlisted M-functions	16	block	yes	user specific usage

- Up to 4 auxiliary M functions, each of a different functional group, can be programmed in a single NC block if no auxiliary S or Q function is programmed.
- No more than four 'M-functions' can be programmed in a single NC-Block.
- 'M-functions' overwrite one another.
- 'M-functions' are always output to the SPS at the end of the NC-block.

(M00,M001,M002,M030)

NC-program stop/optional and stop w/ return to beginning.

(M3/M103,M203,M303)

Activates a clockwise rotation for spindles S/S1,S2,S3 respectively.

(M4/M104,M204,M304)

Activates a counterclockwise rotation for spindles S/S1,S2,S3 respectively.

(M5/M105,M205,M305)

Activates a spindle stop condition for spindles S/S1,S2,S3 respectively.

(M13/M113,M213,M313)

Activates a clockwise rotation, and can be used in the SPS to turn on coolant for spindles S/S1,S2,S3 respectively.

(M14/M114,M214,M314)

Activates a counterclockwise rotation, and can be used in the SPS to turn on coolant for spindles S/S1,S2,S3 respectively.

(M19/M119,M219,M319)

Activates a spindle stop at programmed position for spindles S/S1,S2,S3 respectively. The contents of the spindle position S-word is interpreted in degrees.

The 16 Functional Groups of Auxiliary M-Functions

- | | |
|--------------------------------|--|
| 1. (M000,M001,M002, M030) | Program Control |
| 2. (M046,M047) | Spindle Override |
| 3. (M048,M049) | Feedrate Override |
| 4. (M103,M104,M105, M113,M114) | Spindle "S" and "S1" control commands |
| 5. (M203,M204,M205, M213,M214) | Spindle "S2" control commands |
| 6. (M303,M304,M305, M313,M314) | Spindle "S3" control commands |
| 7. M107,M108,M109) | Spindle "S" and "S1" coolant/lubricant |
| 8. (M207,M208,M209) | Spindle "S2" coolant /lubricant |
| 9. (M307,M308,M309) | Spindle "S3" coolant/lubricant |

10. (M110,M111)	Spindle "S" and "S1" clamp/unclamp
11. (M210,M211)	Spindle "S2" clamp/unclamp
12. (M310,M311)	Spindle "S3" clamp/unclamp
13. (M140...M145)	Spindle "S" and "S1" gear selection
14. (M240...M245)	Spindle "S2" gear selection
15. (M340...M345)	Spindle "S3" gear selection
16. (M119)	Spindle "S" or "S1" orient, NC block active and all undefined M-functions

NOTE: All user defined M-functions that do not belong to a specific functional group are defined under group 16.

Q Auxiliary Q-Function

Syntax: Qx :x = 0...9999 Allows the user to specify 10000 more auxiliary functions. The Q-functions are all user defined. Q-functions Q9000 to Q9999 are reserved for Indramat specific functions starting with version 15VRS. Only one Q-function can be programmed per NC-block.

4.10 Process Control Commands

Process control NC-commands can be used when at least one process (slave) is controlled by the management process number "0". The status of the process control commands are transferred to the SPS via Gateway signals. In the SPS program, their status is then transferred between the processes.

DP Define Process

Syntax: DP_x :x = process/external mechanism number Informs the SPS that the specified process will be needed during the NC program execution and used to initialize the specific signals of the selected process. All other process control commands will be accepted by the CNC only if the process has been defined first.

SP Select Process

Syntax: SP_x_y :x = process/external mechanism number
: y = NC-program number 0...99

Selects the defined NC-program (0...99) for the specified process (0..31). NC-programs can be selected while another NC-program is still executing. The last selected NC-program will be executed with the next process advance or reverse.

RP Reverse Process

Syntax: RP_x :x = process/external mechanism number The active advance program of the specified process (0...31) is stopped immediately and the reverse program is executed when this NC-command is used. The reverse program starts at the active reverse vector or ".HOME" label.

AP Advance Process

Syntax: AP_x :x = process/external mechanism number The currently selected advance program is started immediately at NC-block number 0 ".HOME label" if no other program is active in process (0...31). The management program will wait until an active NC-program is finished and then start the selected program.

WP Wait for Process

Syntax: WP_x :x = process/external mechanism number (0..31). The management process will wait until the NC-program execution of the specified process is finished.

LP Lock Process

Syntax: LP_x :x = process/external mechanism number (0..31). Used in the management program to hold or bypass operation of scheduled processes. This status must be held until the management program is finished.

NOTE: *An error with immediate stop occurs if a process is defined and locked within the same program. The PLC program contains USER logic that locks the process/mechanism.*

POK Process Complete (Full Depth)

Syntax: POK

When programmed in the management process (0) program, full depth indication is set if all slave processes defined in this program have executed their POK commands. When programmed in a slave process (1...6) program, full depth indication is set in the process for which it is programmed when the POK command is executed.

4.11 Program Control NC Commands

RET Return to NC-program Beginning

Syntax: RET

The program branches to the first NC-block of the currently selected NC-program when executed. The default preparatory functions (G-codes) are selected and the process waits for a new start signal.

NOTE: *The NC-command RET resets the active reverse vector to the label ".HOME" and resets the NC-subroutine levels. The CNC will assume it is in the main NC-program level, therefore the RET command is not allowed to be used in subroutines.*

BST Branch with Stop

Syntax: BST.label :refer to the NC-labels section

The NC-program branches to the defined label, sets the default preparatory functions (G-codes) and waits for a new start signal.

15VRS and below

The NC-command BST resets the active reverse vector to the label ".HOME" and resets the NC-subroutine levels. The CNC will assume it is in the main NC-program level, therefore the BST command is not allowed to be used in subroutines.

16VRS and above

The BST command does not use the currently selected NC-program number as the active NC-program. If a NC-subroutine, which resided in another NC-program number is called, and a BST command was issued, the 'Branch with Stop' would occur in the active NC-program (NC-subroutine) and not in the NC-program which the subroutine was initially called.

The NC-command 'RET' continues to use the currently selected NC-program number and branches with stop to the NC-block N0000 of that NC-program.

HLT Programmed Halt

Syntax: HLT

Interrupts the NC-program execution and the process waits for a new start signal. NC-program execution will continue from this point once a new start signal is issued and no reverse program has been executed.

BRA Branch Absolute

Syntax: BRA_label :refer to the NC-labels section

Once issued the NC-program branches to the defined label and continues the NC-program execution.

JMP Jump to NC-program

Syntax: JMP_x :x = NC-program number 0...99 or NC-variable

Once issued the NC-program continues execution with the first NC-block of the defined NC-program. The NC-program number can be programmed as a 2 digit decimal number or be the integer contents (1...99) of a NC-variable.

NOTE: *The NC-command JMP does not influence reverse vectors.*

4.12 Subroutines

The NC-program number 99 of each process can be used to store global NC-subroutines. These global NC-subroutines can be then called via NC-command BSR in any NC-program of the process. The CNC will first search the active NC-program for the NC-subroutines label and if not found the CNC will search NC-program 99 of the same process for the NC-subroutine label.

NOTE: *NC-subroutines can be nested up to 10 levels.*

JSR Jump to NC-program Subroutine

Syntax: JSR_x :x = NC-program number 0...99 or NC-variable

After execution the NC-program continues with the first NC-block of the defined NC-program. This requires the NC-command RTS in the defined NC-program to return from the NC-subroutine.

NOTE: *The NC-command JSR does not influence reverse vectors.*

BSR Branch to Subroutine

Syntax: BSR_label :refer to the NC-labels section

Identifies the NC-block where the NC-subroutine starts. The NC-command RTS is required at the end of the NC-subroutine to return to the main NC-program.

RTS Return from Subroutine

Syntax: RTS

The end of a NC-subroutine is identified by the NC-command RTS. The NC-program execution continues with the NC-block that follows the NC-block containing the last executed NC-command BSR or JSR.

4.13 Reverse Vector

REV Reverse Vectors

Syntax: REV_label :refer to the NC-labels section

Each process of the MT-CNC has its own and independent reverse vector register. The label identifies the NC-block where the NC-program execution continues if the reverse program is started.

The NC-command REV loads the defined label as the active reverse vector, at the end of the NC-block execution.

NOTE: *Each NC-program or NC-program 99 (if the same reverse program for all process NC-programs) must contain the label ".HOME" to identify the beginning of the basic reverse program.*

The basic reverse vector ".HOME" is active after the MT-CNC power is switched on, NC-program end (RET, BST) (M30, M02) and after a control reset for the process. In these cases all other process reverse vectors and NC-subroutines levels are deleted.

NOTE: *All labels defined with the NC-command REV must be available within the same NC-program. The CNC does not search in the global NC-program 99 for a reverse vector label.*

4.14 Re-positioning and NC-block Restart (GUI)

It may be necessary to continue NC-block execution at the point of interruption even though the tool was removed manually via jogging to examine or work on the tool.

The function Re-positioning (PxxC.REPOS) allows to return the tool tip to the exact position where the NC-program was interrupted and to continue NC-program execution from the interrupted point.

4.15 5 Branch Commands

Conditional branches are executed and the NC-program execution continues at the NC-block with specified label only if the corresponding condition is true. NC-program execution continues with the next NC-block if the condition is false.

BRF Branch if Referenced

Syntax: BRF.label :refer to the NC-labels section

A branch to the defined label is performed if all process axes are referenced (homed) and the NC-command BRF is executed.

BES Branch if Event Set

Syntax: BES.labelx :refer to the NC-labels section
x = event number

A branch to the NC-block starting with the defined label is performed if the Boolean status of the defined event is set to "1".

BER Branch if Event Reset

Syntax: BER.labelx :refer to the NC-labels section
x = event number

A branch to the NC-block starting with the defined label is performed if the Boolean status of the defined event is set to "0".

BEQ Branch if Equal to Zero

Syntax: BEQ.label :refer to the NC-labels section

A branch to the NC-block starting with the defined label is performed if the result of the last mathematical operation is equal to zero.

NOTE: *Results may not always be zero because of the resolution. In such a case use BMI and BPL.*

BNE Branch if Not Equal to Zero

Syntax: BNE.label :refer to the NC-labels section

A branch to the NC-block starting with the defined label is performed if the result of the last mathematical operation is NOT equal to zero.

BPL Branch if Equal to or Greater than Zero

Syntax: BPL_.label :refer to the NC-labels section

A branch to the NC-block starting with the defined label is performed if the result of the last mathematical operation is equal to or greater than zero.

BMI Branch if Less than Zero

Syntax: BMI_.label :refer to the NC-labels section

A branch to the NC-block starting with the defined label is performed if the result of the last mathematical operation is less than zero.

4.16 NC-Events

NC-events are Boolean type variables that can be used in the NC-program, similar to markers in SPS. NC-events can be set and reset in the NC-program, SPS program, MUI and SOT for each process. NC-event status is retentive.

NC-events are identified by the process (optional) and the NC-event number. NC-events executed without a process number default to the process in which they were programmed in a NC-program. The character "*" can be used instead of an event number to denote all NC-events of the specified process. Up to 244 NC-events are available in the MT-CNC. 32 local NC-events are allocated to each process. When the process numbers are used for addressing, all 224 NC-events can be used in a process.

Figure 5.16: Syntax for NC-Events

Syntax: event <process number>: <event number>
 process number (1...6)
 event number (0...31, * (all))

SE Set NC-Event

Sets the status of the specified NC-event to a binary status of "1". The event remains set until reset with the NC-command RE. All events in a process can be set to "1" if the character "*" is programmed for the NC-event number.

RE NC-Event

Resets the status of the specified NC-event to a binary status of "0". The event remains reset until it is set with the NC-command SE. MT-CNC power OFF does not reset NC-events. All events in a process can be set to "0" if the character "*" is programmed for the NC-event number.

WES Wait until NC-event is Set

Stops the process NC-program execution until the status of the programmed event is "1" (set). The NC-program execution stops until **at least one** of the process events has the status of "1" (set), if the character "*" is used instead of the NC-event number.

WER Wait until NC-event is Reset

Stops the process NC-program execution until the status of the programmed event is "0" (reset). The NC-program execution stops until **all** of the process events have the status of "0" (reset), if the character "*" is used instead of the NC-event number.

4.17 Interrupting NC-Events

The NC-events 0 to 7 of each process are interrupting NC-events. The branch on NC-event occurs at NC-event signal status of "1". NC-event 0 has the lowest, and NC-event 7 has the highest priority. This allows the CNC to react on an NC-event while one or more NC-events with lower priority are being processed. Interrupting NC-events are processed at the end of NC-block execution.

NOTE: *Interrupting NC-events are monitored in the advance program only.*

The control of and monitoring of interrupting NC-events is canceled at MT-CNC power off, end of NC-program execution, jogging a process axis and if a control reset is performed for the process.

DEV Disable NC-event Supervision

Syntax: DEV

Disables interrupting process NC-events temporarily for a certain NC-program section until they are enabled again via the NC-command EEV or at the end of NC-program execution.

EEV Enable NC-event Supervision

Syntax: EEV

Enables the supervision of all interrupting NC-events, that was previously disabled via the NC-command DEV. The supervision of NC-events is enabled at the end of the NC-program execution.

BEV Branch on NC-event to Subroutine

Syntax: BEV.labelx :refer to the NC-labels section

:x = event number 0..7

A NC-event status of "1" causes the CNC to continue NC-block execution at the NC-block with the defined label. The CNC then ignores NC-events with lower priority or triggering the NC-event itself until the defined NC-subroutine is executed, but still can be interrupted by NC-events with higher priority.

The NC-subroutine must be concluded with the NC-command RTS. The execution of RTS automatically reactivates the monitoring of the triggering NC-event and all NC-events with lower priority.

JEV Jump on Event

Syntax: JEV.labelx :refer to the NC-labels section

:x = event number

The defined process NC-event is monitored by the CNC after executing the NC-command JEV. A NC-event status of "1" causes the CNC to continue execution at the NC-block with the defined label. The JEV NC-command is not a NC-subroutine command and it is not possible to return to the interrupted NC-block via the NC-command RTS.

CEV Cancel NC-event Supervision

Syntax: CEVx :x = event number 0..7

Cancels the supervision of a single interrupting NC-event. The supervision (monitoring) of NC-events must be activated via the NC-commands BEV or JEV.

4.18 Mathematical Expressions

The values for the following can be calculated and assigned via mathematical expressions in NC-programs:

-
- Axis coordinates (X,Y,Z,A,B,C,U,V,W)
 - Interpolation parameters (I,J,K)
 - F-word , feedrate (F)
 - S-word, spindle speed (S,S1,S2,S3)
 - Rotation angle of the active plane (P)
 - Tool number (T)
 - NC-variables (@)
 - D-correction ^{(2),(3),(4),(5)}
 - nominal radius
 - G-codes (G)
 - M-codes (M)
 - Path acceleration limit (ACC)
 - Tool edge (E)

The character "=" must be used in the syntax to assign the result of a mathematical expression. The assignment ends if the "=" is separated by one or more line feed or a space character (s).

Constants Floating point constants can consist of the following elements:

- sign of mantissa
- up to 6 decimal places
- decimal point behind the first to sixth decimal place
- exponent character “E”
- sign of exponent
- up to 2 decimal exponent digits

System Available system constants.

Constants PI = 3.14159... - (for circumference/diameter ratio)
KI = 25.4 - (Inch/Metric and Metric/Inch conversion)

Operators Commonly used symbols are used for the basic mathematical operations:

(+)	Addition
(-)	Subtraction
(*)	Multiplication
(/)	Division
(%)	Rest of integer division (Module)

4.19 NC-Functions

Mathematical functions can enclose their operands with parentheses,“(“ and “)”. The operands of functions can also be mathematical expressions, by which nesting of functions is possible.

ABS Absolute value

Syntax: ABS(x) :x = number

The function ABS(x) provides the positive value of its operand.

INT Integer

Syntax: INT(x) :x = number

The function INT(x) rounds off the value of the number.

SQRT Square Root

Syntax: SQRT(x) :x = number

Provides the square root of its operand.

NOTE: *Negative values used with SQRT will result in a process error.*

SIN Sine

Syntax: SIN(x) :x = number

The operand of the function SIN (X) depends on the active angle unit “radians” (NC-command RAD) or “degrees” (NC-command DEG).

COS Cosine

Syntax: COS(x) :x = number

The operand of the function COS (X) depends on the active angle unit “radians” (NC-command RAD) or “degrees” (NC-command DEG).

TAN Tangent

Syntax: TAN(x) :x = number

NOTE: *The operand of the function TAN (X) depends on the active angle unit “radians” (NC-command RAD) or “degrees” (NC-command DEG). The values (PI/2,-PI/2), (+90,-90) are not defined and will result in a process error when used with TAN.*

ASIN Arc Sine

Syntax: ASIN(x) :x = number

NOTE: *The operand of the function ASIN (X) must be greater than “-1” and less than “1” or a process error occurs.*

The following is valid if the NC-command RAD for “radians” is active.
(Value range: $-\pi/2 \leftarrow \text{ASIN}(X) \Rightarrow +\pi/2$)

The following is valid if the NC-command DEG for “degrees” is active.
(Value range: $-180 \leftarrow \text{ASIN}(X) \Rightarrow +180$)

ACOS Arc Cosine

Syntax: ACOS(x) :x = number

NOTE: *The operand of the function ACOS (X) must be greater than “-1” and less than “1” or a process error occurs.*

The following is valid if the NC-command RAD for “radians” is active.
(Value range: $-\pi/2 \leftarrow \text{ACOS}(X) \Rightarrow +\pi/2$)

The following is valid if the NC-command DEG for “degrees” is active.
(Value range: $-180 \leftarrow \text{ACOS}(X) \Rightarrow +180$)

ATAN Arc Tangent

Syntax: ATAN(x) :x = number

The following is valid if the NC-command RAD for “radians” is active.
(Value range: $-\pi/2 \leftarrow \text{ATAN}(X) \Rightarrow +\pi/2$)

The following is valid if the NC-command DEG for “degrees” is active.
(Value range: $-180 \leftarrow \text{ATAN}(X) \Rightarrow +180$)

E^ Exponential value to the power of “e”

Syntax: E^(x) :x = number

10^ Exponential value to the power of “10”

Syntax: 10^(x) :x = number

2^ Exponential value to the power of “2”

Syntax: 2^(x) :x = number

LN Logarithm to the base of “e”

Syntax: LN(x) :x = number

NOTE: *The operand must be greater than 0 with function E^ to prevent a process error.*

LG Logarithm to the Base of “10”

Syntax: LG(x) :x = number

NOTE: *The operand of the function LG (X) must be greater than 0 to prevent a process error.*

LD Logarithm to the Base of “2”

Syntax: LD(x) :x = number

NOTE: *The operand of the function LD (X) must be greater than 0 to prevent a process error.*

4.20 Time Functions

TIME Syntax: TIME

The function TIME provides a rollover timer source in the CNC with a resolution of 2ms.

4.21 Position Strobe

Each of the 20 axes allowed in the MT-CNC has a register in which a position value that has been strobed can be stored.

PMP Positive Memorized Position

Syntax: PMP(x) :x = axis designation

Reads the position of the defined axis that was strobed with the positive edge of the gateway signal AxxCSTRBP (xx = axis number).

NMP Negative Memorized Position

Syntax: NMP(x) :x = axis designation

Reads the position of the defined axis that was strobed with the negative edge of the gateway signal AxxCSTRBP (xx = axis number).

4.22 Functions for Digital Drives

Data can be read from or written to Indramat digital drives that are connected to the MT-CNC via the SERCOS fiber optic ring. The axis parameter (data) is addressed via its SERCOS Ident-Number and the axis designation (Cxx.001).

AXD Data Exchange with Digital Drives

Syntax: AXD(x:y) :x = axis designation
 :y = Ident number

Different access priorities exist for the SERCOS service channel: Highest priority: NC-program (CPU/APR) Lowest priority: MUI

NOTE: *Digital drive parameter data can be read only into a NC-variable, and is written to the drive at the end of NC-block execution. The assignment to a NC-variable therefore can not be used as a branch condition. No additional operators besides AXD are allowed in an NC-expression.*

4.23 NC-Variables

A NC-variable represents a numerical value (Double REAL) which can be accessed in the NC-program by using the NC-variable identifier “@”. The contents of NC-variables can be read from or be written to using the NC-program, SPS program, the MUI and SOT. NC variable content is retentive.

NOTE: *The NC-variables (100...255) are used in the standard Indramat NC-subroutines. Therefore they should not be used by the user, unless none of the Indramat NC-subroutines are used.*

- Variable content can be assigned to: ^{(3),(4),(5)}
 - Axis designation (X=@1, @1=X3)
 - Interpolation parameter (I=@1, @1=I)
 - F-word (F=@1, @1=F)
 - S-word (S=@1, Q1=S3)
 - Tool Edge (E=@1, @1=E)
 - Active plane angle (P=@1, @1=P)
 - Path acceleration factor (ACC=@1)
 - T-number (T=@1, @1=T)
 - Tool Edge (E=@1, @1=E)
 - Nominal radius (consider rotary axis in path velocity) (RX=@1)
 - Offset Table selection (O=@1)
 - Active G-code of group x (@1=G(x))
 - M-function of group x (@1=M(x))
 - Q-function (@1=Q)
 - Active G-code, M- / Q-function (M=@1, Q=@1, G=@1)
 - Activation of D-correction (D=@21)
 - Digital drive data (AXD(X:P-7-3576)=@2, @23=AXD(X2:S-0-0159))
 - Tool Data (TLD(3,1, @34,,0,9,22)=1, @3=TLD(,0,1,2,0,5))
 - Machine Data (MTD(50,1,1,10)=@2, @8:134=MTD(50,1,1,10))

Each MT-CNC process provides 256 NC-variables. A total of 1792 NC-variables are available in the MT-CNC, which can be used independent on how many processes are defined.

The NC-variable is defined for the process it is programmed in, if the optional process number is not defined.

Syntax: @x:p ;x = variable number 0..255
;p = process number 0..6

4.24 Spindle Selections

SPT Spindle Selection for Tool Management

Syntax: SPTx :x = tool spindle 1..3 ⁽¹⁾⁽²⁾
1..4 ⁽³⁾

Selects the spindle for the next tool selection (programmed T-word). The spindle "S" or "S1" is automatically selected as default at the start of each NC-program, after MT-CNC power on, control reset and execution of the NC-command BST or RET.

SPF Spindle Selection for Feed Programming

Syntax: SPFx :x = spindle number 1 (S or S1)
:x = spindle number 2 (S2)
:x = spindle number 3 (S3)

Selects the spindle that should be synchronized or coupled with the feed axes if more than one spindle is operated by the process. The spindle "S" or "S1" is automatically selected as default at the start of each NC-program, after MT-CNC power on, control reset and execution of the NC-command BST or RET.

4.25 Offset Tables

The Offset table provides offset data for all process axes. The offset table allows to enter the coordinate values for non programmed zero points. These zero points are activated together with the General Offset via the preparatory functions G54...G59. In addition the Offset table displays the values of the currently programmed zero point (G52), active absolute (G50) and incremental offset (G51).

O Offset table Selection

Syntax: O_x : x = Offset Table number 0...9

The maximum number of Offset Tables (1...10) for a process can be specified by the user via the process parameter Bxx.012. The default of this parameter is zero.

NOTE: *If a new Offset Table is being activated, the active workpiece zero point is canceled (G53 becomes active and all other offsets such as G50,G51,G52 are deactivated).*

The Offset Table number zero is activated by default. A selected Offset Table stays modal active until the end of the NC-program (BST,RET) (M2, M30)^{(3),(4),(5)} or until another Offset Table is being activated.

NOTE: *Preparatory functions are processed by the CNC before the NC-command O however, the values for the preparatory functions (G54...G59) depend on the active Offset Table. The NC-command O must be therefore be programmed in a NC-block prior to these preparatory functions.*

OTD Offset Table Data Read/Write

Syntax: OTD(a,b,c,d,e)

:a = NC-memory (1,2) [If this parameter is not declared, the active NC-memory is addressed]

:b = process number (0...6) [If this parameter is not declared, the active NC-memory is addressed]

:c = Offset Table (0...9) [If this parameter is not declared, the active NC-memory is addressed]

:d = Currently active offset [If this parameter is not declared, the active NC-memory is addressed]

0 = "Total" offset value

1 = G50/G51 offset value

2 = G52 offset value

3 = General Offset value

4 = Offset value of G54

5 = Offset value of G55

6 = Offset value of G56

7 = Offset value of G57

8 = Offset value of G58

9 = Offset value of G59

:e = axis via axis-coordinate system association [The axis parameter must be declared]

1 = axis defined as main axis X

2 = axis defined as main axis Y

3 = axis defined as main axis Z

4 = axis defined as parallel (to main axis X) axis U

5 = axis defined as parallel (to main axis Y) axis V

6 = axis defined as parallel (to main axis Z) axis W

- 7 = axis defined as rotary (around main axis X) axis A
- 8 = axis defined as rotary (around main axis Y) axis B
- 9 = axis defined as rotary (around main axis Z) axis C
- 10 = value of plane rotation angle P

Used in the NC-program to read data from and write data to any available Offset Table. Constants or NC-variables can be used for the parameters. Mathematical expressions however, are **NOT** allowed as command parameters. The commas used to separate the parameters must always be used.

NOTE: *No data can be written to G50/G51, G52 and the "Total" offset shift, using the NC-command OTD.*

4.26 Axis Switching

All axes defined in the MT-CNC can be used in up to 4 of the defined processes, allowing the axis to belong to different groups of interpolating axes. All axis defined in the MT-CNC, except tool storage axes (magazines/turret) can be switched between processes. The axis parameter Cxx.000 must be used to define the primary and up to 3 secondary processes. All axis belonging to a different primary process are freed (free) at the end of the program by 'RET', 'BST', control reset or by jogging the axes in setup mode.

NOTE: *Axes cannot be associated and used in more than one process at any point of time.*

FAX Free Axis from Process

Syntax: FAX(x) :x = axis active designation (Cxx.001) or 2(Cxx.) ^{(2),(3),(4),(5)}

An axis can be switched only if it is freed by the process that currently uses the axis, and if the axis is requested (GAX) by another process. The NC-program execution of the process that frees up the axis stops at the NC-block containing the NC-command FAX until the axis is requested by the NC-command GAX.

GAX Get Axis from Process

Syntax: GAX(x:y) :x = active optional process number (0..6)
:y = axis designation (Cxx.001) or 2(Cxx.) ^{(2),(3),(4),(5)}

An axis can be switched only if it is freed by the process that currently uses the axis, and if the axis is requested (GAX) by another process. The NC-program execution of the process that requests the axis stops at the NC-block containing the NC-command GAX until the axis is freed by the NC-command FAX.

4.27 Path Acceleration

ACC Path Acceleration Limit

Syntax: ACC_x :x = constant (0..100)
ACC=y :y = mathematical expression

Used to limit the maximum path acceleration in percent. The process parameter Bxx.007 contains the value for the maximum path acceleration.

NOTE: *A process error is activated if ACC is programmed with a value that exceeds the allowed range of (0..100%).*

4.28 Machine Data

MTD The NC-command MTD (Machine Tool Data) can be used to read Machine Data elements in the NC-program or to write data into Machine Data elements (if the element is not write protected).

	Page	Dim1	Dim2	Element
MTD([1..299],[-1000..+1000],[1000..+1000],[1..1000])				
				Element-No.
				Dimension 2
				Dimension 1
				Page-No.

Name	Symbol	Value range	Explanation
Page-No.	Page	1..299	001 .. 099 Pages of Control Machine Data 100 .. 199 Pages of OEM-Machine Data 200 .. 299 Pages of User Machine Data
Dimension 1	Dim1	min. value ..max. value	min. value: first value in the structure definition (↑ - 1000) max. value: second value in the structure definition (↑ +1000) (max. value - min. value ↑ 1000)
Dimension 2	Dim2	min. value ..max. value	min. value: first value in the structure definition (↑ - 1000) max. value: second value in the structure definition (↑ +1000) (max. value - min. value ↑ 1000)
Element-No.	Element	1..max. value	max. value ↑ 1000

Individual numbers must be separated via comma “,”.
 NC-variables can be used instead of numbers (constants).
 Mathematical expressions can NOT be used instead of numbers (constants) or NC-variables.
 All Parameters of the NC-command MTD must be programmed at all times.

All parameters of the NC-command MTD must be within the specified range. The NC checks for parameter validity at the time of NC-program execution. NC-program execution is interrupted with the process error

- #284 ‘Invalid data address in NC-command’
- #286 ‘Invalid parameter @ in NC-command programmed’

if less or more parameters than required are programmed.

If the Machine Data element is write protected the process error #285 'Data is write protected' is issued and NC-program execution is stopped.

No process error is generated and NC-program execution continues if the defined maximum range of the Machine Data Element is being executed, however the element's value is limited to the defined max. or min. value.

4.29 Tool Data

TLD Read/Write Tool Data from the NC-program

The TLD command can be used to read the tool data into the NC-program and to write them from the NC-program, however some restrictions apply to writing. All data present in the tool list can be read. The individual data elements are addressed by means of elements.

P	A	S/T	L/D	E	D	S	
TLD([0..6],[0],[0...2] [1...999],[0...9],[1...35],[1...32])							
TLD([0..6],[1],[1...9999999],[1...999],[0...9],[1...35],[1...32])							
							Status
							Data Element
							Edge
							Pocket / Index
							No.
							Storage [0..2] /
							Tool Number
							Addressing
							Process

4.30 Modal Functions

MODF_ON Modal Function, ON*R[MODF_OFF]⁽⁵⁾
 Enables the programming of repeated expressions one time only.

MODF_OFF Modal Function, OFF*R[MODF_ON]⁽⁵⁾
 Disables the Modal Function.

4.31 NC-block Look Ahead Function

LA_ON NC-block Look Ahead Function, ON⁽⁵⁾
 The extended NC-block Look Ahead function optimizes the path velocity profile of the profile of the programmed path movement during the compilation process and NC-program download. Required in-between NC-blocks are inserted to create a smooth path velocity profile in this process without contour modification. Most of the global variables for the Look Ahead function can be specified in the NC-program Options menu.

LA_OFF NC-block Look Ahead Function, OFF⁽⁵⁾
 The extended NC-block Look Ahead function is switched OFF.

4.31.1 Global Variable Definitions

METB Minimum Execution Time for a Block⁽⁵⁾
Value range: (2.5...30.0)ms
 The global variable *Minimum Execution Time for a Block* METB defines the minimum execution time of a NC-block within the polygon course that is to be optimized. METB must be selected greater than or equal to the NC-block cycle time.

VFBT Velocity Factor for Block Transition⁽⁵⁾

Value range: (1.0...25.0) [-]

The global variable *Velocity Factor for Block Transition* VFBT allows to influence the velocity change at non tangential NC-block transitions.

BBTPC Block Buffer for Tool Path Compensation⁽⁵⁾

Value range: (1...10) [-]

The global variable '*block buffer for tool path compensation* BBTPC' specifies the number of NC-blocks that must be considered in the pre-calculation and pre-check of the tool radius path correction when using the extended Look Ahead function.

$$\text{Velocity change} = 2 * \text{Axis acceleration} * \text{VFBT}$$

TRC Tool Radius Correction for Look Ahead Function⁽⁵⁾

Value range: (0...2) [-]

TRC=0: The extended Look Ahead function does NOT perform a Radius Path correction.

TRC=1: The extended Look Ahead function performs the Radius Path correction to the left of the contour.

TRC=2: The extended Look Ahead function performs the Radius Path correction to the right of the contour.

NOTE: *Independent of the Tool Path correction of the extended Look Ahead function, the tool path compensation of the MT-CNC should NOT be activated when executing polygon courses.*

TL_RADIUS [T-Nr., E-Nr.]⁽⁵⁾

Value range: ($1 \leq \text{T-Nr.} \leq 9999999$, $1 \leq \text{E-Nr.} \leq 9$)

With the help of the pre-defined fields TL_RADIUS[T-Nr., E-Nr.] the for the extended Look Ahead function required tool radii can be defined centrally at the NC-program begin. This allows to read the data from the Active Tool List at the time of NC-program compilation and download.

In case that no T-Nr. or no E-Nr. is specified, the active T- or E-number in the active tool spindle (SPTx) of the Active Tool List are used by the NC-compiler.

NOTE: *In case the NC-compiler does not find in the NC-program definitions regarding tool radii via T_RADIUS[T-Nr., E-Nr.], the NC-compiler accesses the maximum and minimum tool radius of the corresponding tool in the Tool Setup List and calculates the radius as follows:*

$$\text{Radius} = \frac{1}{2} * (R_{\min} + R_{\max})$$

NOTE: *If the tool radius path correction of the extended Look Ahead function is used (TRC <> 0), the via the field TL_RADIUS[T-Nr., E-Nr.] specified tool radius must exist at the time of processing when compiling the NC-program.*

ADTRC Approach Distance for Tool Radius Comp⁽⁵⁾

Value range: ?

ACC_EFF EFFective ACCeleration⁽⁵⁾

Value range: (1..200%)

In difference to the NC-command ACC, the command ACC_EFF does NOT limit the maximum path acceleration specified in the process parameters, but modifies the real path acceleration depending on the defined value.

ACD_COMP **Actual Data for NC-COMPiler⁽⁵⁾**
Value range: <NC-variable definition>
Allows to embed the actual value of a NC-variable at the time of NC-program compilation and download.

5. SPS Instruction Set

5.1 Elementary Data Types

Table 5-1: Elementary Data Types

Data type	Explanation	Bits	Initial value	Range
BOOL	Boolean	1	0	TRUE/FALSE, 1/0
BYTE	Bit string, length 8	8	0	Bit strings only, numeric values cannot be entered
WORD	Bit string, length 16	16	0	
DWORD	Bit string, length 32	32	0	
CHAR	Character	8	" (empty)	Single character
STRING	Character string	...	" (empty)	String of 1 ... 255 characters
SINT	Short integer number	8	0	-128 -
INT	Integer number	16	0	127
DINT	Double integer number	32	0	-32768 - 32767 -2147483648 - 2147483647
USINT	Unsigned short integer number	8	0	0 - 255
UINT	Unsigned integer number	16	0	0 - 65535
UDINT	Unsigned double integer number	32	0	0 - 4294967295
REAL	Floating point number	32	0.0	-3.402823E38 ... - 1.401298E-45 1.401298E-45 ... 3.402823E38
TIME	Time value	32	T#0s	0 ms ... 23d23h59m59s999ms

Table 5-2: Character String Initialization

Name	Type:	:=	Reserved number of bytes	Character string after initialization
STR1	STRING	'abc'	256 byte	length: 3, content: "abc"
STR2	STRING[10]	'abc'	11 byte	length: 3, content: "abc"

5.2 Derived Data Types

'**Derived data types**' can be created on the basis of the elementary data types. The derived data types declared in the standard library have invariably been defined for the user. The user may employ them (like the elementary data types) for variable declaration, but will not be able to modify them. Before a user-declared data type can be employed, it must be imported into the file that is to be used. Structures (STRUCT) and arrays (ARRAY) have been implemented in the SPS's programming and commissioning system.

5.2.1 Structures (STRUCT)

A '**structure**' consists of several individual data elements that are functionally correlated from the controller manufacturer's or user's perspective (for example the complete NC axis status information that can be processed in the NC, or a complete data record that characterizes a specific workpiece type).

Names, element type, and sequence within a structure are defined by a declaration or the standard library. The name of an element consists of the variable name and its own name, separated by a period (.), (**variablen_name.element_name**).

The individual elements may belong to different data types. The following data element types are employed:

- Elementary data types
- Structures.

Arrays can not yet be used as elements in a structure.

5.2.2 Arrays

An 'array' consists of several individual data elements of the same data type that are functionally correlated from the controller manufacturer's or user's perspective (e.g. information about the storage locations of workpieces on a pallet, or the tool mounted in a tool changer). According to their destination, arrays can be unidimensional or multi-dimensional (maximum 4-dimensional). Each dimension may possess a maximum of 1000 elements. The maximum memory requirement of an array is monitored, and limited to 64 kBytes. **Element type, and sequence within an array are defined by a declaration or the standard library. The name of an element consists of the variable name and the specification of its position in the array, (variablen_name[i,j] i,j - position (2-d array)).**

The following data element types are employed:

- Elementary data types
- Structures.

5.3 Instruction List Instructions

This chapter contains the elementary components of the 'Instruction list' input language. The following '**instructions**' are supported:

- Load, save, set, and reset instructions
- Logic instructions
- Jumps, calls, return (conditional and unconditional)
- Comparison instructions
- Arithmetic instructions

NOTE: *All the instructions that are used for interconnecting data can only be used for operands of the same data type.*

Table 5-3: Instruction List Instruction

Load, save, set, and reset instructions		Logic instructions		Jumps, calls, return (cond. and uncond.)		Comparison instructions		Arithmetic instructions	
LD	SET	AND	AND(JMP	CAL	EQ	EQ	ADD	ADD
LDN	SETC	ANDN	ANDN(JMPC	CALC	EQN	EQN(SUB	SUB(
ST	SETCN	OR	OR(JMPN	CALCN	NE	NE(MUL	MUL(
STN	RES	ORN	ORN(RET	NEN	NEN(DIV	DIV(
	RESC	XOR	XOR(RETC	GT	GT(MOD	MOD(
	RESCN	XORN	XORN(RETCN	GE	GE(
						LT	LT(
						LE	LE(

5.3.1 Load and save instructions

LD loads the operand value
 LDN loads the negated operand value
 ST saves the current value in the operand
 STN saves the bit-by-bit negated value in the operand

Valid for the data types: BOOL, BYTE, WORD, DWORD
 Complex load and save instructions:

LD loads the operand value
 ST saves the current value in the operand

Valid for the data types: all numbers, STRUCT, ARRAY, STRING, CHAR

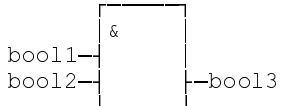
5.3.2 Set and reset instructions

SET sets the bit operand unconditionally
 SETC sets the bit operand if the preceding result is TRUE; otherwise no change
 SETCN sets the bit operand if the preceding result is FALSE; otherwise no change
 RES resets the bit operand unconditionally
 RESC resets the bit operand if the preceding result is TRUE; otherwise no change
 RESCN resets the bit operand if the preceding result is FALSE; otherwise no change

5.3.3 Logic Instructions

AND The '**AND function**' operates bit by bit. Bit positions of the same significance are interconnected if the AND function is applied to 'BYTE', 'WORD', 'DWORD' data.
 AND ANDing the current value and the operand value.
 ANDN ANDing the current value and the bit-by-bit negated operand value.
 AND(ANDing the current value and the value of the subsequent expression.
 ANDN(ANDing the current value and the bit-by-bit negated value of the subsequent expression.
) Termination of an expression.

Example: AND functions

FBD only	IL	Type
	<pre>LD bool1 AND bool2 ST bool3</pre>	<pre>bool1: BOOL bool2: BOOL bool3: BOOL</pre>
<pre>bool1 1 0 1 0 bool2 1 1 0 0 ----- bool3 1 0 0 0</pre>		

OR functions

The '**OR function**' operates bit by bit. Bit positions of the same significance are interconnected if the OR function is applied to 'BYTE', 'WORD', 'DWORD' data.
 OR ORing the current value and the operand value.
 ORN ORing the current value and the bit-by-bit negated operand value.
 OR(ORing the current value and the value of the subsequent expression.
 ORN(ORing the current value and the bit-by-bit negated value of the subsequent expression.
) Termination of an expression.

Example: OR functions

FBD only	IL	Type										
	<pre>LD bool1 OR bool2 ST bool3</pre>	<pre>bool1: BOOL bool2: BOOL bool3: BOOL</pre>										
<table border="1"> <tr> <td>bool1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>bool2</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> </table>	bool1	1	0	1	0	bool2	1	1	0	0		
bool1	1	0	1	0								
bool2	1	1	0	0								
<table border="1"> <tr> <td>bool3</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	bool3	1	1	1	0							
bool3	1	1	1	0								

XOR

The '**XOR function**' operates bit by bit. Bit positions of the same significance are interconnected if the XOR function is applied to 'BYTE', 'WORD', 'DWORD' data.

XOR XORing the current value and the operand value.

XORN XORing the current value and the bit-by-bit negated operand value.

XOR(XORing the current value and the value of the subsequent expression.

XORN(XORing the current value and the bit-by-bit negated value of the subsequent expression.

) Termination of an expression.

Examples: XOR functions

LAD / FBD	IL	Type										
	<pre>LD bool1 XOR bool2 ST bool3</pre>	<pre>bool1: BOOL bool2: BOOL bool3: BOOL</pre>										
<table border="1"> <tr> <td>bool1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>bool2</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> </table>	bool1	1	0	1	0	bool2	1	1	0	0		
bool1	1	0	1	0								
bool2	1	1	0	0								
<table border="1"> <tr> <td>bool3</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	bool3	0	1	1	0							
bool3	0	1	1	0								

5.3.4 Jumps, Calls, Return (conditional and unconditional)

Jumps

'**Jumps**' are elementary instructions that are used for branching in the instruction list. They always lead to a jump target, the label. In order to maintain compatibility in converting instruction list, ladder diagram, and function block diagram into each other, a label is only permitted at the beginning of a network, or before an 'LD' or 'LDN' instruction.

A jump can be executed '*conditionally*' or '*unconditionally*'.

JMP	mLabel	unconditional jump to the label 'mLabel'
JMPC	mLabel	Jump to the label 'mLabel' if the current value is 'TRUE'.
JMPCN	mLabel	Jump to the label 'mLabel' if the current value is 'FALSE'

NOTE: Ensure that using jumps within an instruction list does not lead to an endless loop. Check the termination condition of upward jumps.

Function Block Calls '**Function block calls**' permit the previously declared assignment of an FB type to be called within the instruction list. The call can be '*conditional*' or '*unconditional*'.

CAL	fb1	Unconditional call of the assignment 'fb1' of the FB of the type xx
CALC	fb1	Call of the assignment 'fb1' of the FB of the type xx if the current value is 'TRUE'
CALCN	fb1	Call of the assignment 'fb1' of the FB of the type xx if the current value is 'FALSE'

5.3.4.1 Return from Function Blocks or Functions

The execution of a function block or function usually ends in the last instruction list line of its implementation, without the need of a RET instruction. The user may, conditionally or unconditionally, define any other return point in the IL (in combination with a jump).

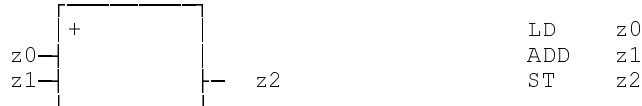
RET Unconditional return from a function block or a function.

RETC Conditional return from a function block or a function if the current value is 'TRUE'.

RETCN Conditional return from a function block or a function if the current value is 'FALSE'.

5.3.5 Arithmetic Instructions

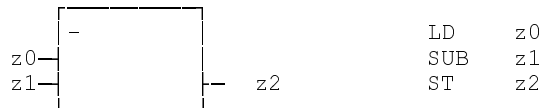
ADD The arithmetic instruction 'Addition' - 'ADD' enables numbers of the same type to be added. The result is of the same type as the Summation.



Example: for the result ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
total greater than pmax		pmax overflow	ErrorFlg: 1 ErrorNr: 2 ErrorTyp: -10004/-10005/-10006 pmax(SINT): 127 pmax(INT): 32767 pmax(DINT): 2147483647
-35 -35 35 35	-7 +7 -7 +7	-42 -28 28 42	ErrorFlg: 0 ErrorNr: 0 ErrorTyp: 0 mmax(SINT): -128 mmax(INT): -35768 mmax(DINT): -2147483648
total less than mmax		mmax overflow	ErrorFlg: 1 ErrorNr: 3 ErrorTyp: -10004/-10005/-10006

SUB The arithmetic instruction 'Subtraction' - 'SUB' permits numbers of the same type to be subtracted. The result is of the same type as the input variables.

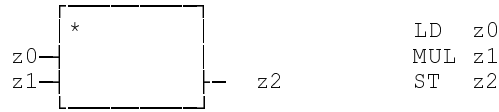


Example: for the result ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
difference greater than pmax		pmax overflow	ErrorFlg: 1 ErrorNr: 2 ErrorTyp: -10014/-10015/-10016 pmax(SINT): 127 pmax(INT): 32767 pmax(DINT): 2147483647
-35 -35 35 35	-7 +7 -7 +7	-28 -42 42 28	ErrorFlg: 0 ErrorNr: 0 ErrorTyp: 0 mmax(SINT): -128 mmax(INT): -32768 mmax(DINT): -2147483648
difference less than mmax		mmax overflow	ErrorFlg: 1 ErrorNr: 3 ErrorTyp: -10014/-10015/-10016

MUL

The arithmetic instruction 'Multiplication' - 'MUL' permits numbers of the same type to be multiplied with each other. The result is of the same type as the factors.



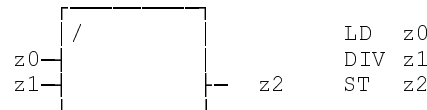
Example: for the result ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
product greater than pmax (INT)		pmax overflow	ErrorFlg: 1 ErrorNr: 2 ErrorTyp: -10024/-10025/-10026 pmax (SINT): 127 pmax (INT): 32767 pmax (DINT): 2147483647
-35	-7	245	ErrorFlg: 0
-35	+7	-245	ErrorNr: 0
35	-7	-245	ErrorTyp: 0
35	+7	245	mmax (SINT): -128 mmax (INT): -35768 mmax (DINT): -2147483648
product less than mmax		mmax overflow	ErrorFlg: 1 ErrorNr: 3 ErrorTyp: -10024/-10025/-10026

DIV

The arithmetic instruction 'Division' - 'DIV' is used to perform the division of two numbers of the same type. The result is the integer part, and of the same type as the input variables.

NOTE: Assigning default-initialized variables to the function will lead to an error.

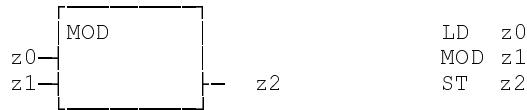


Example: for the result ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
-35	-6	5	ErrorFlg: 0
-35	+8	-4	ErrorNr: 0
35	-8	-4	ErrorTyp: 0
35	+6	5	pmax (SINT): 127 pmax (INT): 32767 pmax (DINT): 2147483647
-7	-35	0	mmax (SINT): -128 mmax (INT): -32768 mmax (DINT): -2147483648
+7	-35	0	
35	0	pmax	ErrorFlg: 1
0	0	pmax	ErrorNr: 5
-35	0	mmax	ErrorTyp: -10034/-10035/-10036

MOD The arithmetic instruction '**Modulo division**' - '**MOD**' is used to perform the Modulo division of two numbers of the same type (remainder from the division 'DIV'). The result is of the same type as the input variables (not defined for REAL).

NOTE: Assigning default-initialized variables to the function will lead to an error.



Example: for the result ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
-35	-6	-5	ErrorFlg: 0 pmax(SINT): 127
35	+8	-3	ErrorNr: 0 pmax(INT): 32767
35	-8)	3	ErrorTyp: 0 pmax(DINT): 2147483647
35	+6	5	mmax(SINT): -128
-7	-35	-7	mmax(INT): -32768
+7	-35	7	mmax(DINT): -2147483648
-7	0	mmax	ErrorFlg: 1, ErrorNr: 5
+7	0	pmax	ErrorTyp: -10044/-10045/-10046

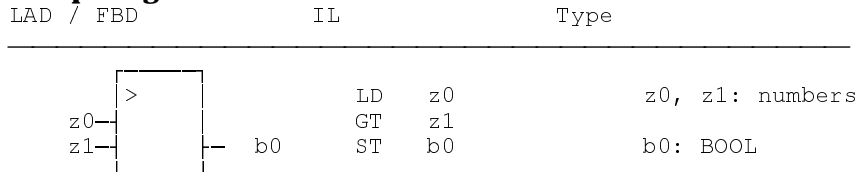
5.3.6 Comparators

'**Comparators**' are expressions that are used for comparing operands and expressions with the current value:

the magnitude of two numbers of the same type; the equality/inequality of two bit strings of the same type; the alphabetic order of two characters (CHAR) or character strings (STRING); two time values (TIME). The result is a Boolean value. An error is not possible. The 'ErrorFlg', 'ErrorNr', 'ErrorTyp' error variables will therefore not be influenced.

GT The output of the comparison function '**Greater than**' - '**GT**' is '1' if the variable/constant at the upper input (current value) is greater than the variable/constant at the lower input. Otherwise, the output will be '0'.

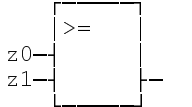
Comparing numbers:



z0	z1	b0
5	3	1
5	5	0
3	5	0

GE The output of the comparison function 'Greater than or equal to' - 'GE' is '1' if the variable/constant at the upper input (current value) is greater than or equal to the variable/constant at the lower input. Otherwise, the output will be '0'.

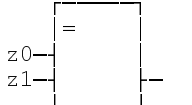
Comparing numbers:

LAD / FBD	IL	Type
	LD z0 GE z1 ST b0	z0, z1: Numbers b0: BOOL

z0	z1	b0
5	3	1
5	5	1
3	5	0

EQ The output of the comparison function 'Equal to' - 'EQ' is '1' if the variable/constant at the upper input (current value) is equal to the variable/constant at the lower input. Otherwise, the output will be '0'.

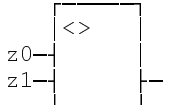
Comparing numbers:

LAD / FBD	IL	Type
	LD z0 EQ z1 ST b0	z0: INT z1: INT b0: BOOL

z0	z1	b0
5	3	0
5	5	1
3	5	0

NE The output of the comparison function 'Not equal to' - 'NE' is '1' if the variable/constant at the upper input (current value) is not equal to the variable/constant at the lower input. Otherwise, the output will be '0'.

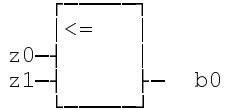
Comparing numbers:

LAD / FBD	IL	Type
	LD z0 NE z1 ST b0	z0: INT z1: INT b0: BOOL

z0	z1	b0
5	3	1
5	5	0
3	5	1

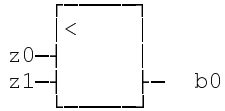
LE The output of the comparison function 'Less than or equal to' - 'LE' is '1' if the variable/constant at the upper input (current value) is less than or equal to the variable/constant at the lower input. Otherwise, the output will be '0'.

Comparing numbers:

LAD / FBD	IL	Type												
	LD z0 LE z1 ST b0	z0, z1: Numbers b0: BOOL												
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>z0</th> <th>z1</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>3</td> <td>0</td> </tr> <tr> <td>5</td> <td>5</td> <td>1</td> </tr> <tr> <td>3</td> <td>5</td> <td>1</td> </tr> </tbody> </table>	z0	z1	b0	5	3	0	5	5	1	3	5	1		
z0	z1	b0												
5	3	0												
5	5	1												
3	5	1												

LT The output of the comparison function 'Less than' - 'LT' is '1' if the variable/constant at the upper input (current value) is less than the variable/constant at the lower input. Otherwise, the output will be '0'.

Comparing numbers:

LAD / FBD	IL	Type												
	LD z0 LT z1 ST b0	z0: INT z1: INT b0: BOOL												
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>z0</th> <th>z1</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>3</td> <td>0</td> </tr> <tr> <td>5</td> <td>5</td> <td>0</td> </tr> <tr> <td>3</td> <td>5</td> <td>1</td> </tr> </tbody> </table>	z0	z1	b0	5	3	0	5	5	0	3	5	1		
z0	z1	b0												
5	3	0												
5	5	0												
3	5	1												

5.4 Ladder Diagram

The ladder diagram editor permits ladder diagram networks to be set up, using




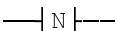

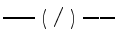
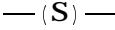
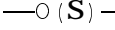
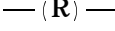
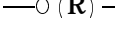




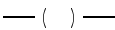
- basic elements (contacts, relays, ...),
- operators (addition, subtraction..., comparators...)
- jumps (conditional and unconditional)
- functions and function blocks.

5.4.1 Basic Elements in the Ladder Diagram

The **ladder diagram** provides the elements that are most frequently used in relay and contactor technique, and permits networks to be set up that are similar to a circuit diagram. Valid for the following data types:

- a: BOOL;
- b: BOOL, Interbus inputs only;
- v: all numbers |TIME|BYTE|WORD|DWORD;

Table 5-4: Basic Elements in the Ladder Diagram

Ladder diagram symbols	Instruction list equivalent
Contacts: a  NO a  NC b  positive edge b  negative edge	LD a, AND a, OR a, Depends on the position in the network LDN a, ANDN a, ORN a, Depends on the position in the network LD> b Up to now, only permitted in process data interconnection LD< b Up to now, only permitted in process data interconnection
Relays: a  relay a  negate + relay a  set a  set a  reset a  reset	ST a Allocation of the network value STN a Allocation of the negated network value SETC a Set if network result is '1' SET a Set unconditionally SETCN a Set if network result is '0' RESC a Reset if network result is '1' RES a Reset unconditionally RESCN a Reset if network result is '0'
Jump:  >>m  >m	JMP m Jump to 'm' if network result is '1' JMP m Unconditional jump to 'm' JMPCN m Jump to 'm' if network result is '0'
Block return:  <RET>  <O RET>	RETC Return from FB/FK if network result is '1' RET Unconditional return from FB/FK RETCN Return from FB/FK if network result is '0'
Intermediate flag: a 	Assumes result from the network to its left ST a (Boolean values)

— (v) —	ST v (non-Boolean values)
---------	---------------------------

5.5 Operations

5.5.1 Logic Bit and Bit String Operations ‘:=’, ‘AND’, ‘OR’, ‘XOR’

:= The ‘**Results as**’ operation enables the input variable value to be transferred to the operation’s output variable.

Example:

FBD only	IL	Type						
	LD bit0 ST bit1	bit0: BOOL bit1: BOOL						
<table border="1"> <thead> <tr> <th>bit0</th> <th>bit1</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table>	bit0	bit1	0	0	1	1		
bit0	bit1							
0	0							
1	1							

AND The ‘**AND operation**’ manipulates individual bits. Bits of the same position will be interconnected if it is used on ‘BYTE’, ‘WORD’, ‘DWORD’.

Example:

FBD only	IL	Type															
	LD bool1 AND bool2 ST bool3	bool1: BOOL bool2: BOOL bool3: BOOL															
<table border="1"> <tbody> <tr> <td>bool1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>bool2</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>bool3</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	bool1	1	0	1	0	bool2	1	1	0	0	bool3	1	0	0	0		
bool1	1	0	1	0													
bool2	1	1	0	0													
bool3	1	0	0	0													

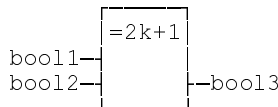
OR The ‘**OR operation**’ manipulates individual bits. Bits of the same position will be interconnected if it is used on ‘BYTE’, ‘WORD’, ‘DWORD’.

Example:

FBD only	IL	Type															
	LD bool1 OR bool2 ST bool3	bool1: BOOL bool2: BOOL bool3: BOOL															
<table border="1"> <tbody> <tr> <td>bool1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>bool2</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>bool3</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	bool1	1	0	1	0	bool2	1	1	0	0	bool3	1	1	1	0		
bool1	1	0	1	0													
bool2	1	1	0	0													
bool3	1	1	1	0													

XOR The 'XOR operation' manipulates individual bits. Bits of the same position will be interconnected if it is used on 'BYTE', 'WORD', 'DWORD'.


Example:

LAD / FBD	IL	Type															
	LD bool1 XOR bool2 ST bool3	bool1: BOOL bool2: BOOL bool3: BOOL															
<table border="1"> <tr> <td>bool1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>bool2</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>bool3</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	bool1	1	0	1	0	bool2	1	1	0	0	bool3	0	1	1	0		
bool1	1	0	1	0													
bool2	1	1	0	0													
bool3	0	1	1	0													

5.5.2 Arithmetic Operations

'Arithmetic operations' are used for linking numbers of the same type.

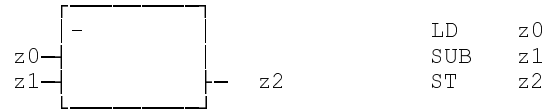
ADD The arithmetic operation 'Addition' enables numbers of the same type to be added. The result is of the same type as the summations.

	LD z0 ADD z1 ST z2
--	--------------------------

Example: Results ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
total	pmax		ErrorFlg: 1
greater than	overflow		ErrorNr: 2
pmax			ErrorTyp: -10004/-10005/-10006
-35	-7	-42	ErrorFlg: 0
-35	+7	-28	ErrorNr: 0
35	-7	28	ErrorTyp: 0
35	+7	42	
total	mmax		ErrorFlg: 1
less than	overflow		ErrorNr: 3
mmax			ErrorTyp: -10004/-10005/-10006

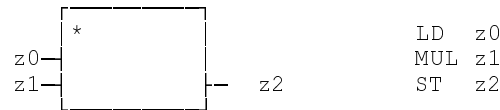
SUB The arithmetic operation '**Subtraction**' permits numbers of the same type to be subtracted. The result is of the same type as the input variables.



Example: Result ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
difference greater than pmax		pmax	ErrorFlg: 1 ErrorNr: 2 ErrorTyp: -10014/-10015/-10016
			pmax(SINT): 127 pmax(INT): 32767 pmax(DINT): 2147483647
-35	-7	-28	ErrorFlg: 0
-35	+7	-42	ErrorNr: 0
35	-7	42	ErrorTyp: 0
35	+7	28	
difference less than mmax		mmax	ErrorFlg: 1 ErrorNr: 3 ErrorTyp: -10014/-10015/-10016
			mmax(SINT): -128 mmax(INT): -32768 mmax(DINT): -2147483648

MUL The arithmetic operation '**Multiplication**' is used to perform numbers of the same type to be multiplied with each other. The result is of the same type as the factors.

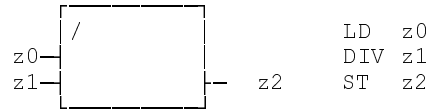


Example: Result ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
product greater than pmax		pmax	ErrorFlg: 1 ErrorNr: 2 ErrorTyp: -10024/-10025/-10026
			pmax(SINT): 127 pmax(INT): 32767 pmax(DINT): 2147483647
-35	-7	245	ErrorFlg: 0
-35	+7	-245	ErrorNr: 0
35	-7	-245	ErrorTyp: 0
35	+7	245	
product less than mmax		mmax	ErrorFlg: 1 ErrorNr: 3 ErrorTyp: -10024/-10025/-10026
			mmax(SINT): -128 mmax(INT): -35768 mmax(DINT): -2147483648

DIV The arithmetic operation '**Division**' is used to perform the division of two numbers of the same type. The result is the integer part, and of the same type as the input variables.

NOTE: Assigning default-initialized variables to the function will lead to an error.

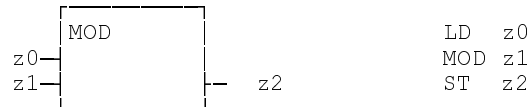


Example: Result ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
-35	-6	5	ErrorFlg: 0 pmax (SINT): 127
-35	+8	-4	ErrorNr: 0 pmax (INT): 32767
35	-8	-4	ErrorTyp: 0 pmax (DINT): 2147483647
35	+6	5	mmax (SINT): -128
-7	-35	0	mmax (INT): -32768
+7	-35	0	mmax (DINT): -2147483648
35	0	pmax	ErrorFlg: 1
0	0	pmax	ErrorNr: 5
-35	0	mmax	ErrorTyp: -10034/-10035/-10036

MOD The arithmetic operation '**Modulo division**' is used to perform the Modulo division of two numbers of the same type (remainder from the division 'DIV'). The result is of the same type as the input variables (not defined for REAL).

NOTE: Assigning default-initialized variables to the function will lead to an error.



Example: Result ranges (data types: SINT, INT, DINT)

z0	z1	z2	Error messages
-35	-6	-5	ErrorFlg: 0 pmax (SINT): 127
-35	+8	-3	ErrorNr: 0 pmax (INT): 32767
35	-8	3	ErrorTyp: 0 pmax (DINT): 147483647
35	+6	5	mmax (SINT): -128
-7	-35	-7	mmax (INT): -32768
+7	-35	7	mmax (DINT): -2147483648
-7	0	mmax	ErrorFlg: 1, ErrorNr: 5
+7	0	pmax	ErrorTyp: -10044/-10045/-10046

5.5.3 Comparators

'Comparators' are instructions that are used for comparing operands and expressions with the current value: the magnitude of two numbers of the same type; the equality/inequality of two bit strings of the same type; the alphabetic order of two characters (CHAR) or character strings (STRING); time values (TIME).

The result is a Boolean value. The error variables:

ErrorFlg,

ErrorNr, and

ErrorTyp

will not be influenced since an error is not possible.

GT The Boolean output of the comparator '**Greater than**' is '1' if the variable/constant at the upper input (current value) is greater than the variable/constant at the lower input. Otherwise, the output will be '0'.

Comparing numbers:

LAD/FBD	IL	Type												
	LD z0 GT z1 ST b0	z0, z1: Numbers b0: BOOL												
<table border="1"> <thead> <tr> <th>z0</th> <th>z1</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>3</td> <td>1</td> </tr> <tr> <td>5</td> <td>5</td> <td>0</td> </tr> <tr> <td>3</td> <td>5</td> <td>0</td> </tr> </tbody> </table>	z0	z1	b0	5	3	1	5	5	0	3	5	0		
z0	z1	b0												
5	3	1												
5	5	0												
3	5	0												

GE The Boolean output of the comparator '**Greater than or equal to**' is '1' if the variable/constant at the upper input (current value) is greater than or equal to the variable/constant at the lower input. Otherwise, the output will be '0'.

Comparing numbers:USINT, UINT, UDINT,SINT, INT, DINT, REAL

LAD/FBD	IL	Type												
	LD z0 GE z1 ST b0	z0, z1: Numbers b0: BOOL												
<table border="1"> <thead> <tr> <th>z0</th> <th>z1</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>3</td> <td>1</td> </tr> <tr> <td>5</td> <td>5</td> <td>1</td> </tr> <tr> <td>3</td> <td>5</td> <td>0</td> </tr> </tbody> </table>	z0	z1	b0	5	3	1	5	5	1	3	5	0		
z0	z1	b0												
5	3	1												
5	5	1												
3	5	0												

EQ The Boolean output of the comparator 'Equal to' is '1' if the variable/constant at the upper input (current value) is equal to the variable/constant at the lower input. Otherwise, the output will be '0'.

Comparing numbers:USINT, UINT, UDINT,SINT, INT, DINT, REAL

LAD/FBD	IL	Type												
	LD z0 EQ z1 ST b0	z0: INT z1: INT b0: BOOL												
<table border="1"> <thead> <tr> <th>z0</th> <th>z1</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>3</td> <td>0</td> </tr> <tr> <td>5</td> <td>5</td> <td>1</td> </tr> <tr> <td>3</td> <td>5</td> <td>0</td> </tr> </tbody> </table>	z0	z1	b0	5	3	0	5	5	1	3	5	0		
z0	z1	b0												
5	3	0												
5	5	1												
3	5	0												

NE The Boolean output of the comparator 'Not equal to' is '1' if the variable/constant at the upper input (current value) is not equal to the variable/constant at the lower input. Otherwise, the output will be '0'.

Comparing numbers:USINT, UINT, UDINT,SINT, INT, DINT, REAL

LAD/FBD	IL	Type												
	LD z0 NE z1 ST b0	z0: INT z1: INT b0: BOOL												
<table border="1"> <thead> <tr> <th>z0</th> <th>z1</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>3</td> <td>1</td> </tr> <tr> <td>5</td> <td>5</td> <td>0</td> </tr> <tr> <td>3</td> <td>5</td> <td>1</td> </tr> </tbody> </table>	z0	z1	b0	5	3	1	5	5	0	3	5	1		
z0	z1	b0												
5	3	1												
5	5	0												
3	5	1												

LE The Boolean output of the comparator 'Less than or equal to' is '1' if the variable/constant at the upper input (current value) is less than or equal to the variable/constant at the lower input. Otherwise, the output will be '0'.

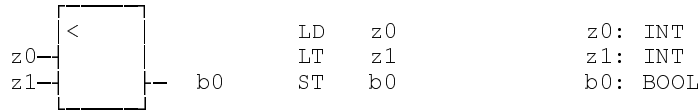
Comparing numbers:USINT, UINT, UDINT,SINT, INT, DINT, REAL

LAD/FBD	IL	Type												
	LD z0 LE z1 ST b0	z0, z1: Numbers b0: BOOL												
<table border="1"> <thead> <tr> <th>z0</th> <th>z1</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>3</td> <td>0</td> </tr> <tr> <td>5</td> <td>5</td> <td>1</td> </tr> <tr> <td>3</td> <td>5</td> <td>1</td> </tr> </tbody> </table>	z0	z1	b0	5	3	0	5	5	1	3	5	1		
z0	z1	b0												
5	3	0												
5	5	1												
3	5	1												

LT The Boolean output of the comparison function **'Less than'** is '1' if the variable/constant at the upper input (current value) is less than the variable/constant at the lower input. Otherwise, the output will be '0'.

Comparing numbers: USINT, UINT, UDINT, SINT, INT, DINT, REAL

LAD/FBD	IL	Type
---------	----	------



z0	z1	b0
5	3	0
5	5	0
3	5	1

5.6 Functions

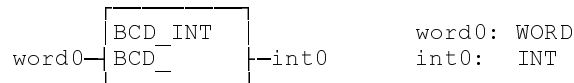
5.6.1 Standard Functions

The standard functions and standard operations for the INDRAMAT control system result from EN 61131-3. Standard functions are available in all programming languages of the system. They may directly be used, but cannot be modified.

Import not necessary

BCD_INT The **'BCD_INT'** function converts a BCD word into an INT number. The two nibbles are converted separately and the result is superimposed. The result is invalid if one of the nibbles shows one of the following assignments: 1010, 1011, 1100, 1101, 1110, 1111.

Error message: ErrorFlg: 1, ErrorNr: 4, ErrorTyp: -52,



word0	int0
0000 0000 0000 0000	0
0000 0000 0000 0001	1
...	...
0000 0000 0000 1001	9
0000 0000 0000 1010	invalid
...	...
0000 0000 0000 1111	invalid
0000 0000 0001 0000	10
...	...
1001 1001 1001 1001	9999
and so on	invalid

BYT_CHAR The 'BYT_CHAR' function converts a byte into a character of the extended ASCII character set.



Example 1:

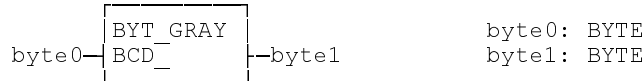
```
byte11: 0011 0001 char1: "1"
```

Example 2:

```
byte10: 1100 1101 char0: "¶"
```

Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

BYT_GRAY The 'BYT_GRAY' function clears the higher-order nibble of the input variable, and converts the lower-order nibble according to the following table.

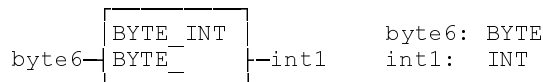


byte0	byte1
**** 0000	0000 0000
**** 0001	0000 0001
**** 0010	0000 0011
**** 0011	0000 0010
**** 0100	0000 0110
**** 0101	0000 0111
**** 0110	0000 0101
**** 0111	0000 0100
**** 1000	0000 1100
**** 1001	0000 1101
**** 1010	0000 1111
**** 1011	0000 1110
**** 1100	0000 1010
**** 1101	0000 1011
**** 1110	0000 1001
**** 1111	0000 1000

 An assignment different from 0000 leads to an error message:
 ErrorFlg: 1,
 ErrorNr: 4,
 ErrorTyp: -50,

BYTE_INT

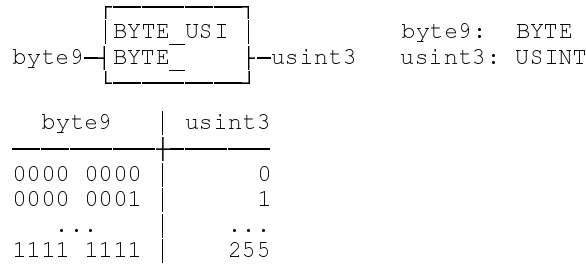
The 'BYTE_INT' function produces an INT number from a byte.



byte6	int1
0000 0000	0
0000 0001	1
...	...
1111 1111	255

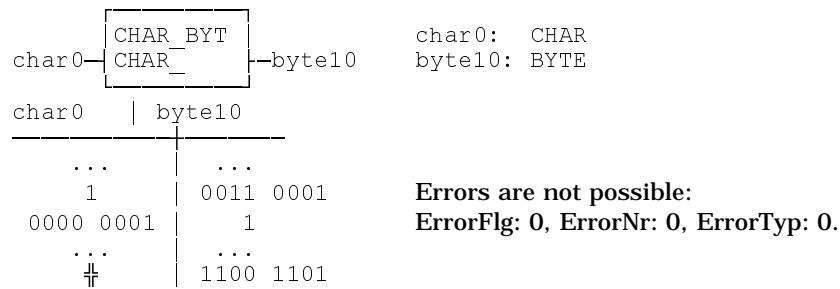
Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

BYTE_USI The 'BYTE_USI' function converts a byte into a USINT number.

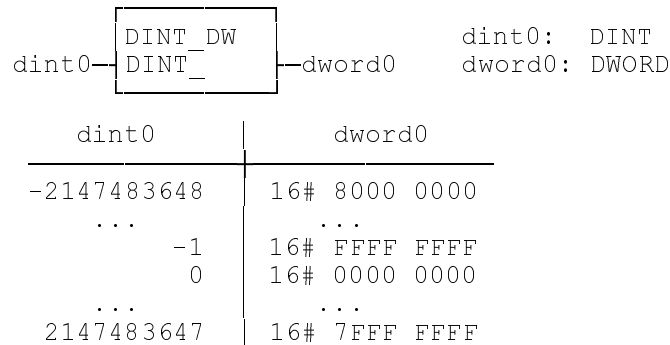


Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

CHAR_BYT The 'CHAR_BYT' converts a character of the extended ASCII character set into a byte.

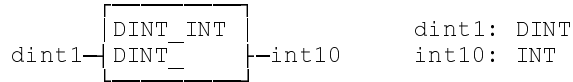


DINT_DW The 'DINT_DW' function converts a DINT number into a DWORD.



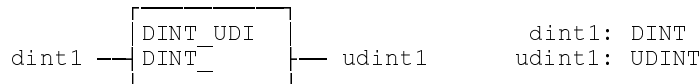
Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

DINT_INT The 'DINT_INT' function converts a DINT number into an INT number.



dint1	int10	Error messages
-2147483648	-32768	ErrorFlg: 1
...	...	ErrorNr: 3
-32769	-32768	ErrorTyp: -154
-32768	-32768	ErrorFlg: 0
...	...	ErrorNr: 0
32767	32767	ErrorTyp: 0
32768	32767	ErrorFlg: 1
...	...	ErrorNr: 3
2147483647	32767	ErrorTyp: -154

DINT_UDI The 'DINT_UDI' function converts the DINT data type into the UDINT data type. Type conversion generates an error if a negative value is applied.



dint1	udint1	Error messages
-2147483648	0	ErrorFlg: 1
...	...	ErrorNr: 3 underlimit
-1	0	ErrorTyp: -171
0	0	ErrorFlg: 0
...	...	ErrorNr: 0
2147483647	2147483647	ErrorTyp: 0

DINT_TIM The 'DINT_TIM' function converts a DOUBLE INTEGER value of the unit milliseconds into a time value. When high input values exceed the maximum time, the time is set to its maximum value of T#23d23h59m59s998ms, and the following error message is issued:

'Overflow in DINT_TIM conversion'

If a negative DOUBLE INTEGER value is applied to the 'DINT_TIM' function, the time is set to T#0s, and the following error message is issued:

'Negative value transferred to DINT_TIM'

Odd values in the millisecond specifications are rounded up to the next even value (INTEGER for millisecond 1; TIME T#2ms time value).

dint2	time1	Error messages
-2147483648	0ms	ErrorFlg: 1
...	...	ErrorNr: 3
-1	0ms	ErrorTyp: -156
0	0ms	ErrorFlg: 0
...	...	ErrorNr: 0
2073599998	23d23h59m59s998ms	ErrorTyp: 0
2073599999	23d23h59m59s998ms	ErrorFlg: 1
...	...	ErrorNr: 2
2147483647	23d23h59m59s998ms	ErrorTyp: -156

DW_DINT The 'DW_DINT' function converts a DINT number into a DWORD.

dword0	dint0
16# 8000 0000	-2147483648
...	...
16# FFFF FFFF	-1
16# 0000 0000	0
...	...
16# 7FFF FFFF	2147483647

Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

GRAY_BYT The 'GRAY_BYT' function clears the higher-order nibble of the input variable, and converts the lower-order nibble according to the following table.

GRAY_BYT GRAY_	byte0: BYTE byte1: BYTE
byte0	byte1

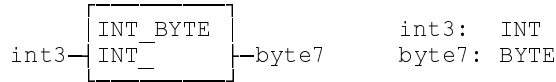
byte0	byte1	
**** 0000	0000 0000	
**** 0001	0000 0001	
**** 0010	0000 0011	****
**** 0011	0000 0010	An assignment different from 0000
**** 0100	0000 0111	leads to an error message:
**** 0101	0000 0110	
**** 0110	0000 0100	ErrorFlg: 1,
**** 0111	0000 0101	ErrorNr: 4,
**** 1000	0000 1111	ErrorTyp: -49,
**** 1001	0000 1110	
**** 1010	0000 1100	
**** 1011	0000 1101	
**** 1100	0000 1000	
**** 1101	0000 1001	
**** 1110	0000 1011	
**** 1111	0000 1010	

INT_BCD The 'INT_BCD' function converts the INT number into a BCD word.

INT_BCD INT_	int5: INT word2: WORD
int5	word2

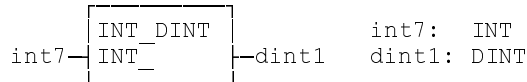
int4	word2	Error messages
-32768	0000 0000 0000 0000	int4 < 0
...	...	ErrorFlg: 1, ErrorNr: 3,
-1	0000 0000 0000 0000	ErrorTyp: -57
0	0000 0000 0000 0000	ErrorFlg: 0
1	0000 0000 0000 0001	ErrorNr: 0
...	...	ErrorTyp: 0
9999	1001 1001 1001 1001	
10000	0010 0111 0000 1111	int4 > 10000
...	...	ErrorFlg: 1, ErrorNr: 2,
32767	0010 0111 0000 1111	ErrorTyp: -57

INT_BYTE The 'INT_BYTE' function converts an INT number into a byte.



int3	byte7	Error messages
-32768	0000 0000	int3 < 0
...	...	ErrorFlg: 1, ErrorNr: 4, ErrorTyp: -55
-1	1111 1111	
0	0000 0000	ErrorFlg: 0
1	0000 0001	ErrorNr: 0
...	...	ErrorTyp: 0
255	1111 1111	
256	0000 0000	int3 > 255
...	...	ErrorFlg: 1, ErrorNr: 4, ErrorTyp: -55
32767	1111 1111	

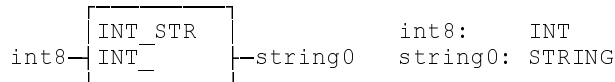
INT_DINT The 'INT_DINT' function converts an INT number into a DINT number.



int10	dint1
-32768	-32768
...	...
32767	32767

Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

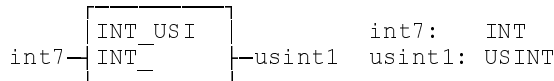
INT_STR The 'INT_STR' function converts an INT number into a character string (STRING).



int8	string0
-32768	'-32768'
-128	'-128'
0	'0'
32767	'32767'

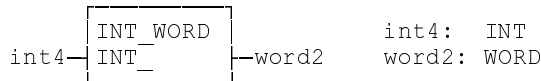
Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

INT_USI The 'INT_USI' function converts an INT number into an USINT number.



int7	usint1	Error messages
-32768	0	int7 < 0
...	...	ErrorFlg: 1, ErrorNr: 3, ErrorTyp: -59
-1	0	
0	0	ErrorFlg: 0
1	1	ErrorNr: 0
...	...	ErrorTyp: 0
255	255	
256	255	int7 > 255
...	...	ErrorFlg: 1, ErrorNr: 2, ErrorTyp: -59
32767	255	

INT_WORD The 'INT_WORD' function converts an INT number into a word.



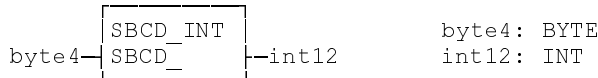
int4	word2
-32768	1000 0000 0000 0000
-32767	1000 0000 0000 0001
...	...
-1	1111 1111 1111 1111
0	0000 0000 0000 0000
1	0000 0000 0000 0001
...	...
32767	0111 1111 1111 1111

Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

SBCD_INT The 'SBCD_INT' function converts a BCD byte into an INT number. The two nibbles are converted separately and the result is superimposed. The result is invalid if one of the nibbles shows one of the following assignments:

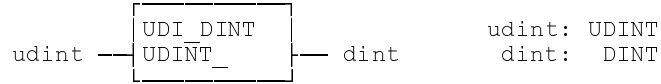
1010, 1011, 1100, 1101, 1110, 1111.

Error message: ErrorFlg: 1, ErrorNr: 4, ErrorTyp: -51,



byte4	int12
0000 0000	0
0000 0001	1
...	...
0000 1001	9
0000 1010	invalid
...	...
0000 1111	invalid
0001 0000	10
...	...
1001 1001	99

UDI_DINT The 'UDI_DINT' function converts an UDINT number into a DINT number.



uint	dint	ErrorFlg	ErrorNr	ErrorTyp
0	0	0	0	0
1	1	0	0	0
...	...	0	0	0
2147483647	2147483647	0	0	0
2147483648	2147483647	1	1	0
...	...	2	2	0
4294967295	2147483647	-172	-172	0

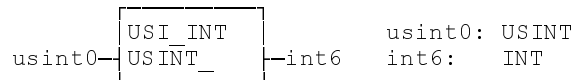
USI_BYTE The 'USI_BYTE' function converts an USINT number into a byte.



usint2	byte8
0	0000 0000
1	0000 0001
...	...
255	1111 1111

Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

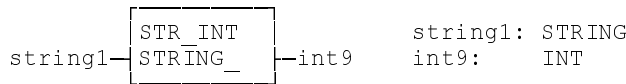
USI_INT The 'USI_INT' function converts a USINT number into an INT number.



usint0	int6
0	0
1	1
...	...
255	255

Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

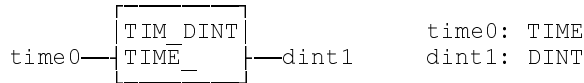
STR_INT The 'STR_INT' function converts a character string (STRING) into an INT number.



string1	int9	Error messages
'-999999'	-32768	ErrorFlg: 1
...	...	ErrorNr: 3
'-32769'	-32768	ErrorTyp: -141
'-32768'	-32768	ErrorFlg: 0
...	...	ErrorNr: 0
'32767'	32767	ErrorTyp: 0
'32768'	32767	ErrorFlg: 1
...	...	ErrorNr: 2
'999999'	32767	ErrorTyp: -141
'Paul'	0	ErrorFlg: 1
...	...	ErrorNr: 4
...	...	ErrorTyp: -141

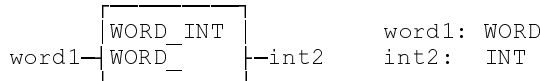
An empty STRING " is converted into the INT number '0'. There will be no error message.

TIM_DINT The "TIM_DINT" function converts a time value into an double INTEGER value of the unit milliseconds.



time0	dint1	Error messages
0ms	0	ErrorFlg: 0
...	...	ErrorNr: 0
23d23h59m59s998ms	2073599998	ErrorTyp: 0
greater than 23d23h59m59s998ms	2073599998	ErrorFlg: 1
		ErrorNr: 2
		ErrorTyp: -157

WORD_INT The 'WORD_INT' function converts a word into an INT number.

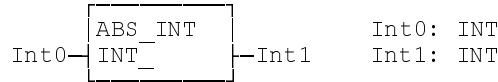


word1	int2
0000 0000 0000 0000	0
0000 0000 0000 0001	1
...	...
0111 1111 1111 1111	32767
1000 0000 0000 0000	-32768
1000 0000 0000 0001	-32767
...	...
1111 1111 1111 1111	-1

Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

5.6.2 Numeric Functions

ABS_INT As a result, the numeric function 'ABS_INT' returns the magnitude of the INTEGER number that is applied to the input.
(Only available for INT numbers)



Examples: for the result ranges

Int0	Int1	Error messages
32767	32767	ErrorFlg: 0
...	...	ErrorNr: 0
-5	5	ErrorTyp: 0
-32767	32767	
-32768	32767	ErrorFlg: 1
		ErrorNr: 2
		ErrorTyp: -69

SIGN_INT As a result, the numeric function 'SIGN_INT' returns the sign of the INTEGER number that is applied to the input.
(Only available for INT numbers)



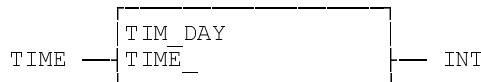
Examples: for the result ranges

Int0	sign
-35	0
0	1
35	1

Errors are not possible: ErrorFlg: 0, ErrorNr: 0, ErrorTyp: 0.

5.6.3 Functions for Time-to-Integer Conversion

TIM_DAY Conversion TIME (unit day) - INTEGER



TIM_HOUR Conversion TIME (unit hour) - INTEGER



TIM_MIN Conversion TIME (unit minute) - INTEGER



TIM_SEC Conversion TIME (unit second) - INTEGER



TIM_MS Conversion TIME (unit millisecond) - INTEGER



TIME : TIME; time value that is to be converted
 Function result:
 INT; Depending on function: day, hour, minute, second, millisecond

5.6.3.1 Method of operation of the time-to-integer conversion

The TIM_DAY, TIM_HOUR, TIM_MIN, TIM_SEC and TIME_MS functions are provided with a time value. Depending on the function, the section day, hour, minute, second, or millisecond is taken from the time value and made available as an integer value at the function output.

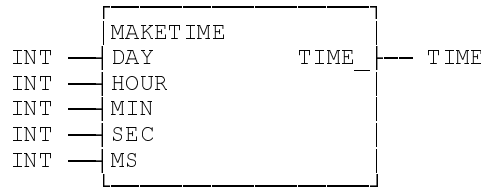
5.6.3.2 Error handling of the time-to-integer conversion

The TIM_DAY, TIM_HOUR, TIM_MIN, TIM_SEC and TIM_MS functions do not generate errors.
 Error type of the function blocks

Conversion TIM_xx	
TIM_DAY:	205
TIM_HOUR:	206
TIM_MIN:	207
TIM_SEC:	208
TIM_MS:	209

5.6.4 INTEGER-to-TIME Conversion

MAKETIME



DAY: INT; day
 HOUR: INT; hours
 MIN: INT; minutes
 SEC: INT; seconds
 MS: INT; milliseconds
 Function result: TIME; Converted time value

Method of operation

The 'MAKETIME' function converts the INTEGER values of day, hours, minutes, seconds, and milliseconds into a time value. The input values are summed up according to their units. The function generates an error message if there are negative input values or if the maximum time value is exceeded.

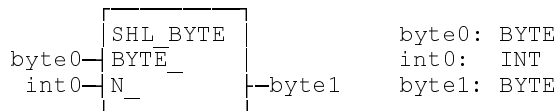
Function result: = MS + 1000 * (SEC + 60 * (MIN + 60 * (HOUR + 24 * DAY)

Error handling

5.6.5 Bit string functions

**SHL_BYTE,
SHL_WORD,
SHL_DW**

The 'SHL_BYTE', 'SHL_WORD', 'SHL_DW' bit string functions permit the bit string that is applied to the upper function input to be shifted to the left, bit by bit. The free bits are filled with '0'.



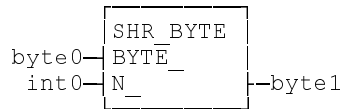
Examples: for the result ranges

byte0	int0	byte1	ErrorFlg	ErrorNr	ErrorTyp
00000000	0 ... 7	00000000	0	0	0
...	0 ... 7	...	0	0	0
11001001	3	01001000	0	0	0
11001001	4	10010000	0	0	0
...	0 ... 7	...	0	0	0
11111111	0 ... 7	...	0	0	0
any	negative	byte0	1	1	-70
any	>7	byte0	1	1	-70

The second input (type INT) defines the number of shift cycles. With a variable of k bits, a minimum of 0 and a maximum of (k-1) shift cycles are permitted. The command will not be executed, and the error variables ErrorFlg, ErrorNr and ErrorTyp will be set if the number of shift cycles is negative or greater than (k-1).

**SHR_BYTE,
 SHR_WORD,
 SHR_DW**

The 'SHR_BYTE', 'SHR_WORD', 'SHR_DW' bit string functions permit the bit string that is applied to the upper function input to be shifted to the right, bit by bit. The free bits are filled with '0'.



valid are:

$$0 \leq \text{int0} \leq 7$$

byte0: BYTE
 int0: INT
 byte1: BYTE

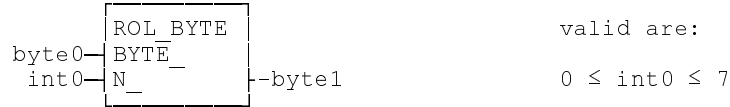
Examples: for the result ranges

byte0	int0	byte1	ErrorFlg	ErrorNr	ErrorTyp
00000000	0 ... 7	00000000	0	0	0
...	0 ... 7	...	0	0	0
11001001	3	00011001	0	0	0
11001001	4	00001100	0	0	0
...	0 ... 7	...	0	0	0
11111111	0 ... 7	...	0	0	0
any	negative	byte0	1	1	-72
any	>7	byte0	1	1	-72

The second input (type INT) defines the number of shift cycles. With a variable of k bits, a minimum of 0 and a maximum of (k-1) shift cycles are permitted. The command will not be executed, and the error variables ErrorFlg, ErrorNr and ErrorTyp will be set if the number of shift cycles is negative or greater than (k-1).

**ROL_BYTE,
 ROL_WORD,
 ROL_DW**

The 'ROL_BYTE', 'ROL_WORD', 'ROL_DW' bit string functions permit the bit string that is applied to the upper function input to be rotated to the left, bit by bit. The outer left bit rotates to the right outside.



byte0: BYTE
 int0: INT
 byte1: BYTE

Examples: for the result ranges

byte0	int0	byte1	ErrorFlg	ErrorNr	ErrorTyp
00000000	0 ... 7	00000000	0	0	0
...	0 ... 7	...	0	0	0
11001001	3	01001110	0	0	0
11001001	4	10011100	0	0	0
...	0 ... 7	...	0	0	0
11111111	0 ... 7	...	0	0	0
any	negative	byte0	1	1	-74
any	>7	byte0	1	1	-74

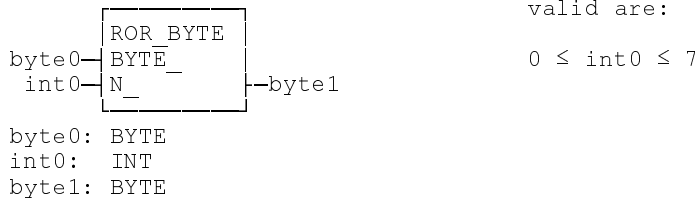
The second input (type INT) defines the number of shift cycles.

With a variable of k bits, a minimum of 0 and a maximum of (k-1) shift cycles are permitted.

The command will not be executed, and the error variables ErrorFlg, ErrorNr and ErrorTyp will be set if the number of shift cycles is negative or greater than (k-1).

**ROR_BYTE,
 ROR_WORD,
 ROR_DW**

The 'ROR_BYTE', 'ROR_WORD', 'ROR_DW' bit string functions permit the bit string that is applied to the upper function input to be rotated to the right, bit by bit. The outer right bit rotates to the left outside.

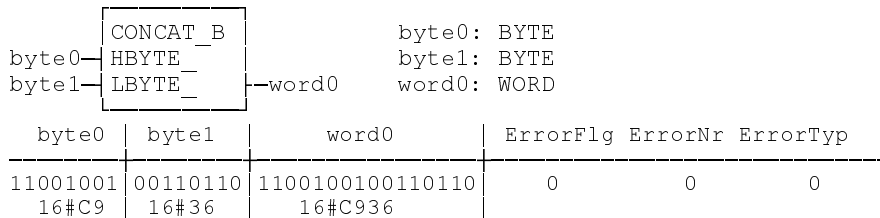


Examples: for the result ranges

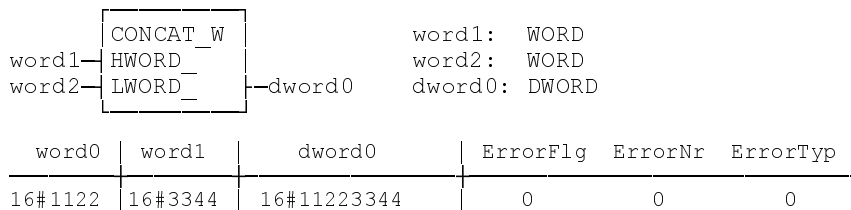
byte0	int0	byte1	ErrorFlg	ErrorNr	ErrorTyp
00000000	0 ... 7	00000000	0	0	0
...	0 ... 7	...	0	0	0
11001001	3	00011001	0	0	0
11001001	4	10001100	0	0	0
...	0 ... 7	...	0	0	0
11111111	0 ... 7	...	0	0	0
any	negative	byte0	1	1	-76
any	>7	byte0	1	1	-76

The second input (type INT) defines the number of shift cycles. With a variable of k bits, a minimum of 0 and a maximum of (k-1) shift cycles are permitted. The command will not be executed, and the error variables ErrorFlg, ErrorNr and ErrorTyp will be set if the number of shift cycles is negative or greater than (k-1).

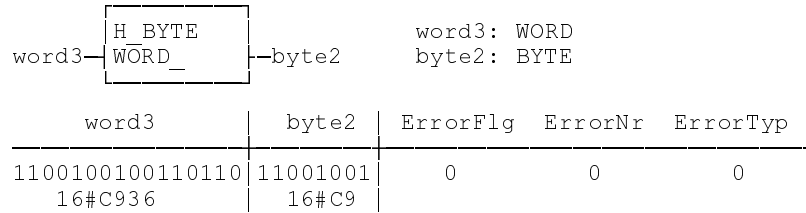
CONCAT_B The 'CONCAT_B' bit string function combines the two adjacent bytes to form a word. The byte at the upper input becomes the high byte, the byte at the lower input becomes the low byte of the word.



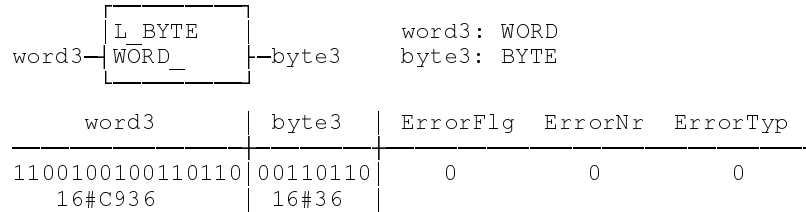
CONCAT_W The 'CONCAT_W' bit string function combines the two adjacent words to form a double word. The word at the upper input becomes the high word, the word at the lower input becomes the low word of the DWORD.



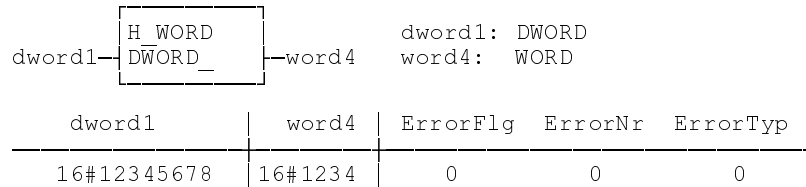
H_BYTE The 'H_BYTE' bit string function removes the most significant byte of the word that is applied to the input.



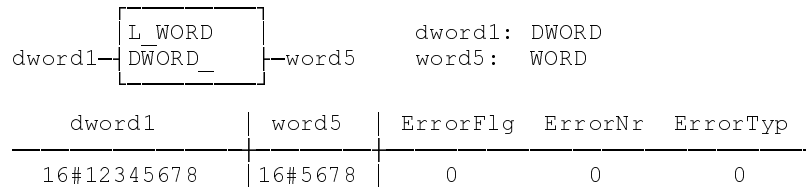
L_BYTE The 'L_BYTE' bit string function removes the least significant byte of the word that is applied to the input.



H_WORD The 'H_WORD' bit string function removes the most significant word of the double word that is applied to the input.

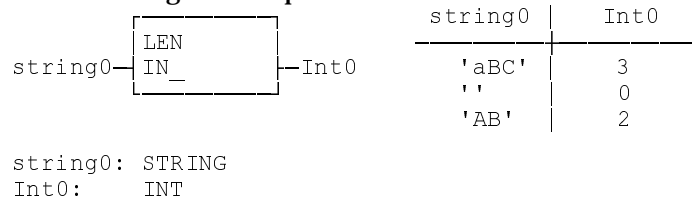


L_WORD The 'L_WORD' bit string function removes the least significant word of the double word that is applied to the input.



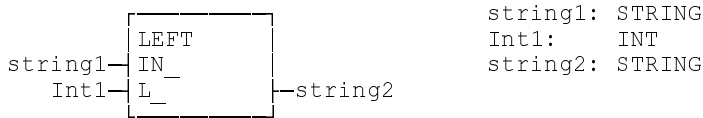
5.6.6 Character String Functions

LEN The 'LEN' character string function determines the length of a character string. An error message is not possible.



LEFT The **'LEFT'** character string function returns the 'L' characters that are furthest left. Error messages are issued if:

- **L < 0** negative number of characters
- **LEN(string1)** limit exceeded

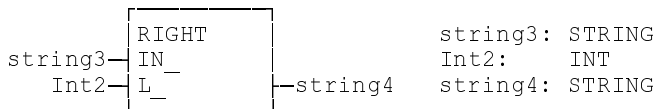


Example:

string1	Int1	string2	ErrorFlg	ErrorNr	ErrorTyp
*	...-1	' '	1	1	-143
' '	0	' '	0	0	0
'abcde'	0	' '	0	0	0
'abcde'	2	'ab'	0	0	0
'abcde'	5	'abcde'	0	0	0
'abcde'	6	'abcde'	1	1	-143
limit exceeded					

RIGHT The **'RIGHT'** character string function returns the 'L' characters that are furthest right. Error messages are issued if:

- **L < 0** negative number of characters
- **L > LEN(string3)** limit exceeded



Examples:

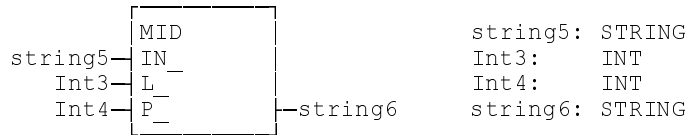
string3	Int1	string4	ErrorFlg	ErrorNr	ErrorTyp
*	...-1	' '	1	1	-144
' '	0	' '	0	0	0
'abcde'	0	' '	0	0	0
'abcde'	2	'de'	0	0	0
'abcde'	5	'abcde'	0	0	0
'abcde'	6	'abcde'	1	1	-144
limit exceeded					

MID The 'MID' character string function determines the next 'L' characters to the right of position 'P'.

Error messages are issued if:

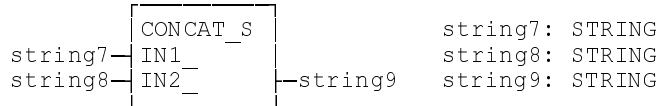
- L < 0 negative number of characters
- P < 1 display before position 1
- L > LEN(string5)+1-P, limit exceeded

NOTE: Assigning default-initialized variables to the function will lead to an error.



string5	Int3	Int4	string6	ErrorFlg	ErrorNr	ErrorTyp
*	...-1	*	''	1	1	-145
*	*	...1	''	1	1	-145
'abcde'	0	1	''	0	0	0
'abcde'	2	1	'ab'	0	0	0
'abcde'	5	1	'abcde'	0	0	0
'abcde'	6	1	'abcde'	1	1	-145
limit exceeded						
'abcde'	2	2	'bc'	0	0	0
'abcde'	3	2	'bcd'	0	0	0
'abcde'	4	2	'bcde'	0	0	0

CONCAT_S The 'CONCAT_S' character string function enables the lower character string to be appended to the upper one.

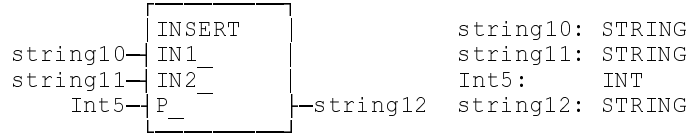


string7	string8	string9
''	''	''
'abc'	''	'abc'
'abc'	'de'	'abcde'

INSERT

The **'INSERT'** character string function enables the lower character string to be inserted, after the position 'P', into the upper character string. Error messages are issued if:

- P < 0 insertion before position 0
- P > LEN(string5) limit exceeded

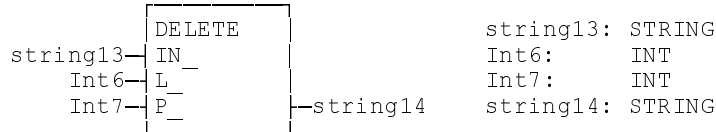


string10	string11	Int5	string12	ErrorFlg	ErrorNr	ErrorTyp
*	*	..-1	*	1	1	-147
''	''	0	''	0	0	0
''	'de'	0	'de'	0	0	0
'abc'	'de'	0	'deabc'	0	0	0
'abc'	'de'	1	'adebc'	0	0	0
'abc'	'de'	2	'abdec'	0	0	0
'abc'	'de'	3	'abcde'	0	0	0
''	'de'	1	string10	1	1	-147
'abc'	'de'	4	string10	1	1	-147
limit exceeded						

DELETE The **'DELETE'** character string function deletes 'L' characters from, and including, position 'P' to the right.

Error messages are issued if:

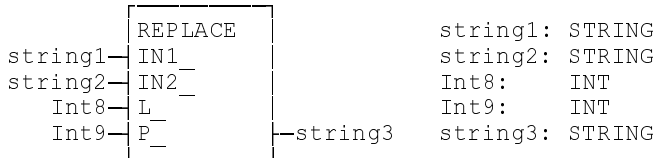
- the function is applied to an empty character string
- L<0 deletion of a negative number
- P<1 deletion before position 1
- L> LEN(string13)+1-P deletion beyond the string boundary



string13	Int6	Int7	string14	ErrorFlg	ErrorNr	ErrorTyp
' '	*	*	' '	1	1	-148
*	...-1	*	' '	1	1	-148
*	*	...0	' '	1	1	-148
'abcde'	0	1	'abcde'	0	0	0
'abcde'	1	1	'bcde'	0	0	0
'abcde'	2	1	'cde'	0	0	0
'abcde'	4	1	'e'	0	0	0
'abcde'	5	1	' '	0	0	0
'abcde'	6	1	' '	1	1	-148
delet. beyond boundary						
'abcde'	0	2	'abcde'	0	0	0
'abcde'	1	2	'acde'	0	0	0
'abcde'	2	2	'ade'	0	0	0
'abcde'	3	2	'ae'	0	0	0
'abcde'	4	2	'a'	0	0	0
'abcde'	5	2	' '	1	1	-148
delet. beyond boundary						

REPLACE Starting from position 'P', the **'REPLACE'** character string function replaces 'L' characters of the upper character string by the lower character string. Error messages are issued if:

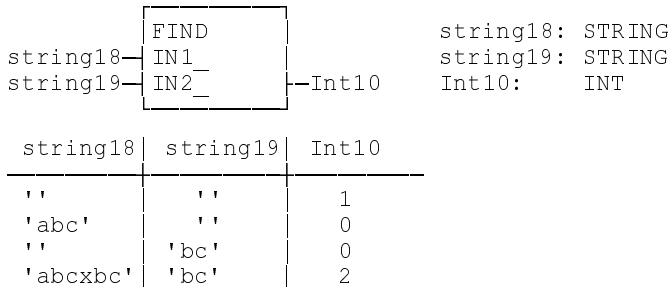
- the function is applied to an empty character string
- L<0 replacing a negative number
- P<1 replacing before position 1
- L> LEN(string1)+1-P replacing exceeds boundary



string1	string2	Int8	Int9	string3	ErrorFlg	ErrorNr	ErrorTyp
''	*	*	*	''	1	1	-149
*	*	...-1	*	''	1	1	-149
*	*	*	...0	''	1	1	-149
'abcde'	'xyz'	0	1	'xyzabcde'	0	0	0
'abcde'	'xyz'	2	1	'xyzcde'	0	0	0
'abcde'	'xyz'	4	1	'xyze'	0	0	0
'abcde'	'xyz'	5	1	'xyz'	0	0	0
'abcde'	'xyz'	6	1	''	1	1	-149
limit exceeded							
'abcde'	'xyz'	2	2	'axyzde'	0	0	0
'abcde'	'xyz'	3	2	'axyze'	0	0	0
'abcde'	'xyz'	4	2	'axyz'	0	0	0
'abcde'	'xyz'	0	5	'abcdexyz'	0	0	0

LEN(string3) > 255: ErrorFlg 1 ErrorNr 239 ErrorTyp -149

FIND The **'FIND'** character string function determines how often the lower character string appears in the upper one. Errors are not possible.



string18	string19	Int10
''	''	1
'abc'	''	0
''	'bc'	0
'abcxbc'	'bc'	2

5.7 Firmware Functions

The firmware functions that are available for a specific controller type are explained in the controller interface description.

5.8 User Functions

The programming system enables the user to write functions that may be employed as reusable units that supplement the standard and firmware functions. The user functions may import other user functions and use them like they use standard or firmware functions.

5.8.1 Import rules

A function may utilize standard functions, firmware functions, and user functions.

<p>The required function is a standard function or a firmware function</p>	<p>The required function is a user function</p>
<ul style="list-style-type: none"> • It is contained in the programming system's library and, consequently, known. • It does not need a permanently reserved memory space. • Thus, it can be used easily. 	<ul style="list-style-type: none"> • It is not contained in the library and, consequently, not known to the programming system • The function is made known by importing it in the program, in the FB or in the function where it is to be employed. • It does not need a permanently reserved memory space. • I can be used easily after it has been imported.

Nesting can be continued to any depth.

NOT ALLOWED is that function 'A' employs itself (recursive utilization), or that function 'A' employs function 'B' that employs function 'A', etc.

5.9 Function Blocks

A **'function block'** (FB) is a program organization unit that may have

- 1...n inputs,
- 1...m outputs, and
- internal variables

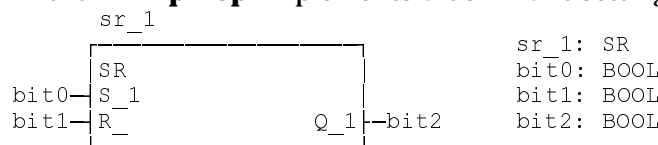
5.10 Standard Function Blocks

The standard function blocks for the INDRAMAT control system result from EN 61131-3. They are available in all programming languages of the system. They may be used, but cannot be modified. They need not be imported.

Errors are not possible when the individual function blocks are processed. Thus, ErrorFlg, ErrorNr and ErrorReg are always '0'. Import not necessary

5.10.1 Flip-Flops

SR - flip-flop The **'SR - flip-flop'** implements a dominant setting of the memory.



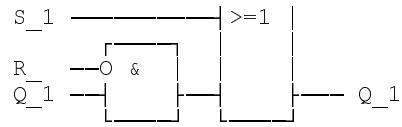
Truth table

bit0	bit1	old bit2	new bit2
0	0	0	0
1	0	0	1

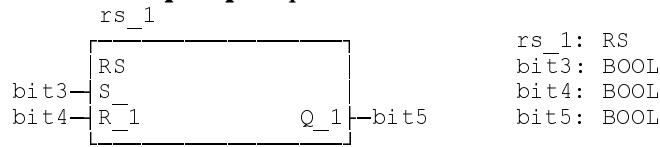
equivalent network:



0	1	0	0
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	0
1	1	1	1



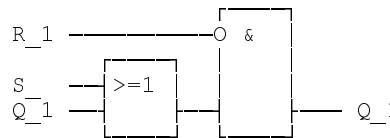
RS - flip-flop The 'RS - flip-flop' implements a dominant resetting of the memory.



Truth table

bit3	bit4	old bit5	new bit5
0	0	0	0
1	0	0	1
0	1	0	0
1	1	0	0
0	0	1	1
1	0	1	1
0	1	1	0
1	1	1	0

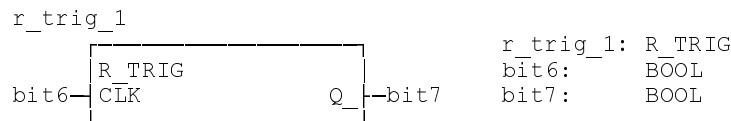
equivalent network:



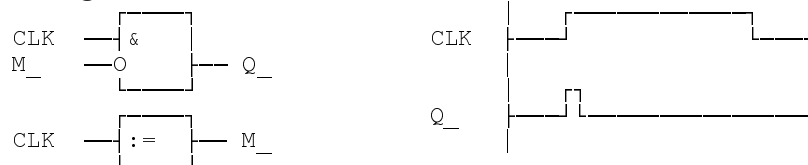
5.10.2 Edge evaluation for rising and falling edges 'R_TRIG', 'F_TRIG'

R_TRIG The 'Edge evaluation for rising edges' 'R_TRIG' function block implements a 0-1-0 transition at the output if the input signal transitions from '0' to '1'. The pulse duration results from the duration of the SPS cycle.

NOTE: After a 0-1 transition at the output it must be ensured that the function block is processed again in the next SPS cycle.

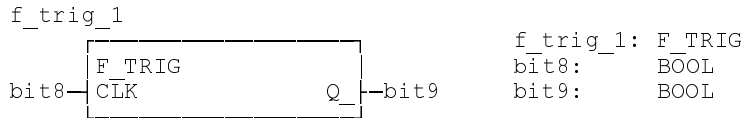


Internal implementation (in FBD) - timing

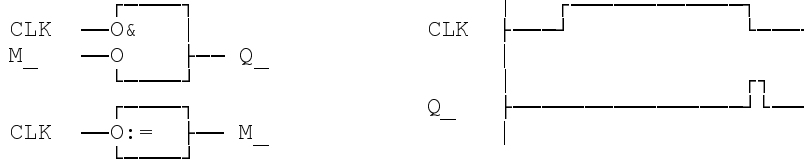


F_TRIG The 'Edge evaluation for falling edges' 'F_TRIG' function block implements a 0-1-0 transition at the output if the input signal transitions from '1' to '0'. The pulse duration results from the duration of the SPS cycle.

NOTE: After a 0-1 transition at the output it must be ensured that the function block is processed again in the next SPS cycle.



Internal implementation (in FBD)
 - timing

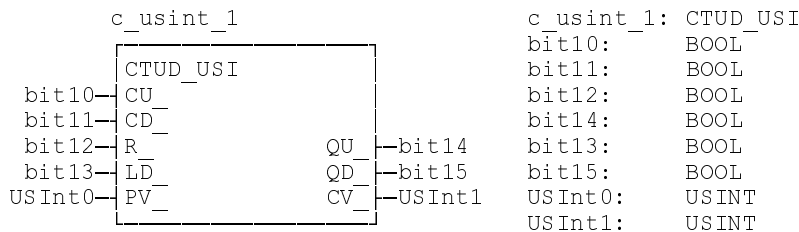


5.10.3 Up and down counters

CTUD_USI The 'CTUD_USI' (0 ... 255) counter has a dominating reset input 'R_'. A '1' at that input sets the counter to 'CV_=0'. Given the boundary condition 'R_=0', the preset value applied to 'PV_' is read as long as 'LD_' = '1'. The signal assignments at the 'CU_' and 'CD_' inputs are irrelevant in this case. Counting is possible with the boundary condition 'R_=0' and 'LD_=0': As long as CU_ = '1' and 'CV_ < 255', 'CV_' is incremented by '1' in each SPS cycle (difference to EN 61131-3!).

NOTE: If 'CU_' = '0', too, 'CV_' will be decremented by '1' in each SPS cycle as long as 'CD_' = '1' and 'CV_ > 0' (difference to EN 61131-3!).

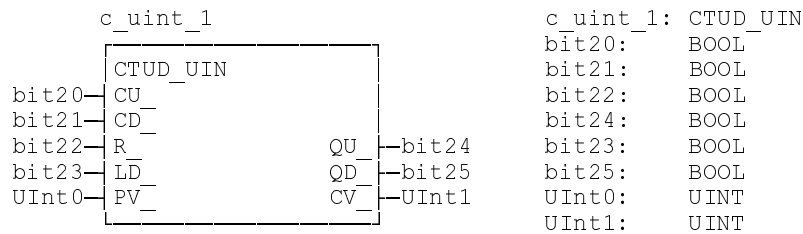
The output 'QU_' issues a '1' signal if 'CV_ ≥ PV'.
 The output 'QD_' issues a '1' signal ab, if 'CV_ = 0'.



CTUD_UIN The 'CTUD_UIN' (0 ... 65535) counter has a dominating reset input 'R_'. A '1' at that input sets the counter to 'CV_=0'. Given the boundary condition 'R_=0', the preset value applied to 'PV_' is read as long as 'LD_' = '1'. The signal assignments at the 'CU_' and 'CD_' inputs are irrelevant in this case. Counting is possible with the boundary condition 'R_=0' and 'LD_=0': As long as CU_ = '1' and 'CV_ < 65535', 'CV_' is incremented by '1' in each SPS cycle (difference to EN 61131-3!).

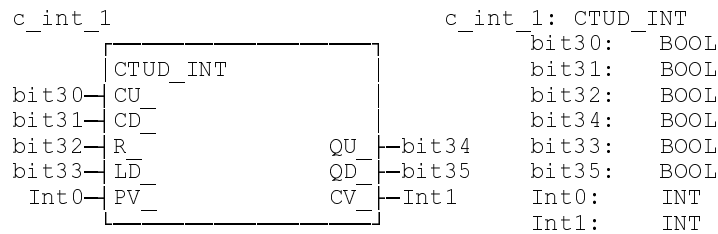
NOTE: If 'CU_' = '0', too, 'CV_' will be decremented by '1' in each SPS cycle as long as 'CD_' = '1' and 'CV_ > 0' (difference to EN 61131-3!).

The output 'QU_' issues a '1' signal if 'CV_ ≥ PV'.
 The output 'QD_' issues a '1' signal ab, if 'CV_ = 0'.



CTUD_INT The 'CTUD_INT' (-32768 ... 32767) counter has a dominating reset input. A '1' at that input sets the counter to 'CV_=0'. Given the boundary condition 'R_=0', the preset value applied to 'PV_' is read as long as 'LD_' = '1'. The signal assignments at the 'CU_' and 'CD_' inputs are irrelevant in this case. Counting is possible with the boundary condition 'R_=0' and 'LD_=0': As long as CU_ = '1' and 'CV_ < 32767', 'CV_' is incremented by '1' in each SPS cycle (difference to EN 61131-3!).

The output 'QU_' issues a '1' signal if 'CV_ ≥ PV'.
 The output 'QD_' issues a '1' signal ab, if 'CV_ = 0'.



5.10.4 Time stages for pulses; ON/OFF delay

TP Such a 'single pulse' appears at the output 'Q' when there is a 0-1 transition at the input 'IN'. The length of the input pulse is irrelevant. The time stage cannot be re-triggered. This means that additional impulses at the input will be ignored as long as the pulse exists at the output. The current pulse duration is counted at the 'ET' output. The value is retained until there is a 1-0 transition at the input.

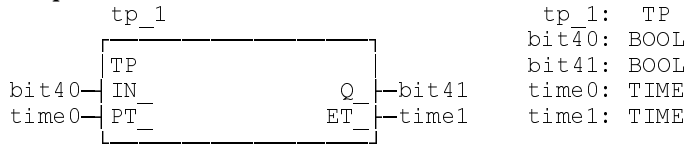
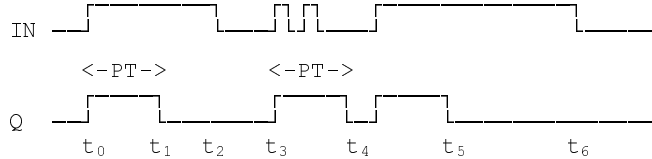


Diagram of the 'TP' time stage (impulse)



TON Delayed by 'PT', a '1' signal appears at the output 'Q' after a 0-1 transition has occurred at the input. The 'Q' output returns to '0' when the input returns to '0'. If the '1' signal at the input is shorter than 'PT', a '1' signal cannot be generated at the output. The ET output indicates the current delay time. The final value is retained until the signal at the input returns to '0'.

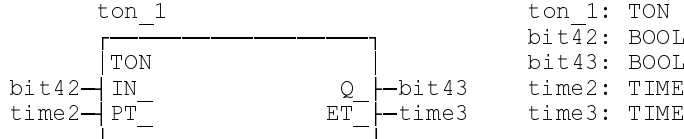
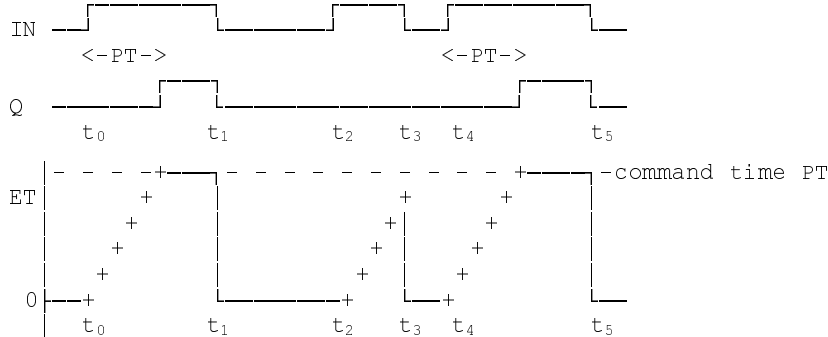


Diagram of the 'TON' time stage (ON-delayed)



TOFF

A '1' signal appears at the output 'Q' when there is a 0-1 transition at the input. When the input signal goes back from '1' to '0', the signal at the 'Q' output remains '1' for the time 'PT' before it returns to '0'. If the input signal returns to '1' inside the delay time 'PT', the process is continued without interruption (the time stage can be re-triggered). The ET output indicates the current delay time. The final value is retained until the signal at the input returns to '1'.

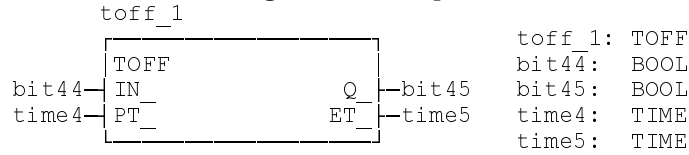
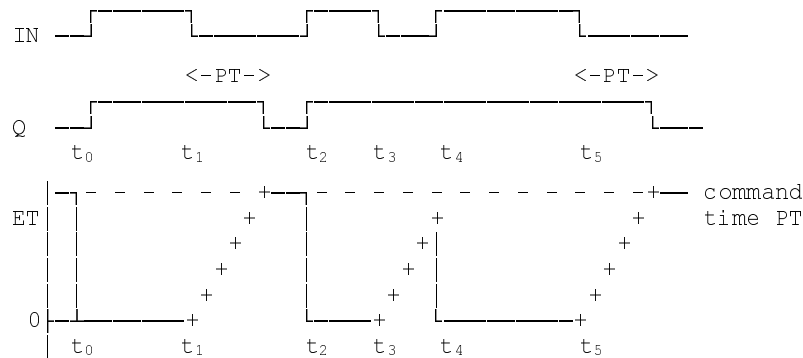
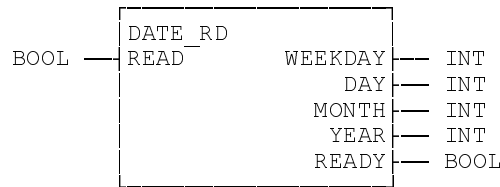


Diagram of the 'TOFF' time stage (OFF-delayed)



5.10.5 Function blocks for date and time

DATE_RD **Reading the Date**



READ: BOOL; 0 - FB not active
 1 - Activates reading the date

WEEKDAY: INT; Day of the week
 0 - Sunday
 1 - Monday
 2 - Tuesday
 3 - Wednesday
 4 - Thursday
 5 - Friday
 6 - Saturday

DAY: INT; Day (1...31)

MONTH: INT; Month (1...12)

YEAR: INT; Year (1980...2035)

READY: BOOL; 0 - Date not valid
 1 - Date valid


```
STOP:          INT;  number of stop bits
PROTOKOL:     INT;  protocol
HANDSH:       INT;  handshake
END_STRUCT;
END_TYPE COM;
```

Explanation of the parameter values that can be selected in the 'COM data type:

DEVICE: 0...999; Device number
 0 - Interface centrally on PCL module
 1...999 Interface on fiber optics loop device
 (the logic device address must be selected in the I/O assignments menu)

SERNR: 1...4 Interface number
 (enter 1 if there is only one interface on the I/O element)

BAUD: 1...21 Baud rate

	Baud	RECO	BTM 3/4	SIO 3	PAS	SOT	BTC 05
1	50	x	x	-	-	-	-
2	75	x	-	-	-	-	-
3	110	x	x	x	-	-	-
4	134.5	x	x	-	x	-	-
5	150	x	x	x	x	-	-
6	200	x	-	-	-	-	-
7	300	x	x	x	x	-	-
8	600	x	x	x	x	-	-
9	1050	-	-	-	-	-	-
10	1200	x	x	x	x	-	-
11	1800	x	x	-	x	-	-
12	2000	-	x	-	x	-	-
13	2400	x	x	x	x	-	-
14	3600	x	x	-	-	-	-
15	4800	x	x	x	x	-	-
16	7200	x	x	-	-	-	-
17	9600	x	x	x	x	x	x
18	19200	x	x	x	x	x	x
19	38400	-	x	x	-	-	x
20	57600	-	-	x	-	-	-
21	115200	-	-	x	-	-	-

DATA: 1...4 Number of user data bits

	Data bit	RECO	BTM 3/4	SIO 3	PAS	SOT	BTC 05
1	8	x	x	x	x	x	x
2	7	x	x	x	x	-	-
3	6	x	x	x	x	-	-
4	5	x	x	x	x	-	-

PARITY: 1...5 Type of parity check

	Parity	RECO	BTM 3/4	SIO 3	PAS	SOT	BTC 05
1	NONE	x	x	x	x	x	-
2	ODD	x	x	x	x	-	-
3	EVEN	x	x	x	x	-	x
4	MARK	x	x	-	-	-	-
5	SPACE	x	x	-	-	-	-

STOP: 1...3 Number of stop bits

	Stop bit	RECO	BTM 3/4	SIO 3	PAS	SOT	BTC 05
1	1	x	x	x	x	x	x
2	1,5	x	x	x	x	-	-
3	2	x	x	x	x	-	-

PROTOKOL: 1...3 Protocol type

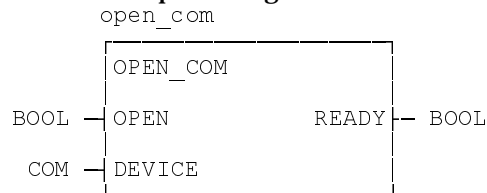
	Protocol	RECO	BTM 3/4	SIO 3	PAS	SOT	BTC 05
1	3964R (SISI)	x	x	-	-	-	-
2	ODD	x	x	-	-	x	-
3	EVEN	x	x	x	x	-	x

The SOT protocol is solely used for communicating with an SOT.

HANSH: 1...3 Handshake type

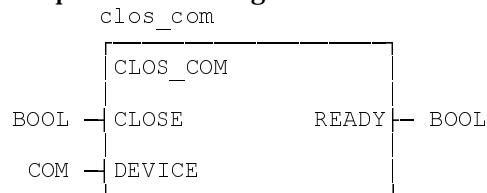
	Handshak	RECO	BTM 3/4	SIO 3	PAS	SOT	BTC 05
1	none	x	x	x	x	x	-
2	Software	x	x	-	-	-	-
3	Hardware	x	x	-	-	-	x

OPEN_COM The 'OPEN_COM' function block initializes the transfer channel to a general serial interface when there is a positive edge at the OPEN function block input. This interface can only be used for transmitting or receiving data when the block's READY output is logic '1'.



OPEN: BOOL; 0 - FB not active
 1 - open interface
DEVICE: COM; parameter of the serial
 interface
READY: BOOL; 0 - interface is not open
 or the FB is not active
 1 - interface is open

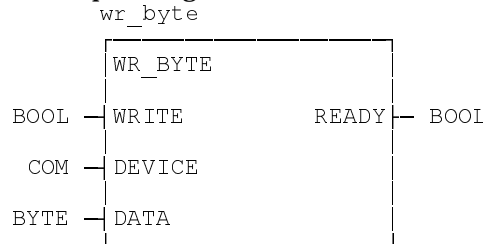
CLOS_COM The 'CLOS_COM' function block closes the transfer channel to a general serial interface, and clears the transmit and receive buffer of the interface. Data that is in the buffers after the closing process has been initiated will be lost. The READY output becomes logic '1' after the serial interface has been closed.



OPEN: BOOL; 0 - FB not active
 1 - close interface
DEVICE: COM; parameter of the serial
 interface
READY: BOOL; 0 - interface is not closed or the FB is not active

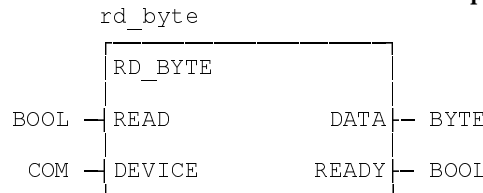
1 - interface is closed

WR_BYTE The 'WR_BYTE' function block writes a data byte into the transmit buffer of the selected interface. As long as the WRITE function block input is logic '1', the data byte that is applied to the DATA function block input is written to the buffer. The SPS firmware provides for the data output from the transmit buffer via the corresponding serial interface.



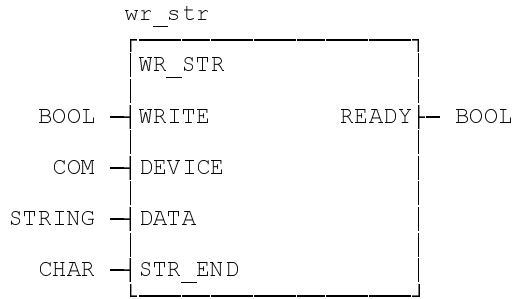
WRITE: BOOL; 0 - FB not active
 1 - write data byte into transmit
 buffer
 DEVICE: COM; Parameter of the serial
 interface
 DATA: BYTE; data byte to be transmitted
 READY: BOOL; 0 - data byte has not yet been
 written to transmit buffer or
 the FB is not active
 1 - data byte has been written
 into the transmit buffer

RD_BYTE The 'RD_BYTE' function block reads a data byte from the receive buffer of the selected interface. Upon each call of the function block and as long as the READ input is logic '1' and the receive buffer is not empty, a data byte is read from the buffer and assigned to the DATA function block. The SPS firmware provides for the transfer of the data from the corresponding serial interface to the receive buffer.



READ: BOOL; 0 - FB not active
 1 - reads data byte from the receive
 buffer
 DEVICE: COM; parameter of the serial
 interface
 DATA: BYTE; received data byte
 READY: BOOL; 0 - data byte has not yet been read
 from the receive buffer or the
 FB is not active
 1 - data byte has been read from the
 receive buffer

WR_STR The 'WR_STR' function block writes a data string into the transmit buffer of the selected interface. The character applied to the STR_END input is appended to the data string as an end-of-string character.. If the input is not connected, the character with ASCII code 0 is appended to the data string. The string is written to the transmit buffer if the free space in the buffer proves sufficient for accommodating the string. The SPS firmware provides for outputting the data from the transmit buffer via the corresponding serial interface.

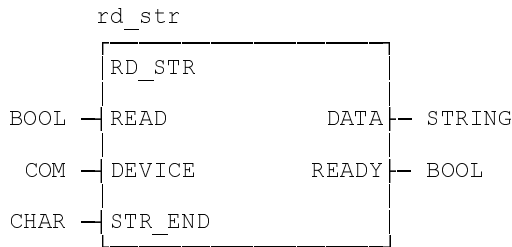


WRITE: BOOL; 0 - FB not active
 1 - write data string to transmit
 buffer
 DEVICE: COM; parameter of the serial
 interface
 DATA: BYTE; data string to be transmitted
 STR_END: CHAR; end-of-string character
 READY: BOOL; 0 - data string has not yet been
 written to the transmit buffer
 or the FB is not active
 1 - data string has been written to
 the transmit buffer

RD_STR

The '**RD_STR**' function block reads a string from the receive buffer of the selected interface. The end of the string is reached when the character that is read from the receive buffer agrees with the end-of-string character that is applied to the **STR_END** input. In this case, the **READY** output becomes logic '1'. If the **STR_END** input is not connected, the default ASCII code 0 is used as the end-of-string character.

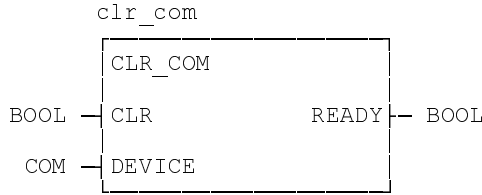
A string variable at the **DATA** output is filled with characters until the end-of-string character is read. The end-of-string character is not a part of the output string. The contents of the string variable is cleared if an end-of-string character is not received before 255 characters have been read. If there are any characters left in the receive buffer, the string variable will be filled with those characters until the end-of-string character is reached. The SPS firmware provides for the data to be transferred from the serial interface to the receive buffer.



READ: BOOL; 0 - FB not active
 1 - read data string from
 receive buffer
 DEVICE: COM; parameter of the serial
 interface
 STR_END: CHAR; end-of-string character
 DATA: BYTE; received data string
 READY: BOOL; 0 - data string has not yet been
 read completely from the
 receive buffer or the FB is
 not active

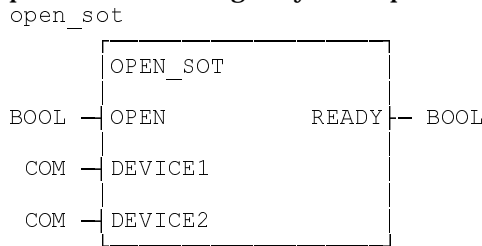
1 - data string has been read from
the receive buffer

CLR_COM The 'CLR_COM' function block is used for clearing the transmit and receive buffer of an open serial interface. Data that is in the buffers after the clearing process has been initiated will be lost.



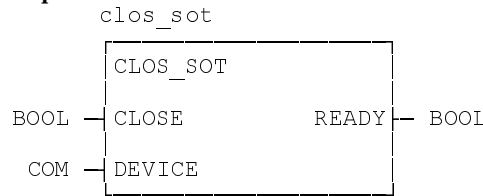
CLR: BOOL; 0 - FB not active
 1 - clear receive and transmit buffer
 DEVICE: COM; parameter of the serial
 interface
 READY: BOOL; 0 - receive or transmit buffer has
 not been cleared or the FB is
 not active
 1 - receive and transmit buffer have
 been cleared

OPEN_SOT Upon a positive edge at the OPEN function block input, the 'OPEN_SOT' function block initializes the transfer channel to an SOT interface and, if necessary, initializes the SOT (station operator terminal) expansion interface. The SOT expansion interface can be used for connecting another SOT (cascading). This block permits cascading only to be performed via the serial interface of a RECO.



OPEN: BOOL; 0 - FB not active
 1 - initialize interfaces
 DEVICE1: COM; parameter of the SOT interface
 (the value of the PROTOKOL
 element of the COM data type
 must be 2)
 DEVICE2: COM; parameter of the SOT
 expansion interface
 READY: BOOL; 0 - interfaces are not initialized
 or the FB is not active
 1 - interfaces are not initialized

CLOS_SOT Upon a positive edge at the CLOSE function block input, the 'CLOS_SOT' function block closes the transfer channel to an SOT interface and, if applicable, to the expansion interfaces of the SOT.



OPEN: BOOL; 0 - FB not active
 1 - close interface
DEVICE: COM; parameter of the serial
 interface
READY: BOOL; 0 - interface is not closed or
 the FB is not active
 1 - interface is closed

5.10.6.1.1 Communication using the MODBUS protocol 'MODBUS'

The 'MODBUS' function block permits an open serial interface to be used with the MODBUS protocol. In the master (e.g. operator terminal), the user defines a slave address that shall be used for addressing the controller. That slave address must be applied to the SL_ADDR function block input. The valid range of the slave address is between 1 and 247. The MODE input permits the communication protocol to be switched over from RTU mode to ASCII mode.

In the MODBUS protocol, the following data areas are distinguished:

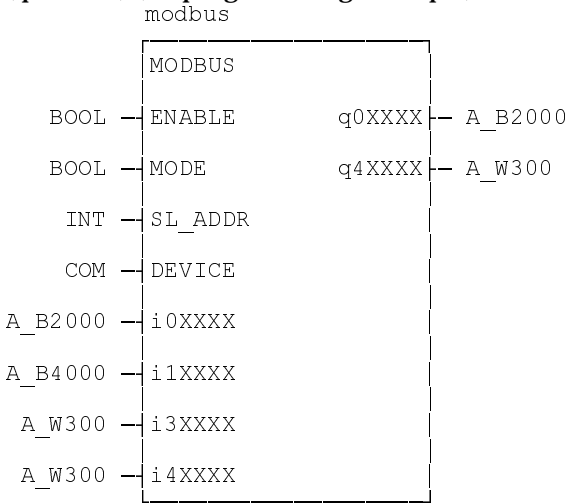
0X	outputs (BOOL)	read and write access from the master
1X	inputs (BOOL)	read access from the master
2X	input registers (WORD)	read access from the master
3X	output registers (WORD)	read and write access from the master

To enable communication, the data areas must be mapped. The user employs the input and/or output function blocks for assigning the addresses that are related in the controller:

		Access	XXXX=	Address
i0XXXX	outputs (BOOL)	read	0...1999	00000...01999 ₁
i1XXXX	inputs (BOOL)	read	0...3999	10000...13999 ₁
i3XXXX	input registers	read	0...299	30000...30299 ₁
i4XXXX	output registers	read	0...299	40000...40299 ₁
q0XXXX	outputs (BOOL)	write	0...1999	00000...01999 ₁
q1XXXX	output registers	write	0...299	40000...40299 ₁

¹ The addresses differ for different manufacturers (here: addresses of Modicon controller)

The same identifier must be used for connecting the input iXXXXX (i4XXXX) with the output qXXXXX (q4XXXX) (cf. programming example).



```

ENABLE: BOOL;      0 - FB not active
                  1 - FB active
MODE:   BOOL;      0 - communication protocol in RTU mode
                  1 - communication protocol in ASCII mode
DEVICE: COM;       parameter of the serial interface
i0XXXX: A_B2000;   Boolean outputs, read access
i1XXXX: A_B4000;   Boolean outputs, read access
i3XXXX: A_W300;    input register, read access
i4XXXX: A_W300;    output register, read access
q0XXXX: A_B2000;   Boolean outputs, write access
q4XXXX: A_W300;    output register, write access
    
```

Special data types:

```

TYPE A_B2000;
ARRAY [0..1999] OF BOOL;
END_TYPE A_B2000;
    
```

```

TYPE A_B4000;
ARRAY [0..3999] OF BOOL;
END_TYPE A_B4000;
    
```

```

TYPE A_W300;
ARRAY [0..299] OF WORD;
END_TYPE A_W300;
    
```

5.10.6.2 Error handling of the function blocks for serial interfaces

Programming or hardware faults prevent the described function blocks from being executed correctly. In this case, the error handling function informs of the cause of the malfunction.

Table 5-5: Error Handling of Function Blocks for Serial Interfaces

ErrorTyp	Function block
-106	OPEN_COM
-107	CLOS_COM
-108	OPEN_SOT

ErrorTyp	Function block
-109	CLOS_SOT
-110	WR_BYTE
-111	RD_BYTE
-203	RD_STR
-204	WR_STR
-216	MODBUS
-227	CLR_COM
ErrorNr	
238	Interface not open
240	Invalid input parameter DEVICE
241	Invalid input parameter SERNR
242	Invalid input parameter BAUD
243	Invalid input parameter DATA
244	Invalid input parameter PARITY
245	Invalid input parameter STOP
246	Invalid input parameter PROTOKOL
247	Invalid input parameter HANDSH
248	Interface cannot be found
249	All COM interfaces have already been opened
250	All SOT interfaces have already been opened
251	No cascading
252	General interface error
253	Transmit buffer is full
254	Receive buffer is full
255	Acknowledgment message frame time-out

5.11 Firmware Function Blocks

The firmware function blocks that are available for a specific controller type are described in the Interface Description of that controller.

5.12 User Function Blocks

The programming system enables the user to write function blocks that may be employed as reusable units that supplement the standard and firmware function blocks. The user function blocks may import other user functions or user function blocks and use them like they use standard or firmware functions or function blocks.

5.12.1 Import rules

Table 5-6: User Function Block Import Rules

The used function is a standard function or a firmware function	The used function is a user function	The used function block is a standard function block or a firmware function block	The used function block is a user function block
<ul style="list-style-type: none"> Direct utilization without import or declaration according to the selected entry language. 	<ul style="list-style-type: none"> Import of the function Utilization according to the selected entry language. 	<ul style="list-style-type: none"> Import not necessary since it is contained in the standard library. A function block needs space for the data that belongs to its assignments (count, 	<ul style="list-style-type: none"> Import of the function block Declaration of the FB assignments Utilization according to the selected entry language.

		time, etc.). The unit that wants to use it must reserve the required space. Declaration of the FB assignments in a separate declaration editor. <ul style="list-style-type: none"> Utilization according to the selected entry language. 	
--	--	--	--

A user function block may utilize standard functions, firmware functions, user functions, user function blocks, standard function blocks, and firmware function blocks.

Nesting can be continued to any depth.

NOT ALLOWED is that function block 'A' employs itself (recursive utilization), or that function block 'A' employs function block 'B' that employs function block 'A', etc.

5.13 Error Handling

When a program executes in the SPS, variable values may be computed for which the subsequent functions and function blocks have not been defined.

Example:

The divisor of an integer division is computed as zero. This is a value that is usually not permitted.

5.13.1 Basics of the Error Handling Concept

Goal

The user shall have the opportunity of locating the above-mentioned errors and of responding to them.

For this purpose, three error variables have automatically been introduced in each function and in each function block, in addition to the known function contents. Uninterrupted execution of the user program is ensured in any case. The program executes as before if the user ignores the error message.

Table 5-7: SPS Error Variables

ErrorFlg (BOOL), Standard FALSE	ErrorNr (USINT), Standard: 0	ErrorTyp (INT), Standard: 0
<ul style="list-style-type: none"> ErrorFlg= 0, no error has occurred during the execution up to now (standard) ErrorFlg= 1, at least one error has occurred that will be specified in detail. 	<ul style="list-style-type: none"> A number > 0 provides detailed information about the error. 	<ul style="list-style-type: none"> Negative error types are reserved for the SPS manufacturer to be used for standard and firmware functions and function blocks. Type numbers greater than zero are available to

		be used by the user.
--	--	----------------------

Occurred errors will be transported upwards to the calling unit, up to the program itself. Without any user activities, the three variables will be declared in the functions and function blocks that are written by the user and are therefore available to be accessed by the user.

In the next cycle, there is again a movement to the right with reset error variables. The following table results when the causes of the error are eliminated. If the error continues to exist, the table 'error in fk FK_B without user reaction' reappears.

Error in fk FK_B no longer exists, no user reaction:

	ErrorFlg	ErrorNr	ErrorTyp
PR TEST (section 1)	1	23	-1 error remains stored
FB FB_X/x1 (section 1)	0	0	0 to right w/o error
FK FK_A (section 1)	0	0	0 to right w/o error
fk FK_B	0	0	0 no more errors now
FK FK_A (section 2)	0	0	0 back w/o errors
FB FB_X/x1 (section 2)	0	0	0 back w/o errors
fk FK_B	0	0	0 to right w/o error
FB FB_X/x1 (section 3)	0	0	0 back w/o errors
PR TEST (section 2)	1	23	-1 error remains stored
FK FK_A (section 1)	0	0	0 to right w/o error
fk FK_B	0	0	0 to right w/o error
FK FK_A (section 2)	0	0	0 back w/o errors
PR TEST (section 3)	1	23	-1 error exists

The user must actively clear the error on program level. The user has the opportunity on an intervention, at the earliest in the 'FK FK_A' user function (section 2) or later in another program organization unit. First, the user recognizes that the 'ErrorFlg' error bit has been set; and can subsequently evaluate 'ErrorNr'/'ErrorTyp'. After evaluation, setting 'ErrorFlg' to '0' proves sufficient since 'ErrorNr'/'ErrorTyp' will thus be invalid. (ErrorNr, ErrorTyp need not be modified).

Immediate interpretation of the error in FK FK_A, (section 2a) and resetting ErrorFlg:

	ErrorFlg	ErrorNr	ErrorTyp
PR TEST (section 1)	0	0	0
FB FB_X/x1 (Section 1)	0	0	0
FK FK_A (Section 1)	0	0	0
fk FK_B	1	23	-1 error location
FK FK_A (Section 2a)	1	23	-1 error recogn. and eliminated
FK FK_A (Section 2b)	0	23	-1 (ErrorFlg modified)
FB FB_X/x1 (Section 2)	0	0	0 error eliminated, no copy
fk FK_B	0	0	0
FB FB_X/x1 (Section 3)	0	0	0
PR TEST (Section 2)	0	0	0
FK FK_A (Section 1)	0	0	0
fk FK_B	0	0	0
FK FK_A (Section 2)	0	0	0
PR TEST (Section 3)	0	0	0

5.13.3 Error Handling of Multiple Errors

Theoretically, a second error with different or even the same assignments of 'ErrorNr' and 'ErrorTyp' can occur in a different program section before the first error has been recognized and cleared. This error will also be passed on to the left in the import tree. 'ErrorNr' and 'ErrorTyp' will not be overwritten as long as 'ErrorFlg' has not been reset. Consequently, the information about the second error goes to a 'waiting state'.

5.13.4 Error Handling in User Files

The three error variables are automatically declared when a new user file is created, and the user knows the error mechanism of standard and firmware files. Thus, the user may employ that mechanism for user-related files. To avoid confusion, the user should only utilize error types with positive numbers.

5.13.5 Overview of Possible Errors and Their Causes

When 'ErrorFlg' = 'TRUE', an error has occurred during the execution of an operation, a function, or a function block.

The 'ErrorTyp' error type names the cause:
 -1 ... -255 functions or function blocks
 from -10000 operations

The error proper is characterized by the 'ErrorNr' error number.

5.13.6 Errors in functions and function blocks

Explanation:

Nr 'ErrorTyp'; specifies the fb/fk that has caused the error.

fk Firmware function, description in /4/

*fk Standard function, description in Section 7

fb Firmware function block, description in /4/

*fb Standard function block, description in Section 8

Table 5-8: Error Causes in Function Blocks, SPS

No.	POE	Name	Comment
-1	fk	M_FKT	Interrogating M aux. function and specifying the aux. function number
-2	fk	M_FKT_Q	Acknowledging M aux. Function and specifying the aux. function number
-3	fk	S_FKT	Interrogating S aux. function and specifying the aux. function number
-4	fk	S_FKT_Q	Acknowledging S aux. function and specifying the aux. function number
-5	fk	T_FKT	Interrogating T aux. function and specifying the aux. function number
-6	fk	T_FKT_Q	Acknowledging T aux. function and specifying the aux. function number
-7	fk	Q_FKT	Interrogating Q aux. function and specifying the aux. function number
-8	fk	Q_FKT_Q	Acknowledging Q aux. Function and specifying the aux. function number
-9	fk	EVENT	Interrogating events
-10	fk	EV_ST	Value transfer to events
-11	fk	EV_SET	Conditional setting of events
-12	fk	EV_RES	Conditional resetting of events
-13	fk	MSG_WR	Diagnosis output; message number directly specified
-14	fk	MSG_RD	Reading the CNC message number
-15	fk	MRF	Interrogating magazine homing
-16	fk	MRF_Q	Acknowledging magazine homing
-17	fk	MMV	Interrogating magazine in new position
-18	fk	MMV_Q	Acknowledging magazine in new position
-19	fk	TCH	Interrogating general tool change
-20	fk	TCH_Q	Acknowledging general tool change
-21	fk	TMS	Interrogating tool change from magazine to spindle
-22	fk	TMS_Q	Acknowledging tool change from magazine to spindle
-23	fk	TSM	Interrogating tool change from spindle to magazine
-24	fk	TSM_Q	Acknowledging tool change from spindle to magazine
-25	fk	XMS	Initializing tool transfer from magazine to spindle
-26	fk	XMS_PA	Tool transfer from magazine to spindle permitted

No.	POE	Name	Comment
-27	fk	XMS_NA	Tool transfer from magazine to spindle not permitted
-28	fk	XMS_Q	Acknowledging tool transfer from magazine to spindle
-29	fk	XSM	Initializing tool transfer from spindle to magazine
-30	fk	XSM_PA	Tool transfer from spindle to magazine permitted
-31	fk	XSM_NA	Tool transfer from spindle to magazine not permitted
-32	fk	XSM_Q	Acknowledging tool transfer from spindle to magazine
-33	fk	XMG	Initializing tool transfer from magazine to gripper
-34	fk	XMG_PA	Tool transfer from magazine to gripper permitted
-35	fk	XMG_NA	Tool transfer from magazine to gripper not permitted
-36	fk	XMG_Q	Acknowledging tool transfer from magazine to gripper
-37	fk	XSG	Initializing tool transfer from spindle to gripper
-38	fk	XSG_PA	Tool transfer from spindle to gripper permitted
-39	fk	XSG_NA	Tool transfer from spindle to gripper not permitted
-40	fk	XSG_Q	Acknowledging tool transfer from spindle to gripper
-41	fk	XGS	Initializing tool transfer from gripper to spindle
-42	fk	XGS_PA	Tool transfer from gripper to spindle permitted
-43	fk	XGS_NA	Tool transfer from gripper to spindle not permitted
-44	fk	XGS_Q	Acknowledging tool transfer from gripper to spindle
-45	fk	XGM	Initializing tool transfer from gripper to magazine
-46	fk	XGM_PA	Tool transfer from gripper to magazine permitted
-47	fk	XGM_NA	Tool transfer from gripper to magazine not permitted
-48	fk	XGM_Q	Acknowledging tool transfer from gripper to magazine
-49	*fk	GRAY_BYT	Type conversion Gray code → hexadecimal
-50	*fk	BYT_GRAY	Type conversion hexadecimal → Gray code
-51	*fk	SBCD_INT	Type conversion BCD code, byte, 2 digits → INTEGER
-52	*fk	BCD_INT	Type conversion BCD code, word, 4 digits → INTEGER
-53	*fk	BYTE_INT	Type conversion BYTE → INTEGER
-54	*fk	WORD_INT	Type conversion WORD → INTEGER
-55	*fk	INT_BYTE	Type conversion INTEGER number → BYTE
-56	*fk	INT_WORD	Type conversion INTEGER number → word
-57	*fk	INT_BCD	Type conversion INTEGER number → 4-digit BCD word
-58	*fk	USI_INT	Type conversion UNSIGNED SHORT INTEGER → INTEGER
-59	*fk	INT_USI	Type conversion INTEGER → UNSIGNED SHORT INTEGER
-60	*fk	USI_BYTE	Type conversion UNSIGNED SHORT INTEGER → BYTE
-61	*fk	BYTE_USI	Type conversion BYTE → UNSIGNED SHORT INTEGER
-62	*fk	CONCAT_B	Appending Low-Byte to High-Byte
-63	*fk	CONCAT_W	Appending Low-Word to High-Word
-64	*fk	H_BYTE	Removing High-Byte from WORD
-65	*fk	L_BYTE	Removing Low-Byte from WORD
-66	*fk	H_WORD	Removing High-Word from DWORD
-67	*fk	L_WORD	Removing Low-Word from DWORD
-68	*fk	SIGN_INT	Sign of an INT number
-69	*fk	ABS_INT	Absolute value of an INT number
-70	*fk	SHL_BYTE	Shift BYTE by n digits to the left
-71	*fk	SHL_WORD	Shift WORD by n digits to the left
-72	*fk	SHR_BYTE	Shift BYTE by n digits to the right
-73	*fk	SHR_WORD	Shift WORD by n digits to the right
-74	*fk	ROL_BYTE	Rotate BYTE by n digits to the left

No.	POE	Name	Comment
-75	*fk	ROL_WORD	Rotate WORD by n digits to the left
-76	*fk	ROR_BYTE	Rotate BYTE by n digits to the right
-77	*fk	ROR_WORD	Rotate WORD by n digits to the right
-78	*fb	SR	FLIP_FLOP, dominating setting
-79	*fb	RS	FLIP_FLOP, dominating resetting
-80	*fb	R_TRIG	Edge recognition of a rising edge
-81	*fb	F_TRIG	Edge recognition of a falling edge
-82	*fb	CTUD_USI	Up/down counter, range UNSIGNED SHORT INTEGER
-83	*fb	CTUD_UIN	Up/down counter, range UNSIGNED INTEGER
-84	*fb	CTUD_INT	Up/down counter, range INTEGER
-85	*fb	TP	Timer impulse
-86	*fb	TON	ON delay timer
-87	*fb	TOFF	OFF delay timer
-88	fb	SC_WRITE	<i>Function is no longer supported</i>
-89	fb	SC_READ	<i>Function is no longer supported</i>
-90	fb	VAR_WR	Writing to a CNC variable of the INTEGER data type
-91	fb	VAR_RD	Reading the value of the INTEGER data type variable
-92	fb	SEL_MEM	Selecting the NC program memory
-93	fb	ACT_MEM	Interrogating the active NC program memory
-94	fk	XMS_CA	Aborting tool transfer from magazine to spindle
-95	fk	XSM_CA	Aborting tool transfer from spindle to magazine
-96	fk	XMG_CA	Aborting tool transfer from magazine to gripper
-97	fk	XSG_CA	Aborting tool transfer from spindle to gripper
-98	fk	XGS_CA	Aborting tool transfer from gripper to spindle
-99	fk	XGM_CA	Aborting tool transfer from gripper to magazine
-100	fk	MHP	<i>Function is no longer supported</i>
-101	fk	MHP_Q	<i>Function is no longer supported</i>
-102	fk	GRP	<i>Function is no longer supported</i>
-103	fk	GRP_Q	<i>Function is no longer supported</i>
-104	fk	REL	<i>Function is no longer supported</i>
-105	fk	REL_Q	<i>Function is no longer supported</i>
-106	fb	OPEN_COM	Initializing a general data channel
-107	fb	CLOS_COM	Terminating data transfer of a general data channel
-108	fb	OPEN_SOT	Initializing a transfer channel with SOT
-109	fb	CLOS_SOT	Terminating data transfer of a transfer channel with SOT
-110	fb	WR_BYTE	Writing a BYTE into the transmit buffer
-111	fb	RD_BYTE	Reading a BYTE on the general transfer channel
-112	fb	CTRL_COM	Interrogating the status of a serial interface
-113	fk	MAG_ACT	Interrogating the magazine axis preselection with comb. spindle/turret axis
-114	fk	MAG_Q	Acknowledging magazine axis preselection with comb. spindle/turret axis
-115	fk	SPDL_ACT	Interrogating the spindle preselection with comb. spindle/turret axis
-116	fk	SPDL_Q	Acknowledging the spindle preselection with comb. spindle/turret axis
-117	fk	M_ALL	Interrogating M aux. functions w/o specifying the aux. function number
-118	fk	M_ALL_Q	Acknowledging M aux. functions w/o specifying the aux.

No.	POE	Name	Comment
			function number
-119	fk	S_ALL	Interrogating S aux. functions w/o specifying the aux. function number
-120	fk	S_ALL_Q	Acknowledging S aux. functions w/o specifying the aux. function number
-121	fk	T_ALL	Interrogating T aux. functions w/o specifying the aux. function number
-122	fk	T_ALL_Q	Acknowledging T aux. functions w/o specifying the aux. function number
-123	fk	Q_ALL	Interrogating Q aux. functions w/o specifying the aux. function number
-124	fk	Q_ALL_Q	Acknowledging Q aux. functions w/o specifying the aux. function number
-125	fb	USERBOF	<i>Function is no longer supported</i>
-126	fk	M_NR	Reading the M auxiliary function number
-127	fk	S_NR	Reading the S auxiliary function number
-128	fk	Q_NR	Reading the Q auxiliary function number
-129	fk	XFER_CHK	De-activating tool transfer verification
-130	fk	MSG_WR_N	Message output with auxiliary information as a number
-131	fk	MSG_WR_A	Message output with auxiliary information as an axis designation
-132	fb	AXD_WR	Writing required data
-133	fb	AXD_RD	Reading required data
-134	fk	SPMOD	Interrogating spindle mode presel. for main spindle with rotary axis capab.
-135	fk	SPMOD_Q	Acknowledg. spindle mode presel. for main spindle with rotary axis capab.
-136	fk	ROTMOD	Interrog. rotary axis mode presel. for main spindle with rotary axis capab.
-137	fk	ROTMOD_Q	Ack. rotary axis mode presel. for main spindle with rotary axis capab.
-138	*fk	CHAR_BYT	Type conversion CHAR → BYTE
-139	*fk	BYT_CHAR	Type conversion BYTE → CHAR
-140	*fk	INT_STR	Type conversion INTEGER → STRING
-141	*fk	STR_INT	Type conversion STRING → INTEGER
-142	*fk	LEN	Length of a STRINGS
-143	*fk	LEFT	Left L_ characters of a STRINGS
-144	*fk	RIGHT	Right L_ characters of a STRINGS
-145	*fk	MID	From P_ th character L_ characters of a STRINGS
-146	*fk	CONCAT_S	Concatenating two strings
-147	*fk	INSERT	Inserting a STRINGS after the L-th character
-148	*fk	DELETE	Delete L_ characters of a STRING from the P-th character onward
-149	*fk	REPLACE	Replace L_ characters of a STRING from the P-th character by IN2_
-150	*fk	FIND	Locate character string IN2_ in IN1_
-151	fb	GUI_SK	Transfer of soft keys to the graphical user interface
-152	*fk	DINT_DW	Type conversion DOUBLE INTEGER → DOUBLE WORD
-153	*fk	DW_DINT	Type conversion DOUBLE WORD → DOUBLE INTEGER

No.	POE	Name	Comment
-154	*fk	DINT_INT	Type conversion DOUBLE INTEGER → INTEGER
-155	*fk	INT_DINT	Type conversion INTEGER → DOUBLE INTEGER
-156	*fk	DINT_TIM	Type conversion DOUBLE INTEGER → time
-157	*fk	TIM_DINT	Type conversion time → DOUBLE INTEGER
-158	fk	HNDWHEEL	Transfer of handwheel position
-159	*fk	SHL_DW	Shift DOUBLE WORD by n characters to the left
-160	*fk	SHR_DW	Shift DOUBLE WORD by n characters to the right
-161	*fk	ROL_DW	Rotate DOUBLE WORD by n characters to the left
-162	*fk	ROR_DW	Rotate DOUBLE WORD by n characters to the right
-163	fb	RLVAR_WR	Writing to a CNC variable of the REAL data type
-164	fb	RLVAR_RD	Reading the value of the variable of the REAL data type
-165	*fk	DINT_RL	Type conversion DOUBLE INTEGER → REAL
-166	*fk	RL_DINT	Type conversion REAL → DOUBLE INTEGER
-167	*fk	STR_RL	Type conversion STRING → REAL
-168	*fk	RL_STR	Type conversion REAL → STRING
-169	fb	TLD_WR	Writing tool data
-170	fb	TLD_RD	Reading tool data
-171	*fk	DINT_UDI	Type conversion DOUBLE INTEGER → UNSIGNED DOUBLE INTEGER
-172	*fk	UDI_DINT	Type conversion UNSIGNED DOUBLE INTEGER → DOUBLE INTEGER
-173	fb	DATE_RD	Reading the date
-174	fb	TOD_RD	Reading the time
-175	fb	OTD_WR	Writing zero point data
-176	fb	OTD_RD	Reading zero point data
-177	fb	MTD_WR	Writing zero point data
-178	fb	MTD_RD	Reading zero point data
-179	fb	NETIO_RD	Reading real-time data bits
-180	fk	T_NR	Reading the T auxiliary function number for process (PROC)
-181	fk	E_FKT	Reading the E auxiliary functions (EDGE) for process (PROC)
-182	fk	E_FKT_Q	Acknowledging E auxiliary functions (EDGE) for process (PROC)
-183	fk	E_ALL	Interrogating any E auxiliary function for process (PROC)
-184	fk	E_ALL_Q	Acknowledging any E auxiliary function for process (PROC)
-185	fk	E_NR	Reading the E auxiliary function number for process (PROC)
-186	fb	TLBD_WR	Writing base tool data
-187	fb	TLED_WR	Writing tool edge data
-188	fb	TL_ENABLE	Enabling tool data
-189	fb	TLBD_RD	Reading base tool data
-190	fb	TLED_RD	Reading tool edge data
-191	fb	TL_RESET	Resetting a tool
-192	fb	TL_DELETE	Deleting a tool
-193	fb	TL_MOVE	Moving a tool
-195	*fb	BOOL_BYTE	Conversion 8-bit → Byte
-196	*fb	BYTE_BOOL	Conversion Byte → 8-bit
-197	*fb	BOOL_WORD	Conversion 16-bit → Word
-198	*fb	WORD_BOOL	Conversion Word → 16-bit
-199	*fb	BOOL_DW	Conversion 32-bit → Double word

No.	POE	Name	Comment
-200	*fb	DW_BOOL	Conversion Double word → 32-bit
-201	fb	FLASH	Clock generator
-202	fb	TOGGLE	Toggling a bit
-203	fb	RD_STR	Reading a string via the serial interface
-204	fb	WR_STR	Writing a string into the transmit buffer
-205	fk	TIM_DAY	Conversion of TIME_ into numeric value day
-206	fk	TIM_HOUR	Conversion of TIME_ into numeric value hour
-207	fk	TIM_MIN	Conversion of TIME_ into numeric value minute
-208	fk	TIM_SEC	Conversion of TIME_ into numeric value second
-209	fk	TIM_MS	Conversion of TIME_ into numeric value millisecond
-210	fk	MAKETIME	Conversion FK inputs day 'D', hour 'H', minute 'M', second 'S' and millisecond 'MS' into a time value
-211	fb	CLTVA_RD	Reading CLT variables
-212	fb	CLTVA_WR	Writing CLT variables
-213	fb	IB_GRDEF	Group definition Interbus S
-214	fb	IB_GRON	Activating group - Interbus S
-215	fb	IB_GROFF	De-activating group - Interbus S
-216	fb	MODBUS	Communication block for MODBUS communication
-217	fb	IB_SAERR	Interbus service No. 5C - Send All Module Error Request
-218	fb	IB_QAERR	Interbus service No. 65 - Quit Module Error All-Service
-219	fb	IB_ALSTP	Interbus service No. 4A - Alarm-Stop Bus Reset
-220	fb	IB_CLRD	Interbus service No. 4E - Clear Display
-221	fb	DCD_RD	Reading D corrections
-222	fb	DCD_WR	Writing D corrections
-223	fb	NCVAR_RD	Reading NC variables
-224	fb	NCVAR_WR	Writing NC variables
-225	fb	GUI_SK16	Enabling (16) machine function keys MUI/Menu9
-226	fk	REV_SYNC	Synchronous swiveling of the turret in the NC block
-227	fb	CLR_COM	Clearing receive and transmit buffer of a serial interface
-228	fk	SINT_INT	Conversion SINT number into INT number
-229	fk	INT_SINT	Conversion INT number into SINT number
-230	fk	SINT_BYT	Conversion SINT number into BYTE
-231	fk	BYT_SINT	Conversion BYTE into SINT number

5.13.7 Errors in operations and IL instructions

Explanation:

No: 'ErrorTyp', specifies the operation (for which data type) that has caused the error.

Table 5-9: Error cause in Operations and IL, SPS

No.	Comment
	Error in addition: ADD, ADD
-10000	USINT
-10001	UINT
-10002	UDINT
-10003	ULINT

No.	Comment
-10004	SINT
-10005	INT
-10006	DINT
-10007	LINT
-10008	REAL
-10009	LREAL
	Error in subtraction SUB/SUB(
-10010	USINT
-10011	UINT
-10012	UDINT
-10013	ULINT
-10014	SINT
-10015	INT
-10016	DINT
-10017	LINT
-10018	REAL
-10019	LREAL
	Error in multiplication MUL/MUL(
-10020	USINT
-10021	UINT
-10022	UDINT
-10023	ULINT
-10024	SINT
-10025	INT
-10026	DINT
-10027	LINT
-10028	REAL
-10029	LREAL
	Error in division DIV/DIV(
-10030	USINT
-10031	UINT
-10032	UDINT
-10033	ULINT
-10034	SINT
-10035	INT
-10036	DINT
-10037	LINT
-10038	REAL
-10039	LREAL
	Error in modulo division MOD/MOD(
-10040	USINT
-10041	UINT
-10042	UDINT
-10043	ULINT
-10044	SINT
-10045	INT
-10046	DINT
-10047	LINT

No.	Comment
	Indexing beyond type boundaries
-10050	USINT/SINT - Index
-10051	UINT/INT - Index
-10052	UDINT/DINT - Index
-10053	ULINT/LINT - Index
	Comparison cannot be performed, type REAL
-10060	GT, Greater than
-10061	GE, Greater than or equal to
-10062	EQ, Equal to
-10063	LE, Less than or equal to
-10064	LT, Less than
-10065	NE, Not equal to

Table 5-10: Error contents in Operations and IL, SPS

No.	Comment
0	No error
1	Invalid input parameter; operation, function, or function block is not executed. Undefined results may be returned.
2	Overlimit. The maximum value is returned.
3	Underlimit. The minimum value is returned.
4	Conversion error; the input parameter could not correctly be converted. Conversion is performed with an internally modified input parameter. Undefined results may be returned.
5	Division by zero.
7	Index error; overlimit; command is not executed.
8	Operation is not defined.
236	Process data channel overflow. More than 8 TLD, OTD, MTD, NC_VAR, TLED, TLBD, DCD have simultaneously been programmed.
237	To many variable accesses. More than 100 NC variables have been programmed.
238	Interface not open. WR_STRING or RD_STRING access a serial interface that has not yet been opened.
239	STRING processing overflow. A STRING of more than 255 characters has occurred during STRING function processing.
240	Invalid input parameter DEVICE; the DEVICE number specified in the selection of the serial interface parameters is negative or too high.
241	Invalid input parameter SERNR; the SERNR number specified in the selection of the serial interface parameters is negative or too high.
242	Invalid input parameter BAUD; the baud rate specified in the selection of the serial interface parameters is negative or too high.
243	Invalid input parameter DATA; the DATA number of data bits specified in the selection of the serial interface parameters is negative or too high.
244	Invalid input parameter PARITY; the parity bit evaluation number specified in the selection of the serial interface parameters is negative or too high.
245	Invalid input parameter STOP; the STOP number of stop bits specified in the selection of the serial interface parameters is negative or too high.

No.	Comment
246	Invalid input parameter PROTOKOL; the PROTOKOL number for the serial interface type specified in the selection of the serial interface parameters is negative or too high.
247	Invalid input parameter HANDSH; the HANDSH number for the handshake type specified in the selection of the serial interface parameters is negative or too high.
248	Interface cannot be found; the interface that has been specified in the selection of the serial interface parameters cannot be found.
249	All COM interfaces open; all serial interfaces have been opened.
250	All SOT interfaces open; all serial interfaces have been opened as SOT interfaces.
251	Cascading SOTs is not possible.
252	General interface error; parity, frame, overrun
253	Transmit buffer overflow; more than ... characters have been written to the transmit buffer.
254	Receive buffer overflow; more than ... characters have been stored in the receive buffer.
255	Acknowledgment message frame time-out.

5.13.8 Errors in exceptional REAL operations

Explanation:

NaN Not a Number: Result of an unexecutable operation.

∞ Infinity: Result of an overlimit condition

Number any number except zero, ∞ and NaN

The second line contains 'ErrorTyp' / 'ErrorNr' or 'no error'.

Table 5-11: Error in REAL ADD Operations

ADD	Number	Zero	∞	$-\infty$	NaN
Number +	Number	Number	∞	$-\infty$	NaN
	No error	No error	-10008/2	-10008/3	-10008/8
Zero +	Number	Zero	∞	$-\infty$	NaN
	No error	No error	-10008/2	-10008/3	-10008/8
∞ +	∞	∞	∞	NaN	NaN
	-10008/2	-10008/2	-10008/2	-10008/8	-10008/8
NaN +	NaN	NaN	NaN	NaN	NaN
	-10008/8	-10008/8	-10008/8	-10008/8	-10008/8

Table 5-12: Error in REAL SUB Operations

SUB	Number	Zero	∞	$-\infty$	NaN
Number -	Number	Number	∞	$-\infty$	NaN
	No error	No error	-10018/3	-10018/2	-10018/8
Zero -	Number	Zero	∞	$-\infty$	NaN
	No error	No error	-10018/3	-10018/2	-10018/8
$\infty -$	∞	∞	∞	NaN	NaN
	-10018/2	-10018/2	-10018/8	-10018/2	-10018/8
NaN -	NaN	NaN	NaN	NaN	NaN
	-10018/8	-10018/8	-10018/8	-10018/8	-10018/8

Table 5-13: Error in REAL MUL Operations

MUL	Number	Zero	∞	$-\infty$	NaN
Number *	Number	Number	∞	$-\infty$	NaN
	No error	No error	-10028/2	-10028/3	-10028/8
Zero *	Number	Zero	∞	$-\infty$	NaN
	No error	No error	-10028/8	-10008/8	-10028/8
$\infty *$	∞	∞	∞	NaN	NaN
	-10028/2	-10028/2	-10028/2	-10028/3	-10028/8
NaN *	NaN	NaN	NaN	NaN	NaN
	-10028/8	-10028/8	-10028/8	-10028/8	-10028/8

Table 5-14: Error in REAL DIV Operations

DIV	Number	Zero	∞	$-\infty$	NaN
Number /	Number	Number	Zero	- Zero	NaN
	No error	No error	No error	No error	-10038/8
Zero /	Zero	NaN	Zero	- Zero	NaN
	No error	-10038/8	No error	No error	-10038/8
$\infty /$	∞	∞	∞	NaN	NaN
	-10038/2	-10038/2	-10038/8	-10038/8	-10038/8
NaN /	NaN	NaN	NaN	NaN	NaN
	-10038/8	-10038/8	-10038/8	-10038/8	-10038/8

6. GATEWAY SIGNALS

6.1 Introduction to CNC \leftrightarrow SPS Gateway Signals

The MT-CNC contains 7 “CNCs in the form of processes and one programmable controller (SPS). Since the SPS in the interface to machine automation, the SPS has to exchange information with the processes of the MT-CNC. This information is exchanged via the Gateway signals.

Because Gateway input and output signals are not wired in hardware they are referred to as Software I/Os. Each signal is specified by its address and function and the programmer of the SPS program must use them accordingly.

The Gateway signals are divided into five groups:

- **Process** These status and control signals are available once for each of the 7 available MT-CNC processes (0...6).
- **Axis** These status and control signals are available once for each of the 20 available MT-CNC axes.
- **General** These signals are available once and influence the complete MT-CNC system.
- **Indirectly Addressable** Parts of the common memory area between CNC and SPS can be accessed only by using provided FUNCTIONS and FUNCTION_BLOCKS in the SPS. This assures proper timing during data access and eliminates complicated data access programs.
- **Service Channels to Digital Drives** Service channels (1 for SERCOS and 1 for APR) are used to exchange data (non real-time) between digital drive systems and the SPS of the MT-CNC, using the Serial Real-time Communication System (SERCOS) interface.

6.2 Declaring Gateways in SPS

The SPS allows simple one-line declaration of all input or output Gateway signals for a process, axis and external mechanism by declaring it of the type STRUCTURE (derived data type) in the declaration list of the SPS.

The direction between the SPS and the CNC can be specified in the identifier with “C” for control (SPS \Rightarrow CNC) and “S” for status (CNC \Rightarrow SPS). The remaining letters of the identifier describe the signal (short mnemonic)

The identifiers used for the Gateway signals in the SPS Declaration List consist of a maximum of 9 user characters followed by up to 9 fixed characters which describe the individual signal.

NOTE: *The user part of the identifier is defined as PxxC or PxxS for process signals, and AxxC or AxxS for axis signals in this guide. This may be different in your SPS program if something else is specified in your declaration list.*

13VRS and below

The Gateway signals may also be defined by using the PBL.EXE file on MUI install disk #1. Copy this file to :\

6.3 Process Signals

Each of the 7 available processes and 25 available external mechanisms has Gateway signals between the CNC and SPS. Each process has its own signals and is therefore independent of other processes.

Table 6-1: Process signal definition

P03C.EXT24	
P	P=process / M=mechanism
03	process / mechanism number (0...31)
C	C=control signal / S=status signal
“.”	Separator between user defined and system defined identifier (derived data type)
EXT24	Signal name or mnemonic

PxxC.EXT24 External 24V Power Present

“0”= External 24V power missing

“1”= External 24V power is present

This signal is used to determine whether the 24VDC power for the station inputs is present or not. The PxxC.EXT24 must have a status of “1” for the Power Logic in the CNC. This signal must have a status of ‘1’ for the power logic in the CNC.

PxxC.LINE Control Voltage Present

“0”= Control voltage missing

“1”= Control voltage is present

The signal is used to check whether single phase power is applied to the drive’s power supply. Some power supplies (e.g., TVD) do not require any control voltage. The status of PxxC.EXT24 should be assigned to this signal.

PxxC.ESAT Station E-Stop PB Not Active

“0”= Station E-Stop PB is active

“1”= Station E-Stop PB is not active

The signal is used to monitor the E-Stop pushbutton of a station. Refer to the System Commissioning Manual for further hardware information.

PxxC.EMACH Machine E-Stop Not Active

“0”= Machine E-Stop device active

“1”= Machine E-Stop device not active

The signal is used to monitor a E-Stop condition of the complete machine.

PxxC.ESTP(y) E-Stop condition (y=1...3) active

“0”= E-Stop condition (1...3) active

“1”= E-Stop condition (1...3) not active

Each process has 3 additional signals to separate E-Stop conditions for better diagnostics. The signals must be wired in the correct sequence to provide proper diagnostics.

NOTE: *These E-Stop conditions should only cause an E-Stop for the single station (process). Incorrect diagnostics may be displayed if the E-Stop condition causes E-Stop of the complete machine. These signals must have the status of “1” to enable the power to the process.*

PxxC.BBSUP Drive Power Supply Ready (Bb1 signal)

“0”= Drive power supply not ready (Bb contact open)

“1”= Drive power supply is ready (Bb contact closed)

The status of the drive power supply (Bb) ready to operate signal must have a status of “1” to enable power for the process. The Bb1 signal of the power supply should be wired to a SPS input, which is then assigned to the PxxC.BBSUP signal. If a Bb1 signal does not exist, set the status of this signal to ‘1’.

PxxC.TPSUP Overtemperature Drive Power Supply

“0”= Overtemperature of drive power supply

“1”= Temperature of drive power supply OK

The status of this signal must be “1” to enable process power. The status of “0” will result in the immediate power loss of the process.

PxxC.PWRDY Drive Power Contactor Status

“0”= Drive power contactor is OFF

“1”= Drive power contactor is ON

This signal is used to monitor the auxiliary contact of the drive power contactor. The signal PxxS.POWON is used to switch the three phase power for the process. An auxiliary contact of the power contactor should be wired to a SPS input. Its status should then be assigned to the signal PxxC.PWRDY.

PxxC.UDRDY Drive DC Bus Voltage Ready

“0”= Drive DC Bus voltage not OK

“1”= Drive DC bus voltage present and OK

With the status of this signal the Power Logic monitors the DC bus voltage of the drive package. This signal must switch to “1” after the power contactor is switched on. Most Indramat power supplies provide a “UD or Bb2” contact (DC bus voltage OK) that can be wired to a SPS input, which then must be assigned to the signal PxxC.UDRDY.

NOTE: *The drive power supply may NOT provide a “UD or Bb2” contact to monitor the DC bus voltage, but requires a soft start circuit. In this case combine (binary AND) the AxxC.READY signals of all the process axes and assign the RLO to the signal PxxC.UDRDY.*

PxxC.BBMSP Main Spindle Drive Ready

“0”= Main spindle drive not ready to operate

“1”= Main spindle drive ready to operate

This signal is used to monitor the “Drive Ready” (Bb) signal of the process main spindle drive. The signal PxxC.BBMSP status “1” must be simulated in the SPS to enable process power if no main spindle is used for the process.

6.4 Power On Control

Each process has its own Power On control interface and power control signals. It is possible to switch the power ON or OFF of each signal station (process) independently of other processes. Refer to the System Commissioning Instruction Manual (IAE74760) for information about hardware wiring and the proper drawings.

PxxS.POWEN CNC Power Enable

“0”= No enable power from the CNC Power Logic

“1”= Power enabled by the CNC Power Logic

The power Logic enables the SPS to request Power On for the station (process). The signal has a status of “1” if all necessary conditions for the process Power On are fulfilled.

Table 6-2: Requirements for PxxS.POWEN=“1”

SIGNAL	STATUS	DESCRIPTION
PxxS.ERROR	0	No process error active
PxxC.EXT24	1	External 24VDC is present
PxxC.LINE	1	Single phase for drives is present
PxxC.ESTAT	1	Station E-Stop is not active
PxxC.EMACH	1	Machine E-Stop is not active
PxxC.ESTP1 PxxC.ESTP2 PxxC.ESTP3	1	Further E-Stop conditions are not active
AxxC.OTRVL	1	No axis overtravel limit switches are activated
PxxC.BBSUP	1	Drive power supply of the process is ready
PxxC.TPSUP	1	No overtemperature of the drive power supply
AxxC.MTAS	1	No axis overtemperature active for the process

PxxC.POWON Process Power ON Request

“0”= No process power request

“1”= Process Power ON request

This signal is used by the SPS to request Power On for the process. The request is ignored if the status of the Power Enable signal PxxS.POWEN is “0”. The signal PxxS.POWON assumes status ‘1’ if the power on is enabled by the CNC and requested by the SPS.

PxxS.POWON CNC Power ON

“0”= Power is not switched on by CNC

“1”= Power ON by CNC

This signal must be used in the SPS to switch the process power contactor on. The signal must be transferred from the SPS to the Main power contactor output without any further signal combinations.

NOTE: External conditions in the SPS program must be combined with the signal PxxC.POWON.

PxxS.POWER Power is ON

“0”= Process power is OFF

“1”= Process power is ON

This signal status of “1” indicates to the SPS that the process power is on and NC program execution with axis motion is possible.

Table 6-3: Conditions required for PxxS.POWER=“1”

SIGNAL	STATUS	DESCRIPTION
PxxS.POWON	1	CNC Power ON
PxxC.PWRDY	1	Drive power contactor is ON
PxxC.UDRDY	1	Drive bus voltage OK and present
PxxC.BBMSP	1	Main spindle drive ready

6.5 CNC Operating Modes

The CNC has different operating modes in which different functions and operations are executed.

Operating modes available:

- Automatic
- Semi-Automatic
- Manual
 - Axis parameter test
 - Jog Modes
 - Single Block Mode
 - Rapid Traverse
 - Manual Data Input

PxxC.MODE(y) CNC Operating Mode Selection (y=0,1)

Table 6-4: Operating mode selection

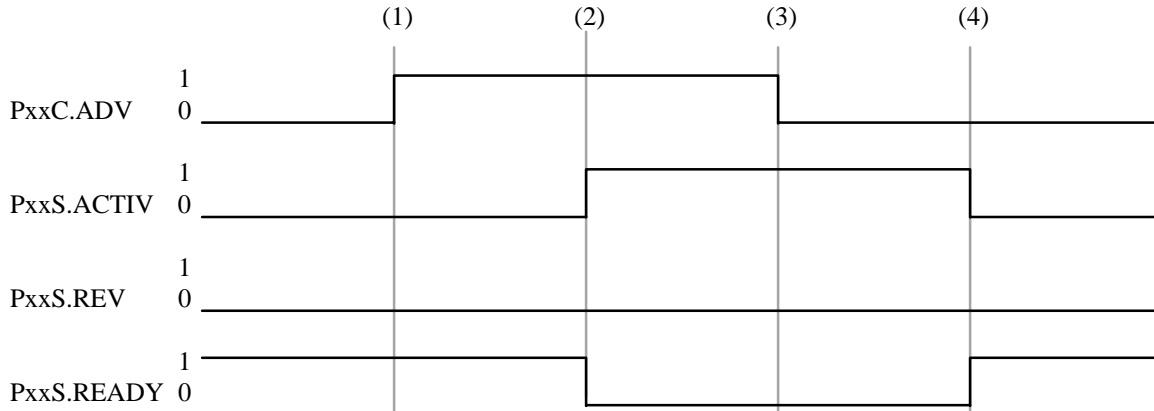
Operating Mode	PxxC.MODE0	PxxC.MODE1
Automatic	0	0
Semi-Automatic	1	0
Manual	0	1
Parameter Test (analog)	1	1

NOTE: Mode changes during NC-program execution will cause an immediate process stop.

6.5.1 AUTOMATIC MODE

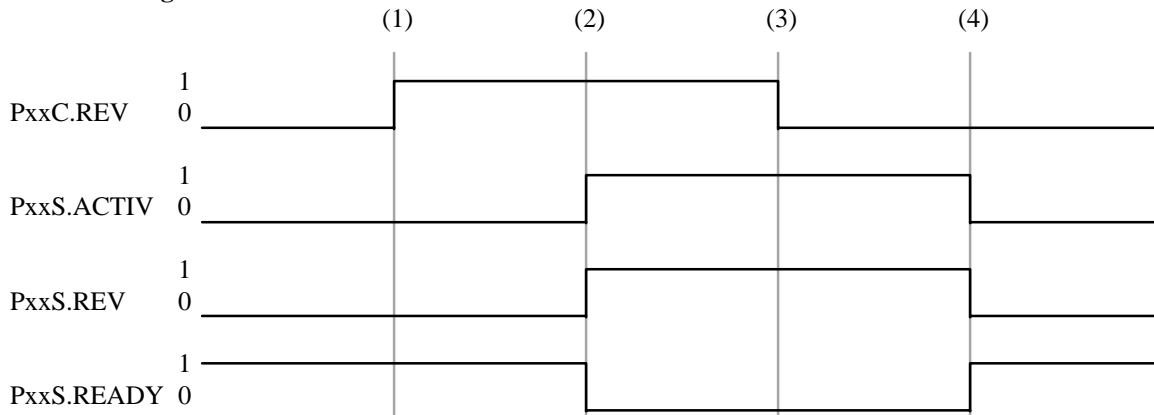
The CNC disables all manual axes functions and can execute the selected NC-program in either continuous, single cycle, or single block mode (selected in the SPS)

Advance Program Start:



- (1) NC-Part program is started with the Advance Program start (PxxC.ADV) switching from “0” to “1”.
- (2) PxxS.ACTIV is “1” during any NC-program execution. The signal PxxS.REV remains “0” if no reverse program is active, while the PxxS.READY signal switches to “0” when the NC-program is active.
- (3) The status of PxxC.ADV may switch back to “0” once the NC-program is started (PxxS.ACTIV=“1”) and single block mode not active.
- (4) The NC-program execution is finished after processing a NC command “BST” or “RET”. The status of PxxS.ACTIV switches back to “0” and is ready for start (PxxS.READY=“1”).

Reverse Program Start:

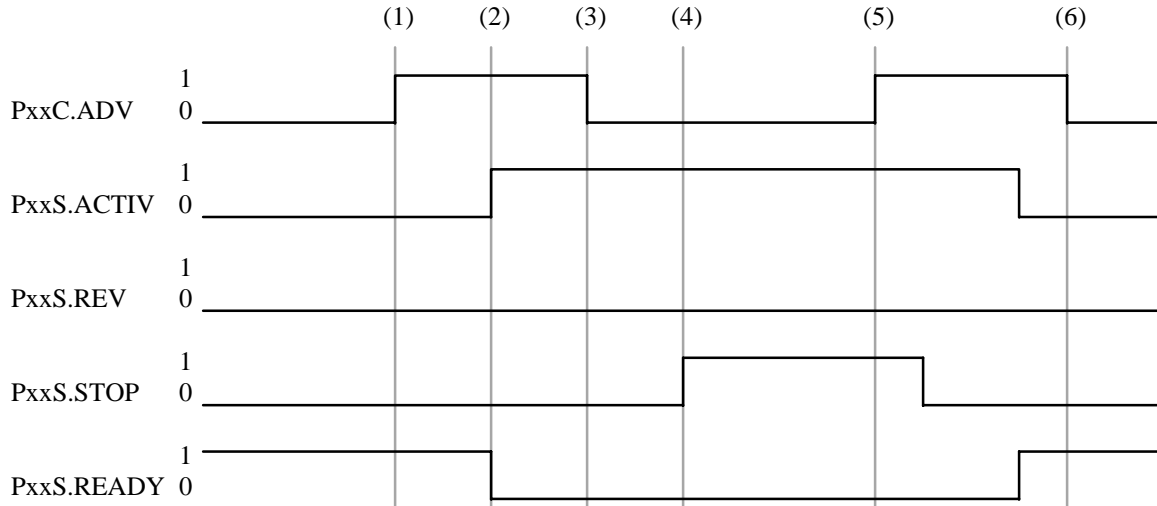


- (1) The reverse part program of the NC-program is started with the Reverse Program Start (PxxC.REV) switching from “0” to “1”.
- (2) PxxS.REV and PxxS.ACTIV =“1” indicate that the reverse NC-program is being executed. PxxS.READY has the status of “0” during the NC-program execution.
- (3) PxxC.REV may switch to “0” after starting the reverse program.
- (4) The NC-program execution is finished after processing a NC command “BST” or “RET”. The status of PxxS.ACTIV switches back to “0” and is ready for start (PxxS.READY=“1”).

6.5.2 SEMI-AUTOMATIC MODE

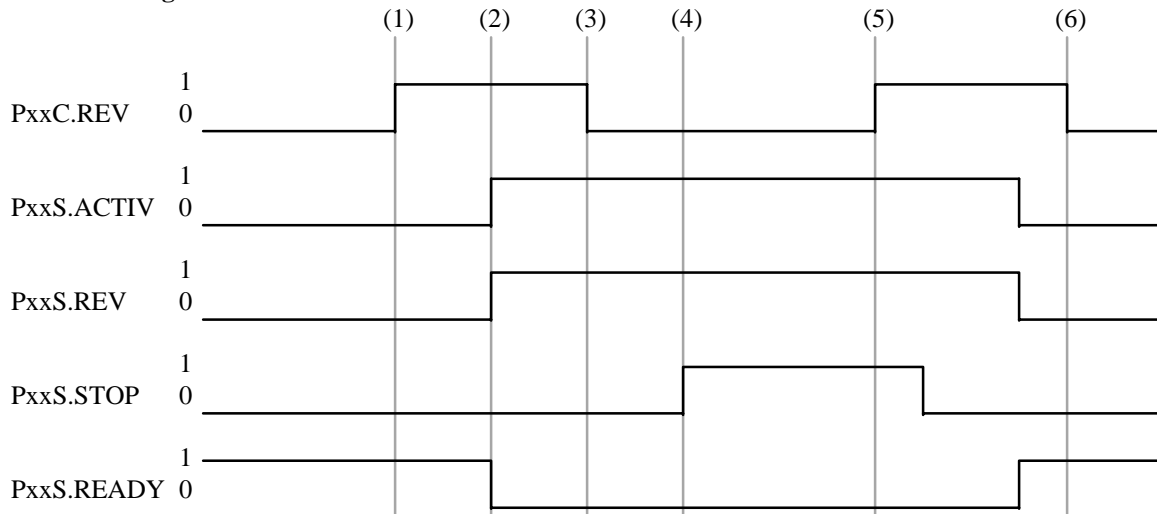
The CNC disables all manual axes functions and can execute the selected NC-program in either continuous ,single cycle, or single block mode (selected in the SPS)

Advance Program Start:



- (1) NC-Part program is started with the Advance Program start (PxxC.ADV) switching from “0” to “1”.
- (2) PxxS.ACTIV=“0” while the NC-program is being processed. PxxS.REV=“0” while the advance program is active. The status of PxxS.READY switches to “0” once the NC-program is started.
- (3) The NC-program is executed only while PxxC.ADV=“1”.
- (4) The NC-program stops immediately (PxxS.STOP=“1”) whenever the status of PxxC.ADV switches to “0”. This stop is similar to a feed hold for the process.
- (5) NC-program execution continues when the status of the signal (PxxC.ADV) switches back to “1”.
- (6) The NC-program execution is finished after processing a NC command “BST” or “RET”. The status of PxxS.ACTIV switches back to “0” and is ready for start (PxxS.READY=“1”).

Reverse Program Start:



- (1) The reverse part program of the NC-program is started with the Reverse Program Start (PxxC.REV) switching from “0” to “1”.
- (2) PxxS.REV and PxxS.ACTIV =“1” indicate that the reverse NC-program is being executed. PxxS.READY has the status of “0” during the NC-program execution.

- (3) The NC-program is executed only while PxxC.REV="1".
- (4) The NC-program stops immediately (PxxS.STOP="1") whenever the status of PxxC.REV switches to "0". This stop is similar to a feed hold for the process.
- (5) NC-program execution continues when the status of the signal (PxxC.REV) switches back to "1".
- (6) The NC-program execution is finished after processing a NC command "BST" or "RET". The status of PxxS.ACTIV switches back to "0" and is ready for start (PxxS.READY="1").

6.5.3 MANUAL MODE (Setup)

This mode disables the Automatic and Semi-Automatic mode. The CNC enables all manual axes functions depending on the selected Jog Mode and also allows NC-block execution via Manual Data Input (MDI).

PxxC.JOGM(y) Jog Mode Selection (y=0...2)

These modes are only active in Manual Mode. The Jog Mode specifies the type of axis jog movement. Predefined distances are specified as units, in a parameter. One unit is the last digit to the right of the decimal point (Bxx.002=4, then 1 unit = 0.0001)

Table 6-5: Jog Mode Selection

JOG MODE	PxxC.JOGM0	PxxC.JOGM1	PxxC.JOGM2
Continuous	0	0	0
1 unit	1	0	0
10 units	0	1	0
100 units	1	1	0
1000 units	0	0	1
10000 units	1	0	1
100000 units	0	1	1
Parametric	1	1	1

• Continuous Jog Mode:

The selected axis moves with the feedrate that is specified in parameter Cxx.020. The axis moves while the corresponding axis Jog commands, AxxC.JGPOS or AxxC.JGNEG have the status of "1".

NOTE: *Jog movements are allowed if the axis is not homed unless interlocks are implemented in the SPS program.*

• Incremental Jog Mode:

The selected axis moves the chosen incremental position if the status of the signal AxxC.JGPOS or AxxC.JGNEG changes from "0" to "1". One unit is specified as the smallest digit of the actual position resolution, specified in parameter Bxx.002.

NOTE: *A new Jog Increment will be started whether the previous increment was interrupted or not!*

• CNC Parametric Jog Mode:

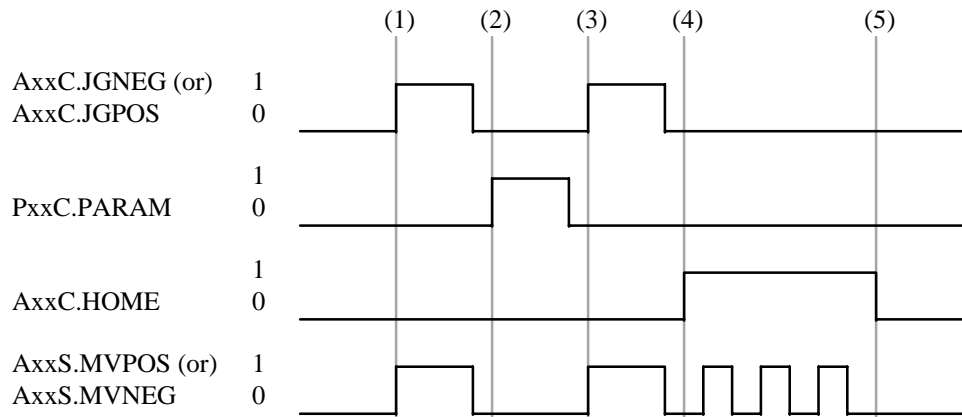
This mode works similar to the incremental Jog Mode. The axis moves the distance that is specified in the parameter Cxx.022.

NOTE: *A new Jog Increment will be started whether the previous increment was interrupted or not!*

PxxC.PARAM Parameter Test Mode

The best values for Maximum Axis Velocity, Maximum Axis Velocity Change and Maximum Axis Acc/Decel can be determined for an analog axis, using the parameter test mode. The oscilloscope functions of the MUI can be used for digital drives.

Automatic Parameter test , Analog Axis:



- (1) The automatic parameter test is performed within the distance specified via tracing the distance by jogging the axis. First the axis must be jogged to the test start position, using the AxxC.JGPOS or AxxC.JGNEG signals.
- (2) The axis position is stored by setting the status of the signal PxxC.PARAM="1". This position is the start positions used for the test movements.
- (3) The axis must then be jogged to its test end position. The distance between the test start and end position is used to automatically move the axis during the test.

NOTE: *The axis will move with its maximum velocity within the specified start and end position. Make sure that the axis is free of any possible collisions.*

- (4) The automatic test is started via the single axis home command (AxxC.HOME="1"). During the test the axis moves within the specified distance several times at maximum velocity. The test is performed only while the signal (AxxC.HOME="1"), the test will stop if (AxxC.HOME="0").
- (5) The test results are available in the MUI and may be used for optimizing the axis parameters and performance.

6.6 NC-Program Selection

The MT-CNC provides two memories for NC-programs (Memory A and Memory B). Each one of the memories can contain one NC-program package. A NC-program package can consist of up to 99 NC-programs (0..99) and one NC-cycle package (0) for each process.

PxxC.PRGNR NC-Program Selection with SPS

(0..99)= Unsigned short integer

The NC-program number must be transferred into the data byte PxxC.PRGNR of the corresponding process. The CNC confirms the NC-program selection by returning the selected program number in PxxS.PRGNR.

NOTE: *The NC-program number may be selected from within the process 0 NC-program (NC-command SP) via the MUI/GUI/ SOT (for process 0 only) or the SPS (PxxC.SP, PxxC.PRGNR). The NC-program that has been selected **last** will be executed with the next process start (PxxC.ADV).*

A new NC-program can be selected while another NC-program is active. The newly selected program requires a process advance/reverse start for its execution.

PxxS.PRGNR Selected NC-Program Number

(0..99)= Unsigned short integer

14VRS and below:

The contents of PxxS.PRGNR provides the number of the NC-program that is currently being selected (not currently active) from the time the CNC accepts the selection until the number is not selected anymore. It is necessary to compare the contents of PxxC.PRGNR and PxxS.PRGNR during the selection process to assure correct selection. The contents are not equal until the selection is accepted by the CNC.

NOTE: The NC-program number may be selected from within the process '0' NC-program (NC-command SP), via the MUI and SOT (for process #0 only) or the SPS. The NC-program that has been selected last will be executed with process start (PxxC.ADV/PxxC.REV). PxxC.PRGNR and PxxS.PRGNR containing the same number means only that the selected number was accepted by the CNC.

15VRS and above:

The contents of PxxS.PRGNR provides the number of the NC-program that is executed with the next advance or return signal. It is necessary to compare the contents of PxxC.PRGNR and PxxS.PRGNR during the selection process to assure correct selection. The contents are not equal until the selection is accepted by the CNC.

NOTE: The NC-program number may be selected from within the process '0' NC-program (NC-command SP), via the MUI/GUI/SOT (for process 0..6) or the SPS. The NC-program that has been selected **last** will be executed with the next process start (PxxC.ADV).

Using PxxS.PRGNR with external mechanisms (M7 - M31)

The NC-program for external mechanisms can be selected from within the NC-program of process '0'. The NC-command SP xx pp (xx = mechanism 7...31, pp = program number 0..99) writes the selected NC-program number pp into MxxS.PRGNR of the mechanism xx. The SPS can read the number from the PxxS.PRGNR and transfer it to the external mechanism.

PxxC.SP Selected NC-Program Number^{(3),(4),(5)}
 '0' = No NC-program selection
 '1' = Select NC-program contained in PxxC.PRGNR
 The state of '1' will select the NC-program number contained in the SPS program selection Gateway Signal PxxC.PRGNR.

ATTENTION: *No NC-program selection can be performed via MUI/GUI, SOT of NC-program if this signal is continuously set to a state of '1', since the program number contained in PxxC.PRGNR is always forced.*

6.7 Process Control Signals (SPS⇒CNC)

The following sections explain the functionality of a process, external mechanism and a management process. Each process or external mechanism has signals in the gateway that control the NC-program execution. These signals are also needed for a slave process and external mechanism control via the management process.

PxxC.ENABL Enable Process
 "0" = Process is not enabled
 "1" = Process is enabled
 NC-program execution and axis motion is only possible if (PxxC.ENABL="1"). If this signal switches to "0" while NC-program execution or axes jog movements the axes are stopped using a controlled ramp.

PxxC.ADV Start Advance Program
 "0" = Signal not active
 "0 ⇒ 1" = Advances the process program
 The function of the PxxC.ADV signal changes with different process operating modes.

Table 6-6: Requirements to Advance a NC-Program.

SIGNAL	STATUS	DESCRIPTION
PxxS.POWER	1	Process power ON
PxxS.ERROR	0	No process error
PxxC.ENABL	1	Process enabled
PxxS.REV	0	No reverse part program active
PxxS.READY	1	Process ready for start
PxxC.STOP	0	Process stop is not active
PxxC.SINGL	0 or (1)	If single cycle mode is selected (PxxC.SINGL="1") the advance NC-program will be interrupted after each NC-block.

- **Automatic Mode:**

A signal change from "0" to "1" starts the NC-program if all of the requirements are met. The signal PxxC.ADV must not maintain a status of "1" after the start. NC-program execution and axis motion may be stopped (feed hold) using PxxC.STOP. The NC-program can be continued after a (feed hold) by toggling PxxC.ADV if no process error had occurred.

- **Semi-Automatic and Manual Mode, except Parameter Test Mode:**

A signal change from "0" to "1" starts the NC-program if all of the requirements are met. A signal status of "1" is required at the signal PxxC.ADV to obtain axis movement and NC-program execution.

Block execution and axes motion is stopped, controlled ramp, if PxxC.ADV is switched to “0”. NC-program execution may continue at this point if the PxxC.ADV signal is switched to “1” and no process error had accrued.

PxxC.REV**Start Reverse Program**

“0”= Signal not active

“0” -> “1”= The signal change from ‘0’ to ‘1’, causes a reverse program start

The function of the PxxC.REV signal changes with different process operating modes.

Table 6-7: Requirements to Reverse a NC-program

SIGNAL	STATUS	DESCRIPTION
PxxS.POWER	1	Process power ON
PxxS.ERROR	0	No process error
PxxC.ENABL	1	Process is enabled

- **Automatic Mode:**

The signal change from ‘0’ to ‘1’ starts the NC-program if the process is enabled, has power and no error is active. The signal (PxxC.REV) must not maintain a status of ‘1’ after the start. NC-program execution and axis motion may be stopped (feed hold) using (PxxC.STOP). The NC-program can be continued from that position via (PxxC.REV = ‘1’) if no process error occurred.

- **Semi-Automatic and Manual mode, except parameter test mode**

The signal change from ‘0’ to ‘1’ starts the NC-program if the process is enabled, has power and no error is active. A signal status of ‘1’ is required for the signal (PxxC.REV) to perform axis movement and NC-program execution.

NC-block execution and axes motion is stopped, ramp controlled, if the signal status of (PxxC.REV) switches to ‘0’. NC-program execution may be continued from that position while the signal (PxxC.REV) has a status of ‘1’. The NC-program execution can not be continued if a process error occurred.

PxxC.STOP**Process Stop (Feed Hold)**

“0”= No SPS process stop

“0 ⇒ 1”= SPS process stop

A process stop occurs if this signal toggles from a status of “0” to “1”. NC-program execution and all axis movements are stopped under a controlled ramp. The process can be restarted with a PxxC.ADV or PxxC.REV signal if no process error had occurred. Manual axis jogging is not influenced by PxxC.STOP.

PxxC.SINGL**CNC Single Block Mode**

“0”= Continuous block mode active

“1”= Single block mode active

A status “1” of this signal stops process program execution (PxxS.STOP=“1”) after each NC-program block. The execution of the next advance program block can be started with a change from “0” to “1” at PxxC.ADV. Reverse program blocks can be started with a signal change from “0” to “1” at PxxC.REV.

- PxxC.BLSKP** **Skip NC-block**^{(2),(3),(4),(5)}
 “0”= NC-blocks are not skipped
 “1”= NC-blocks are skipped
 NC-blocks that are marked by a forward slash (/) are skipped when this signal is set to ‘1’. A series of blocks can only be skipped if (PxxC.SINGL = ‘0’) and (PxxC.BLSKP = ‘1’) before the first NC-block of the series has been processed. In single block mode (PxxC.SINGL = ‘1’), the NC checks prior to processing each block whether skipping the block has been activated through the signal (PxxC.BLSKP = ‘1’). The block is not processed if the signal is ‘1’ before the NC-block is processed.
- PxxC.M001** **Conditional Stop**^{(2),(3),(4),(5)}
 “0”= M01 does not interrupt NC-program execution
 “1”= M01 interrupts NC-program execution
 If this signal is set to ‘1’, programming the ‘M01’ function in the NC-program causes the program to stop. The auxiliary function is transferred to the PLC and must be acknowledged. The NC interprets the (PxxC.M001) signal after the M-function has been acknowledged. The (PxxS.STOP) signal is not influenced when the NC-program is stopped.
- NOTE:** *A change of the (PxxC.M001) interface signal state after a program interruption does not have an effect on the program interruption.*
- PxxC.RAPID** **CNC Rapid Traverse**
 “0”= No CNC rapid traverse
 “1”= CNC rapid traverse
 This signal is only active in Manual mode after the axis has been homed. The signal (PxxC.RAPID=“1”) selects rapid traverse for all process axes. The rapid feedrate is specified individually for each axis in its axis parameter Cxx.021.
- NOTE:** *In versions 15VRS and above this signal selects rapid traverse for all process axes except main spindles, AxxC.RAPID must be used for rapid traverse of main spindles.*
- PxxC.REPOS** **Repositioning to the Contour**^{(3),(4),(5)}
 “0”= No activation of repositioning
 “1”= Repositioning can be performed
 This signal is only activated in the NC if the operator has interrupted the NC-program execution in Automatic, Semi-Automatic, Program Execution in Manual Mode (PxxS.ACTIV = ‘1’), and has left the contour by axis jogging in Manual mode. The NC ignores the interface signal in all other situations.
- During repositioning in manual mode, the jog keys respond like in the normal jog mode. The NC ensures that the axes do not overrun the start or target position.
 - In the program modes, executing a subroutine ‘.REPOS’ is used to restore the state that existed before the interruption. The (PxxC.ADV) signal is used to start the subroutine.
- PxxC.RESTA** **Restarting to the Contour**^{(3),(4),(5)}
 “0”= No activation of restart
 “1”= Restart can be performed
 The PLC sets this signal to activate the restart option. The NC interprets the signal in the same way as repositioning. With respect to restarting in manual mode and in the program modes, the NC also refers to the REPOS subroutine. The same rules are valid for restart and for repositioning.

PxxC.CLEAR Clear Error / Control Reset

“0”= Signal not active

“1”= Clear Error in process if an error is active or
Control Reset if no process error is active.

This signal is valid in all operating modes and has 2 functions:

(1) Clear Error (PxxS.ERROR=“1”)

A process error can be cleared with a PxxC.CLEAR signal change from “0” to “1”. The error will return immediately if the error cause has not been removed. Advance programs that have been interrupted by a process error can't be restarted after the error is cleared if the process parameter Bxx.032 is set to “yes”. The reverse program must be executed first (PxxC.REV=“0 ⇒1”). The program execution starts with the NC-block which contains the label of the last processed reverse vector. The block with the label “.HOME” will be processed if no reverse vectors were active.

(2) Control Reset (PxxS.ERROR=“0”)

A control reset to performed if the status of PxxC.CLEAR switches from “0” to “1” and no process error is active for the same process. The SPS program must contain the correct logic to prevent an accidental Control Reset, especially if reverse vectors are used.

CAUTION: *When performing a Control Reset the active reverse vectors will be cleared! Recovery sequences that are programmed via Reverse vectors will not be executed anymore! The basic reverse vector “.HOME” will be the only reverse vector active.*

6.8 Process Status Signals (CNC⇒SPS)

These signals are global process status signals and are valid with or without using a management process.

PxxS.ERROR Process Error is Active

“0”= No process error active

“1”= Process error active

This signal is valid in all operating modes. The MUI and SOT will display a diagnostic message text or number if PxxS.ERROR=“1” for the active error. A process error can be reset via PxxC.CLEAR. The error can not be cleared if the cause is still apparent.

PxxS.READY Process is Ready for Start

“0”= Process not ready for advance start

“1”= Process is ready for advance start

The signal indicates that the process is ready for an advance NC-program start. Its status depends on the setting of the following process parameters:

- **Process Parameter Bxx.032**

- **“Reverse Program Execution Required”**

- The reverse program must be executed if this parameter is set to “yes” and (PxxS.READY=“0”). Any type of manual axis movement will reset PxxS.READY to “0”, manually homing the process axes will not reset PxxS.READY to “1”, only activating the reverse program will reset PxxS.READY to “1”.

- **Process Parameter Bxx.033**

- **“Reference Required”**

- All incremental process axes must be homed using (G74) in the reverse program in order to receive a status of “1” at PxxS.READY. Process parameters Bxx.032 and Bxx.033 both must be set to “yes” if the process operates one or more incremental axes.

- The status of PxxS.READY is set to “1” after the CNC commands “BST” or “RST” are executed in the advance or reverse program. PxxS.READY is reset to “0” after the start of the advance program.

PxxS.RUN NC-program Block Active

“0”= CNC process is not executing a program block

“1”= CNC is executing a program block

(PxxS.RUN=“1”) indicates that a NC-program block is currently being executed. The status of PxxS.RUN remains “1” if a process stop occurs during the NC-program execution.

(PxxS.RUN=“0”) whenever the CNC commands ‘BST’, ‘RET’, ‘M02’ or ‘M30’ are executed in the NC-program. (PxxS.RUN=“0”) after each NC-block if single block mode is active. Any type of manual axis , except tool magazine, movement or a Control Reset also resets the signal PxxS.READY to “0”.

PxxS.ACTIV Process Program Active

“0”= Process program is not active

“1”= Process program is active

(PxxS.ACTIV=“1”) indicates that a NC-program is currently being executed. The status of this signal will remain “1” if a process stop occurs during NC-program execution. If single block mode is active the status of PxxS.ACTIVE will remain “1” until the CNC commands “BST” or “RET” are executed in the NC-program.

PxxS.ACTIV will be reset to “0” whenever the CNC commands ‘BST’, ‘RET’, ‘M02’ or ‘M30’ are executed in the NC-program. Any manual axis movements or a Control Reset also resets the signal to “0”. Manual tool magazine movements do not influence this signal.

- PxxS.REV Process Reverse Program is Active**
 “0”= CNC reverse program is not active
 “1”= CNC reverse program is active
 (PxxS.REV=“1”) indicates that a reverse NC-program is currently being executed. PxxS.REV will remain “1” if a process stop occurs during the reverse program. If single block mode is active the status of PxxS.REV will remain “1” until the CNC commands ‘BST’, ‘RET’, ‘M02’ or ‘M30’ are executed in the NC-program.
 PxxS.REV will be reset to “0” whenever the CNC commands ‘BST’, ‘RET’, ‘M02’ or ‘M30’ are executed in the NC-program. Any manual axis movements or a Control Reset also resets the signal to “0”. Manual tool magazine movements do not influence this signal.
- PxxS.STOP Process Program Stopped**
 “0”= Process program execution has not stopped
 “1”= Process program execution has stopped
 This signal status switches to “1” after program execution and all process axes motion has stopped. PxxS.STOP remains “0” if single block mode is active and no PxxC.STOP is issued.
 PxxS.STOP switches to “0” if the NC-program is restarted via PxxC.ADV or PxxC.REV. Any manual axis movements or a Control Reset also resets the signal to “0”. Manual tool magazine movements do not influence this signal.
- PxxS.POWIN Process Control Power Interrupted**
 “0”= Process control power not interrupted
 “1”= Program stop due to process power interrupt
 This signal indicates that the power of the MT-CNC was interrupted during NC-program execution. The status of PxxS.POWIN is reset to “0” by starting the reverse program. The process will be ready for process start (PxxS.READY=“1”) after the reverse program is executed.
 Any manual axis movements or a Control Reset also resets the signal to “0”. Manual tool magazine movements do not influence this signal.
- PxxS.MDIAC Manual Data Input (MDI) active**
 “0”= No MDI NC-program block ready for execution
 “1”= MDI NC-program block active or ready for execution.
 MDI mode is activated for the process if a NC-block has been entered in MDI, was transferred to the MT-CNC, and is ready for execution. PxxS.MDIAC changes its state to “1” after a NC-program block has been transferred to the MT-CNC via the MUI, GUI or SOT while entered in MDI. The NC-program block is executed while the signal PxxC.ADV has a status of “1”.
 PxxS.MDIAC remains “1” until the NC-program block is executed or MDI is aborted. The MDI mode can be aborted (PxxS.MDIAC=“0”) via the start of a reverse NC-program, any type of axis jogging, control reset, or a MT-CNC power loss.
- PxxS.TRANS Translation Active**
 “0”= Translation is not active
 “1”= Translation is active
 This signal is used to inform the PLC whether or not the translation from Cartesian coordinates into polar coordinates has been activated.

- PxxS.CREST Repositioning / Restart Conditions Restored^{(3),(4),(5)}**
“0”= Repositioning / Restart conditions are not restored
“1”= Repositioning / Restart conditions are restored (completed)
The NC sets this status signal to notify the PLC that all axes relevant to repositioning or restart have assumed the state they had before the interruption. In this context, relevant axes are:
- all spindles,
 - main spindles with rotary axis capability,
 - combined turret/spindle axes that are in spindle mode,
 - all feed axes whose ‘Adjustment window for...’ axis parameter is different from ‘0’, and
 - the tool magazine axis if the ‘Reposition tool storage axis’ process parameter has been set to ‘Yes’.
- Provided that the related control signals (PxxC.REPOS, PxxC.RESTA) have been set, the NC only updates the status signal during repositioning or restart.
- PxxS.REPOS Repositioning / Restart Active^{(3),(4),(5)}**
“0”= Repositioning / restart is not active
“1”= Repositioning / restart is active
The NC sets this signal to notify the PLC that repositioning or restart has been activated.
- PxxS.G00 Rapid Traverse with G0 Active^{(3),(4),(5)}**
“0”= Rapid traverse (G00) not active
“1”= Axis movement is performed at rapid traverse rate (G00)
This status signal informs the PLC whether there is a movement with rapid traverse rate (G00) of the related process.
- PxxS.G74 Active Homing of Feed Axes with G74^{(3),(4),(5)}**
“0”= Homing a feed axis (G74) not active
“1”= Homing a feed axis (G74) active
This signal informs the PLC whether homing with ‘G74’ is active for a feed axis.
- PxxS.G08 Velocity Optimized Block Transition Active^{(3),(4),(5)}**
“0”= Velocity optimized block transition (G08) not active
“1”= Velocity optimized block transition (G08) active
This signal informs the PLC whether a velocity-optimized block transition (G08) is active.
- PxxS.TREAD Threading or Tapping Active^{(3),(4),(5)}**
“0”= Threading/tapping not active
“1”= Threading/tapping active
This signal informs the PLC whether a threading or tapping G-code (G33, G63, G64, G65) is active for the related process.
- PxxS.G96 G-code for Constant Surface Speed Active^{(3),(4),(5)}**
“0”= G96 not active
“1”= G96 active
This signal informs the PLC whether the G-code for constant surface speed (G96) is active for the related process.
- PxxS.PROC Process Number**
INT = Process number
This signal is used to inform the PLC the process number.

6.9 Feed and Spindle Override (SPS⇒CNC)

These functions allow the user to change spindle speed and axis feedrate via the SPS program.

PxxC.FOVRD Feed Override

“0...255”= USINT representing % of Feed Override

The feedrate override influences the velocity of linear and rotary axis. This signal is active in operating all modes and valid for all axis movements, except during homing of digital servo drives (DDS).

The range of feedrate override is “0...255” (decimal Unsigned Short Integer) which is equal to “0...255%” in the CNC. The axis velocity is limited to the value entered in the Maximum Feedrate (parameters).

NOTE: *In version 14VRS and above only the main axes are influenced via PxxC.FOVRD. The corresponding axis signal AxxC.OVRD must be used to control the feed override of NON-interpolating axes (e.g., tool storage axis).*

NOTE: *Since the default of PxxC.FOVRD is 0%, the Feedrate Override and its limits must be generated within the SPS program.*

PxxC.SOVRD Spindle Override^{(1),(2)}

“0...255”= USINT representing % of Spindle Override

The spindle override influences the velocity the process main spindles. This signal is active in operating all modes and valid for all spindle movements, except during homing of a digital main spindle.

The range of spindle override is “0...255” (decimal Unsigned Short Integer) which is equal to “0...255%” in the CNC. The spindle velocity is limited to the value entered in the Maximum Feedrate (parameters).

NOTE: *Since the default of PxxC.SOVRD is 0%, the Spindle Override and its limits must be generated within the SPS program.*

NOTE: *Only main axes are influenced via PxxC.FOVRD. The corresponding axis signal AxxC.OVRD and AxxC.RAPID must be used to control the feed override of NON-interpolating axes (e.g., tool storage axis). Assign the value of PxxC.FOVRD to PxxC.ROVRD if no extra rapid override switch exists.*

NOTE: *In version 15VRS and above the signal PxxC.SOVRD is replaced by PxxC.ROVRD.*

PxxC.ROVRD Rapid Override^{(3),(4),(5)}

PxxC.ROVRD = 0...255 (%), 16#00...FF

In software releases 0X.15/00 and up, PxxC.ROVRD replaces PxxC.SOVRD. See AxxC.OVRD for spindle override if axis xx is defined as a spindle axis.

The rapid feedrate override influences the path velocity of all linear and rotary main axis that participate in G00 interpolation. The rapid feedrate override is processed by the CNC independent of the active operating mode. The rapid feedrate override is active at all process main axis G00 movements.

The range of the feedrate override is 0...2550 (decimal Unsigned Short Integer) which is equal to 0...255% in the CNC. The axis velocity is limited by the CNC to its maximum velocity (parameters) if the selected override would exceed the maximum (parameter) axis velocity. The G00 interpolation remains active since the feedrate of all axes is reduced accordingly.

NOTE: *The Rapid Feedrate Override and its limits must be generated within the SPS program since the default of PxxC.ROVRD is 0%. A status message is automatically issued to the MUI or SOT by the MT-CNC if the feedrate override is at 0%.*

NOTE: *Only main axes are influenced via PxxC.FOVRD. The corresponding axis signal AxxC.OVRD and AxxC.RAPID must be used to control the feed override of NON-interpolating axes (e.g., tool storage axis). Assign the value of PxxC.FOVRD to PxxC.ROVRD if no extra rapid override switch exists.*

NOTE: *In version 15VRS and above the signal PxxC.SOVRD is replaced by PxxC.ROVRD.*

6.10 Process / Part Program Synchronization

The MT-CNC contains 7 processes (0...6), process 0 can be used as a Management Process to control processes (1...6). The CNC provides special NC-commands in the Management Process (0) that influence certain Gateway signals. The SPS is used to link these required Gateway signals of the Management Process (0) and Slave Process (1...6) together.

Pxx(y).DP Define Process (y=C,S)**Table 6-8: Define Process Signals**

SIGNAL	STATUS	DESCRIPTION
PxxS.DP	0	Process is not defined in Mgmt. process
PxxS.DP	1	Process defined in Mgmt. process
PxxC.DP	0	Process is not defined
PxxC.DP	1	Define process

P :P for Process, M for External Mechanism
 xx :Process (0...6), External Mechanism (7...31)
 C,S :C for Control signal, S for Status signal

A slave process must be defined in the management process first before it can be advanced or reversed.

The SPS reads the signal PxxS.DP and transfers it's status with PxxC.DP to the process defined in the management process via the NC-command DP xx.

PxxS.LP

Lock Process

Table 6-9: Lock Process Signals

SIGNAL	STATUS	DESCRIPTION
PxxS.LP	0	Do not lock process
PxxS.LP	1	Lock (disable) process

P :P for Process, M for External Mechanism
 xx :Process (0..6), External Mechanism (7...31)
 C,S :C for Control signal, S for Status signal

A slave process can be locked via the NC-command "LP xx" in the Mgmt. NC-program. A locked condition can disable axis motion for that process. The signal PxxS.LP has a status of "1" until the NC-command 'BST', 'RET', 'M02' or 'M30' is executed in the Mgmt. process.

The NC-command "LP xx" in the Mgmt. program sets the status of the corresponding Gateway signal (PxxS.LP="1").

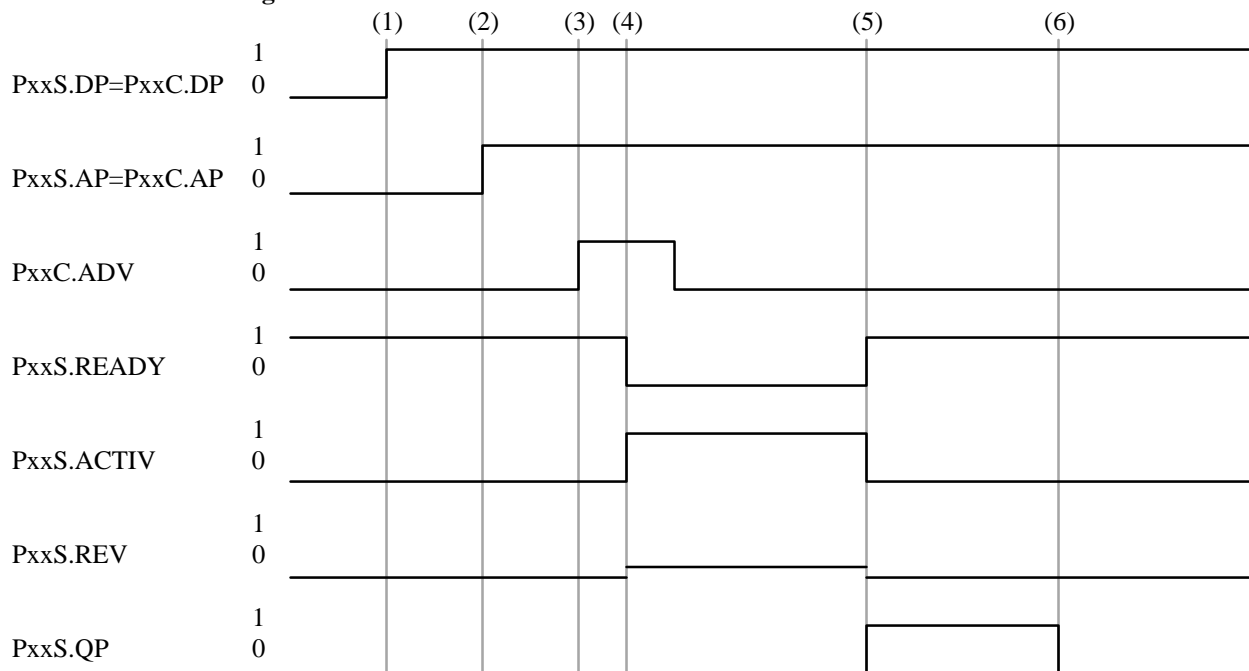
NOTE: *A system error is generated for the Mgmt. process if a process that is being locked is not defined. The conditions to lock a process must be programmed in the SPS.*

Pxx(y).AP Advance Process (y=C,S)**Table 6-10: Advance Process Signals**

SIGNAL	STATUS	DESCRIPTION
PxxS.AP	0	No advance process from the Mgmt. proc
PxxS.AP	1	Advance process from the Mgmt. process
PxxC.AP	0	No advance process
PxxC.AP	1	Advance process

P :P for Process, M for External Mechanism
 xx :Process (0..6), External Mechanism (7...31)
 C,S :C for Control signal, S for Status signal

The NC-command AP xx must be used in the Mgmt. NC-program to advance a process or external mechanism. The slave process or mechanism must have been defined (DP xx) previously in the Mint. NC-program.

Time Diagram for Slave Process Advance:

- (1) The slave process is defined DP xx in the Mgmt. NC-program and confirmed in the SPS program using the (PxxS.DP and PxxC.DP) signals.
- (2) The PxxS.AP signal is generated from the NC-command AP xx in the Mgmt. program.
- (3) The advance slave process start signal (PxxC.ADV) is derived from the (PxxC.AP) signal in the SPS.
- (4) (PxxS.ACTIV="1") and (PxxS.READY="0") when the slave process executes the advance process program section.
- (5) (PxxS.ACTIV="0") and (PxxS.READY="1") after the slave process advance NC-program is finished. The slave process acknowledges with (PxxS.QP="1"), using the NC-command POK, at the end of the slave process program execution.
- (6) The slave recognizes the status "0" of the PxxS.AP signal and resets its PxxS.QP signal to "0".

Pxx(y).RP Reverse Process (y=C,S)

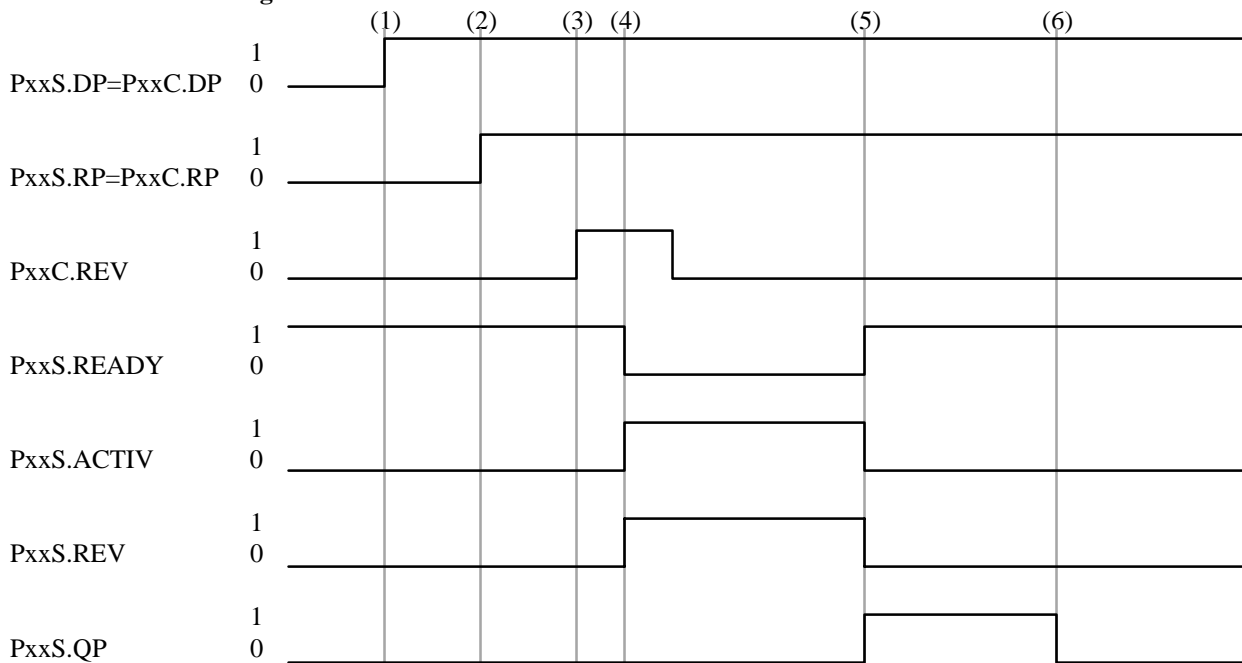
Table 6-11: Reverse Process Signals

SIGNAL	STATUS	DESCRIPTION
PxxS.RP	0	No reverse process from the Mgmt. proc
PxxS.RP	1	Reverse process from the Mgmt. process
PxxC.RP	0	No reverse process
PxxC.RP	1	Reverse process

P :P for Process, M for External Mechanism
 xx :Process (0..6), External Mechanism (7..31)
 C,S :C for Control signal, S for Status signal

The NC-command RP xx must be used in the Mgmt. NC-program to reverse a process or external mechanism. The slave process or mechanism must have been defined (DP xx) previously in the Mgmt. NC-program.

Time Diagram for Slave Process Reverse:



- (1) The slave process is defined DP xx in the Mgmt. NC-program and confirmed in the SPS program using the (PxxS.DP and PxxC.DP) signals.
- (2) The PxxS.RP signal is generated from the NC-command RP xx in the Mgmt. program.
- (3) The reverse slave process start signal (PxxC.REV) is derived from the (PxxC.RP) signal in the SPS.
- (4) (PxxS.ACTIV="1"), (PxxS.READY="0"), and (PxxS.REV="1") when the slave process executes the reverse process program section.
- (5) (PxxS.ACTIV="0") and (PxxS.READY="1") after the slave process reverse NC-program is finished. The slave process acknowledges with (PxxS.QP="1"), using the NC-command POK, at the end of the slave process program execution.
- (6) The slave recognizes the status "0" of the PxxS.RP signal and resets its PxxS.QP signal to "0".

Pxx(y).QP Process Acknowledgment (y=C,S)

Table 6-12: Process Acknowledgment Signals

SIGNAL	STATUS	DESCRIPTION
--------	--------	-------------

PxxS.QP	0	No process Ack. from slave process
PxxS.QP	1	Process Ack. from the slave process
PxxC.QP	0	No Ack. to the management process
PxxC.QP	1	Process Ack. to the Mgmt. process

The PxxS.QP signal is set to “1” after the slave process has finished its advance or reverse program execution using the NC-commands ‘BST’, ‘RET’, ‘M02’ or ‘M30’.

The signal PxxS.QP is reset “0” whenever the signal PxxC.AP or PxxC.RP from the SPS is set to “0”. The status of these signals is controlled by the Mgmt. process. Their status is reset to “0” after the Mgmt. process received a status of “1” on the PxxC.QP signal from the slave process.

Pxx(y).POK Process Full Depth (y=C,S)

Table 6-13: Process Full Depth Signals

SIGNAL	STATUS	DESCRIPTION
PxxS.POK	0	No full depth
PxxS.POK	1	Process full depth
PxxC.POK	0	No full depth
PxxC.POK	1	Process full depth

The status of PxxS.POK depends on the use of the NC-command POK in the management and slave program:

(1) The NC-command POK sets the status of PxxS.POK immediately to “1” if **no other processes have been defined** previously in process xx. In this case the process program in which POK is programmed does not contain the NC-command DP xx.

(2) The PxxS.POK status of a process in which **other processes have been defined** previous to the NC-command POK is set to “1”, after execution of POK and receiving of PxxC.POK=“1” of all defined processes.

The signal PxxS.POK is reset to “0” if the process is advanced or reversed. Also a process Control Reset or manual axis jogging will reset PxxS.POK to “0”.

6.11 Spindle Control (SPS⇒CNC)

A process spindle is specified in the system parameters by assigning it to one of the possible 20 axes. A spindle can be controlled like an axis however, a spindle and axes normally depend on each other. Further signals are therefore required to control a spindle.

PxxC.SPSTP Stop Spindle Together with Process^{(1),(2)}

“0”= No spindle stop

“1”= Stop spindle together with process

If (PxxC.SPHLT=“1”) the spindle is stopped together with the process, and the process is commanded to stop (PxxC.SPHLT=“1”).

NOTE: In version 14VRS and above PxxC.SPSTP has been replaced by AxxC.SPSTP.

PxxC.SPHLT Stop Spindle if Process Stopped^{(1),(2)}

“0”= No spindle stop

“1”= Stop spindle if process already stopped

The spindle may be stopped (PxxC.SPHLT=“1”) if the process has already stopped (PxxS.STOP=“1”).

NOTE: In version 14VRS and above PxxC.SPHLT has been replaced by AxxC.SPHLT.

PxxC.N_CMD Spindle at Commanded Speed^{(1),(2)}

“0”= Spindle at commanded speed

“1”= Spindle NOT at commanded speed

This signal is required by the CNC whenever external spindle drives are used. CNC monitors the signal during spindle speed changes (accel/decel) when advancing or reversing the process. Refer to the signal AxxS.N_CMD when using SERCOS based digital spindle drives.

NOTE: *In version 14VRS and above PxxC.N_CMD has been replaced by AxxC.N_CMD.*

NOTE: *The SPS must contain the logic to monitor the N=CMD of the main spindle drive.*

6.12 Synchronous Axes**PxxC.SCONn Activate Synchronous Axes Group^{(3),(4),(5)}**

“0”= De-activate synchronous axes group

“1”= Activate synchronous axes group

(n = 1...4)

This signal can be used for synchronously utilizing up to four feed axes. This functionality is required for ‘slave’ or ‘Gantry’ axes functions. The PLC sets the (PxxC.SCONn) signal to activate the related synchronous axes group. The NC cancels synchronous operation as soon as the PLC resets the control signal. A maximum of four synchronous axes groups (PxxC.SCON1...PxxC.SCON4) can be defined. The allocation of the axes to the synchronous axes group is done in the machine data.

NOTE: *If the PLC activates a synchronous axes group asynchronously to NC-program execution without auxiliary function output, the NC will not activate the synchronous axes group before all blocks of the NC preparation memory have been processed. If an error occurs that affects the synchronous axes group, that error can only be cleared if the synchronous axes group is de-activated before the Clear Error signal is issued.*

PxxC.SSnMT Minimize Torsion Moment^{(3),(4),(5)}

“0”= No minimization torsion for synchronized spindle

“1”= Minimized torsion moment for synchronized spindle

(n = 1,2)

The CNC minimizes mechanical torsion (preventing torque build up due to control loop integration time differences) between main spindle (1 / 2) and its lead spindle while (PxxC.SS1MT, PxxC.SS2MT) has the state of ‘1’.

NOTE: *This signal is only evaluated by the CNC if Main Spindle Synchronization of spindle (1/2) are active (PxxC.SS1ON / PxxC.SS2ON = ‘1’).*

- PxxC.SSnON Spindle Synchronization ON^{(3),(4),(5)}**
“0”= Spindle Synchronization OFF
“1”= Spindle Synchronization ON
(n = 1,2)
Main spindle synchronization between the lead and slave spindle is activated while (PxxC.SS1ON / PxxC.SS2ON) has a status of ‘1’. The synchronization is deactivated immediately when switching to a status of ‘0’. This signal is valid in all operating modes.
- PxxS.SCONn Synchronous Axes Group Activated^{(3),(4),(5)}**
“0”= Synchronous axes group not activated
“1”= Synchronous axes group activated
(n = 1...4)
The NC sets the corresponding status signal to ‘1’ if one of the four possible synchronous axes groups (PxxC.SS1ON1, PxxC.SS1ON2, PxxC.SS1ON3 or PxxC.SS1ON4) has been activated. The NC resets this signal:
- if the PLC removes the controller enabling signal (AxxC.ENABL) from one axes of the synchronous group during the movement.
 - as soon as the NC losses control of the drives (e.g., residual energy of the DC bus is insufficient for decelerating the drives)
 - in conjunction with axis transfer the NC returns the transferred axes to the related process after a ‘Control Reset’ and/or the end of NC-program execution (BST, RET, M02, M30).
- PxxS.SSnOK Synchronous Operation is OK^{(3),(4),(5)}**
“0”= The amount of actual position difference exceeds the synchronous operation window.
“1”= The amount of actual position difference is smaller than or equal to the synchronous operation window.
(n = 1,2)
The NC uses this signal to indicate whether or not the amount of the actual position difference exceeds the synchronous operation window. The synchronous operation window is used for continually monitoring synchronous operation. The NC checks the difference between the actual position value of the leadscrew and the actual position value of the synchronous spindle, taking the corresponding gearing ration into consideration.
- PxxS.SSnER Synchronous Execution Error Limit^{(3),(4),(5)}**
“0”= The amount of the actual position value difference is smaller than or equal to the synchronous operation error limit.
“1”= The amount of the actual position value difference exceeds the synchronous operation error limit.
(n = 1,2)
The NC uses this signal to indicate whether or not the amount of the actual position value exceeds the synchronous operation error limit. The synchronous operation error limit is used as a limit value for continually monitoring the actual position value difference, taking the corresponding gearing ration into account.

6.13 Mechanism Control Signals

MxxC.POK Part Machined

“0”= Part machining not completed

“1”= Part machining completed

The system must be set to logic ‘1’ if a ‘POK’ has been programmed within a MASTER NC-program and the mechanism has been defined as a subordinate unit in that process. Refer to the explanation of the internal process signal ‘PxxC.POK’ for details on the ‘part machined’ signal. If an external mechanism has been allocated to a process in a SLAVE MT-CNC, the ‘PxxS.POK’ signal of the SLAVE process must be mapped via the I/O level onto the ‘MxxC.POK’ signal in the MASTER controller.

MxxC.QP Acknowledge Mechanism Execution

“0”= No Acknowledgment that mechanism has completed execution

“1”= Acknowledgment that mechanism has completed execution

This signal must be set if, within a MASTER NC-program, a ‘WP’ has been programmed for the corresponding mechanism, and the process that is allocated to the mechanism has completed its program. If the external mechanism has a process in a SLAVE MT-CNC allocated, the ‘PxxS.QP’ signal must be mapped via the I/O level onto the ‘MxxC.QP’ signal in the MASTER controller.

6.14 Mechanism Status Signals

MxxS.DP Define Mechanism

“0”= Mechanism has not been defined

“1”= Mechanism has been defined

The NC sets this signal if the ‘DPxx’ command (xx = mechanism number) has been programmed within a NC-program. Refer to the explanation of the internal process signal ‘PxxS.DP’ for details about the ‘define process’ signal.

MxxS.RP Reverse Program Start

“0”= No reverse program start from the MASTER NC-program

“1”= Reverse program start from the MASTER NC-program

The ‘reverse program enabling’ signal is used for informing the PLC program that an ‘RP’ reverse program start has been programmed for that mechanism in the higher level process. Refer to the explanation of the process related interface signal ‘PxxS.RP’ for more information about this signal.

MxxS.AP Advance Program Start

“0”= No advance program start from the MASTER NC-program

“1”= Advance program start from the MASTER NC-program

The ‘advance program enabling’ signal is used for informing the PLC program that an ‘AP’ advance program start has been programmed for that mechanism in the higher level process. Refer to the explanation of the process related interface signal ‘PxxS.AP’ for more information about this signal.

MxxS.LP Disable Mechanism

“0”= Mechanism not disabled

“1”= Disable mechanism

This signal informs the PLC that a ‘LP’ disable mechanism has been programmed for that mechanism in the higher level process. Refer to the explanation of the process related interface signal ‘PxxS.LP’ for more information.

MxxS.PRGNR Mechanism Program Number

USINT = Program number

The CNC uses this signal to inform the PLC the program number that has been selected for the corresponding mechanism. This signal is used if the external mechanism is a MT-CNC process whose machining program is selected via the higher level process. Refer to the explanation of the process related interface signals (PxxC.PRGNR and PxxS.PRGNR).

MxxS.PROC Mechanism Number

INT = Mechanism

This signal is used to inform the PLC the mechanism number.

6.15 Axis Signals

Each of the 20 axes available in the MT-CNC has it's own axis interface between CNC and SPS. The axis signals are processed by the MT-CNC only if the axis is specified in the system parameters.

Table 6-14: Axis Signal Structure

A03C.OVTRL	
A	A=Axis
03	.
C	C=control signal / S=status signal
“.”	Separator between user defined and system defined identifier (derived data type)
OVTRL	Signal name or mnemonic

6.15.1 Axes Hardware Limits (SPS⇒CNC)

The control of limit switches (E-Stop) and overtemperature switches has been implemented into the CNC. The signals are related in the CNC to the process control signals to generate required and correct diagnostic messages at any time.

NOTE: *Transfer all axis control signals to the CNC without any further logic combinations in the SPS to assure correct diagnostics.*

AxxC.OTRVL Axis E-STOP Overtravel Switch

“0”= Switch open, axis overtravel (E-STOP)

“1”= Switch closed, no axis overtravel

The opening of this switch (AxxC.OVTRL=“0”) will result in a overtravel condition. The message for the axis with the highest priority (lowest axis number) will be displayed in the MUI if more than one overtravel switch is actuated. The signal status ‘1’ must be assigned to the axis signal if there is no switch wired to the SPS input.

AxxC.MTAS Axis Motor Overtemperature

“0”= Axis motor overtemperature

“1”= Axis motor temperature OK

This signal affects the process differently depending on the active process mode.

If process power is OFF:

Process power cannot be switched ON if one of the process axis motors has an overtemperature fault.

If process power is ON, but no part program is active:

The process power is switched OFF immediately if one of the process axis motors has an overtemperature fault.

Process part program active:

The signal “Axis Motor Overtemperature” is recognized by the CNC. The station power will be switched OFF after cycle complete if the cycle is finished within 60 seconds after the overtemperature occurred. The CNC will perform an immediate process stop and switch OFF the process power if the cycle exceeds 60 seconds. There is no error message displayed and the process power is not switched OFF if the axis motor overtemperature disappears within the 60 seconds.

NOTE: *The status of “1” must be assigned to AxxC.MTAS if there is no motor overtemperature signal available for the axis.*

6.15.2 Axis Control Signals (SPS⇒CNC)

The Axis Control Signals are the signals from the SPS ⇒CNC that are required to control the axis. Some of these signals are only active in certain operating modes.

AxxC.READY Axis Ready to Operate

“0”= Axis is not ready to operate

“1”= Axis is ready to operate

This signal is valid in all operating modes. The status “1” of this signal indicates to the CNC that the axis is ready to operate. The CNC outputs the signal AxxS.RF=“1” (Axis Regulator Release) back to the SPS if PxxS.POWER=“1” (Process Power) and AxxC.READY=“1” (Axis Ready to Operate).

NOTE: *Further conditions may be necessary to influence the regulator release (AxxS.RF) of a drive. These conditions must be combined in the SPS program with the (AxxC.READY) signal.*

AxxC.ENABL Axis Enable

“0”= Axis disabled

“1”= The SPS enables the axis

This signal is only valid and active in Manual Operating Mode. Manual axis movements are not possible if (AxxC.ENABL=“0”). The axis stops immediately if the status of AxxC.ENABL switches to “0” while the axis is moving.

NOTE: *The status of AxxC.ENABL must be set to “1” to allow axis jog movements in manual mode.*

AxxC.HOMLS Axis Home Limit Switch

“0”= Home limit switch not actuated

“1”= Home limit switch actuated

This signal is used to search for the axis reference position of incremental analog drive systems. The reference position is determined by the switch (AxxC.HOMLS) together with the zero marker pulse of the incremental measuring system.

NOTE: *The signal AxxC.HOMLS is not necessary if an absolute measuring system is used. In the case of digital servo drives (DDS), the home switch should be wired directly to the DDS so the signal is independent of the SPS program scan.*

AxxC.STRBP Strobe Axis Position

“0⇒1”= Store axis position with positive edge

“1⇒0”= Store axis position with negative edge

This signal allows to store two axis positions in non-retentive APR (Axis Processor) registers for each axis. The following NC-commands can be used to read the strobed axis position in the NC-program.

- PMP: Reads the axis position that was stored with the AxxC.STRBP signal change from “0⇒1”.
- NMP: Reads the axis position that was stored with the AxxC.STRBP signal change from “1⇒0”.

AxxC.HOME Single Axis Home

This signal is only active in manual mode and is not necessary when using an absolute measuring system.

Table 6-15: Requirements for Single Axis Home

SIGNAL	STATUS	DESCRIPTION
PxxC.MODE1	1	Manual operating mode
PxxC.MODE0	0	
AxxC.ENABL	1	Axis enabled
AxxS.RF	1	Axis regulator release
AxxC.READY	1	Axis ready
PxxC.ENABL	1	Process enabled
PxxS.POWER	1	Process power active
PxxS.ERROR	0	No process error active
PxxS.RUN	0	No NC-block active
PxxS.STOP	0	No process stop active
AxxC.JGPOS	0	No jog positive
AxxC.JGNEG	0	No jog negative

The signal AxxS.HOMED is set to “1” if the axis has successfully completed its homing.

NOTE: *AxxC.HOME must remain a status of “1” during the homing sequence. If AxxC.HOME switches to “0” the homing sequence will be interrupted.*

NOTE: *The reverse program still needs to be executed even if all process axes have been homed manually, to achieve a ready for start signal (PxxS.READY=“1”).*

AxxC.JGPOS Jog Axis in Positive Direction

“0”= Jog axis positive command is not active

“1”= Jog axis manually in positive direction

The type of movement depends on the active Jog mode (parametric, incremental, or continuous). The axis speed, specified in the axis parameters, may be influenced via the feedrate override and rapid traverse mode.

Table 6-16: Requirements for Manual Axis Jogging in Positive Direction

SIGNAL	STATUS	DESCRIPTION
PxxC.MODE1	1	Manual operating mode
PxxC.MODE0	0	
AxxC.ENABL	1	Axis enabled
AxxS.RF	1	Axis regulator release
AxxC.READY	1	Axis ready
PxxC.ENABL	1	Process enabled
PxxS.POWER	1	Process power active
PxxS.ERROR	0	No process error active
PxxS.RUN	0	No NC-block active
PxxS.STOP	0	No process stop active
AxxC.HOME	0	No single axis home
AxxC.JGNEG	0	No jog negative

AxxC.JGNEG Jog Axis in Negative Direction

“0”= Jog axis negative command is not active

“1”= Jog axis manually in negative direction

The type of movement depends on the active Jog mode (parametric, incremental, or continuous). The axis speed, specified in the axis parameters, may be influenced via the feedrate override and rapid traverse mode.

Table 6-17: Requirements for Manual Axis Jogging in Negative Direction

SIGNAL	STATUS	DESCRIPTION
PxxC.MODE1	1	Manual operating mode
PxxC.MODE0	0	
AxxC.ENABL	1	Axis enabled
AxxS.RF	1	Axis regulator release
AxxC.READY	1	Axis ready
PxxC.ENABL	1	Process enabled
PxxS.POWER	1	Process power active
PxxS.ERROR	0	No process error active
PxxS.RUN	0	No NC-block active
PxxS.STOP	0	No process stop active
AxxC.HOME	0	No single axis home
AxxC.JGPOS	0	No jog positive

AxxC.OVRD Axis Override^{(2),(3),(4),(5)}

AxxC.OVRD = ‘0...255’ (16#00...FF): Speed/feed override in %.

This gateway signal control Byte for all NON-interpolating axes such as main spindle and tool storage (turret, magazine) axes. AxxC.OVRD must be used to modify the feed or speed, depending on the type of axis.

The override is processed by the CNC independent of the active operating mode. The override is active at all process axis movements, except during homing of digital servo drives (DDS)

The range of the axis override is 0...255 (decimal Unsigned Short Integer) which is equal to 0...255% in the CNC. The axis velocity is limited by the CNC to its maximum velocity (parameters) if the override would exceed the maximum (parameter) axis velocity.

NOTE: *In version 14VRS and above only the main axes are influenced via PxxC.FOVRD. The corresponding axis signal AxxC.OVRD must be used to control the feed override of NON-interpolating axes (e.g., tool storage axis).*

AxxC.QDDS SERCOS Real Time Control Bit #2

“0”= The status of the allocated bit in SERCOS parameter S-0-0303 assumes the status of “0”.

“1”= The status of the allocated bit in SERCOS parameter S-0-0303 assumes the status of “1”.

Indramat’s digital servo and spindle drives use the SERCOS interface when connected to the MT-CNC. SERCOS provides two real time control bits to transfer Boolean information every 2ms from the MT-CNC to the drive. The axis Gateway control signal AxxC.QDDS can be used in the SPS to influence the status of the drive bit that is allocated to the real time control bit #2 (drive parameter S-0-0303).

NOTE: *The parameter S-0-0303 is not buffered in most digital drives (e.g., DDS, TDA, RAC, KDA) and will be lost during drive power off.*

AxxC.GEAR1 Actual Spindle Gear Selection^{(2),(3),(4),(5)}**AxxC.GEAR2****AxxC.GEAR3****Table 6-18: Actual Spindle Gear Selection**

Actual gear range	AxxC.GEAR1	AxxC.GEAR2	AxxC.GEAR3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
not defined	1	1	1

The axis Gateway signals AxxC.GEAR(1...3) are available for each spindle axis and is used to inform the CNC about the current gear range selection. The CNC considers the status of these signals only if more than 2 gear ranges exist (see spindle gear range Axis Parameters).

AxxC.M3**Spindle on CW (M3)^{(2),(3),(4),(5)}**

“0”= No function

“1”= Start spindle clockwise

This signal is valid in manual mode only. A status change from “0” to “1” can be used in the SPS program to start the spindle in a clockwise rotation. The value of the axis parameter Cxx.020 is used as the speed command value.

Table 6-19: Requirements for AxxC.M3

SIGNAL	STATUS	DESCRIPTION
PxxC.MODE1	1	Manual operating mode
PxxC.MODE0	0	
AxxC.ENABL	1	Axis enabled
AxxS.RF	1	Axis regulator release
AxxC.READY	1	Axis ready
AxxC.MHOLD	0	No motion hold
AxxC.M4	0	No spindle ccw command
AxxC.M5	0	No spindle stop command
AxxC.M19	0	No spindle stop command

AxxC.M3 must not maintain a status of “1” to keep the spindle running. The spindle will run until a signal change “0 ⇒1” occurs at AxxC.M4, AxxC.M5, AxxC.M19 or if the corresponding aux. NC-function Mx03, Mx04, Mx05, or Mx19 (x=, ,1,2,3) is executed or the drive power is switched off.

AxxC.M4 Spindle on CCW (M4)^{(2),(3),(4),(5)}

“0”= No function

“1”= Start spindle counterclockwise

This signal is valid in manual mode only. A status change from “0” to “1” can be used in the SPS program to start the spindle in a counterclockwise rotation. The value of the axis parameter Cxx.020 is used as the speed command value.

Table 6-20: Requirements for AxxC.M3

SIGNAL	STATUS	DESCRIPTION
PxxC.MODE1	1	Manual operating mode
PxxC.MODE0	0	
AxxC.ENABL	1	Axis enabled
AxxS.RF	1	Axis regulator release
AxxC.READY	1	Axis ready
AxxC.MHOLD	0	No motion hold
AxxC.M3	0	No spindle cw command
AxxC.M5	0	No spindle stop command
AxxC.M19	0	No spindle stop command

AxxC.M4 must not maintain a status of “1” to keep the spindle running. The spindle will run until a signal change “0 ⇒1” occurs at AxxC.M3, AxxC.M5, AxxC.M19 or if the corresponding aux. NC-function Mx03, Mx04, Mx05, or Mx19 (x=, ,1,2,3) is executed or the drive power is switched off.

AxxC M5 Spindle Stop (M5)^{(2),(3),(4),(5)}

“0”= No function

“1”= Stop spindle axis

This signal is valid in manual mode only. A status change from “0” to “1” can be used in the SPS program to stop the spindle. AxxC.M5 must not maintain a status of “1” to keep the spindle stopped. The spindle will remained stopped until a signal change “0 ⇒1” occurs at AxxC.M3, AxxC.M4, AxxC.M19 or if the corresponding aux. NC-function Mx03, Mx04, or Mx19 (x=, ,1,2,3) is executed.

AxxC.M19 Orient (Position) Spindle (M19)^{(2),(3),(4),(5)}

“0”= No function

“1”= Orient spindle axis

A status change from “0” to “1” can be used in the SPS program to orient the spindle to the position defined in axis parameter Cxx.059/Cxx.059 (Parametric Spindle Jogging Distance). AxxC.M19 must not maintain a status of “1” while the spindle positions. The spindle remains at its commanded position until a signal change “0 ⇒1” occurs at AxxC.M3 or AxxC.M4 or if the corresponding aux. NC-function Mx03, Mx04, or Mx19 (x=, ,1,2,3) is executed or if the drive power is switched off.

AxxC.MHOLD Axis Motion Hold^{(2),(3),(4),(5)}

“0”= Axis motion allowed

“1”= Axis motion on hold (not allowed)

The AxxC.MHOLD signal reacts differently between interpolated/coupled axes and non-interpolating axes:

Non-interpolating axes:

Axis motion is on hold (not allowed) if (AxxC.MHOLD = ‘1’) before the motion is commanded.

Motion is allowed and occurs once AxxC.MHOLD switches to ‘0’. Ongoing motion of an axis is interrupted if AxxC.MHOLD switches to ‘1’.

NOTE: *Because of signal transfer time delays (SPS ⇔ CNC), axis motion may be started if the status of AxxC.MHOLD is set to ‘1’ upon the motion command signals. However, motion will stop immediately once the status ‘1’ of AxxC.MHOLD is recognized by the CNC.*

Interpolating/coupled axes:

Interpolating axes or axes participating in the coupling are stopped with maintaining the programmed path. The motion is continued if the SPS program sets the status of AxxC.MHOLD to ‘0’.

NOTE: *It is not allowed to program an axis whose motion is on hold (AxxC.MHOLD=‘1’) while G8 ‘Contouring Mode, Acceleration’ is active.*

AxxC.N_CMD Spindle at Commanded Speed^{(2),(3),(4),(5)} (N=CMD)

“0”= Spindle at commanded speed

“1”= Spindle NOT at commanded speed

The spindle command value (S-word) is output to the drive at the beginning of NC-block execution. Feed axis movement programmed in the same NC-block is allowed by the CNC after the **commanded** speed or position is achieved (depends on acceleration axis parameters). This can cause problems if the commanded spindle acceleration, speed or position can NOT be achieved by the axis drive. The CNC would then allow feed axis motion even though the actual speed/position differs from the commanded speed/position. To prevent such problems, AxxC.N_CMD and the corresponding status signals of the spindle drive must be used to allow feed axis movement only if (AxxC.N_CMD = ‘0’) the actual speed/position is maintained (within the window parameter settings) by the drive.

NOTE: *In version 14VRS and above the signal AxxC.N_CMD replaces the signal PxxC.N_CMD.*

AxxC.RAPID Rapid Jog Spindle Speed^{(4),(5)}

“0”= Jog spindle without using the rapid speed

“1”= Jog spindle using rapid speed

The rapid speed defined in the corresponding setting registers of the Machine Data is used to jog the spindle while the signal AxxC.RAPID has the state of ‘1’ and the Manual Operating Mode is active.

NOTE: *The signal AxxC.RAPID is valid for spindle axis only.*

AxxC.SPSTP Spindle Stop Upon Process Stop^{(2),(3),(4),(5)}

“0”= Process stop without spindle stop

“1”= Spindle stop with process stop

A running spindle is stopped upon a process stop if (AxxC.SPSTP = '1'). The spindle restarts when the NC program is restarted. Once the command speed is reached the NC-program is continued and other axes start moving. The spindle will not be stopped if (AxxC.SPSTP = '1') and there is no process stop. If a process stop is active when the spindle stop signal is set, the spindle will not be stopped.

NOTE: *The signal AxxC.SPSTP is valid for spindle axis only.*

14VRS and above:

NOTE: AxxC.SPSTP replaced PxxC.SPSTP.

AxxC.SPHLT Spindle Stop with Stopped NC Program^{(2),(3),(4),(5)}

“0”= No spindle stop with stopped NC-program

“1”= Spindle stop with stopped NC-program

This signal stops a running spindle after the NC-program has been interrupted. The spindle starts first when the NC-program is restarted. Once command speed has been reached, the NC-program is continued and other axes start moving.

NOTE: *This signal is only valid for spindle axes.*

NOTE: *In version 14VRS and above AxxC.SPHLT replaced PxxC.SPHLT.*

AxxC.SPSTE Spindle Stop at Program End^{(2),(3),(4),(5)}

“0”= No spindle stop at program end

“1”= Spindle stop at program end

If this signal is set to '0', a 'BST', 'RET', 'M30' or 'M02' in the NC-program will not stop a running spindle. The spindle will only be stopped if the signal is set to '1' before the program end is reached. The stopped spindle will not be started when the NC-program is started again.

NOTE: *This signal is only valid for spindle axes.*

AxxC.SPRST Spindle Stop Upon Control-Reset^{(2),(3),(4),(5)}

“0”= Control-reset stops the spindle

“1”= Control-reset does not influence the spindle

If this signal is set to '1', Control-reset will not stop a running spindle. If this signal is set to '0', Control-reset will stop a running spindle.

6.15.3 Axis Status Signals (CNC⇒SPS)

The Axis Signals include all signals transferred from the CNC to the SPS that provide information about the operating status of the axis.

AxxS.RF Axis Regulator Release

“0”= Axis regulator not release

“1”= Axis regulator released

The function of this signal changes depending on type of drive being used:

- **Analog Axis**

The signal AxxS.RF must be transferred in the SPS program to the corresponding output to which the drive RF-signal is wired to. There should not be any other SPS signals incorporated with this signal that could cause delays.

- **Digital Axis**

SERCOS transfers the regulator release signal direct to the digital drive using the AxxC.READY signal. In this case the AxxS.RF signal can be used as status information.

AxxS.BBDIG Digital Drive Ready

“0”= Digital drive is not ready to operate

“1”= Digital drive is ready to operate

A status of “1” for this signal indicates that the digital drive is ready for main power.

AxxS.HOMED Axis has been Homed

“0”= The axis is not homed

“1”= The axis has been homed

The SPS recognizes with this signal whether the axis with incremental position feedback has been homed since switching power on. Axes with incremental position feedbacks must be homed after every MT-CNC control power ON. The axis can be homed via the CNC (G74) or single axis home. Complete reverse program execution is necessary to establish “Ready for Start” condition (PxxS.READY=“1”).

AxxS.MVPOS Axis Movement in Positive Direction

“0”= No axis movement in the positive direction

“1”= Axis movement in the positive direction

This signal does not indicate feedback device movements of a disabled analog axis if the feedback device is moved manually. Indramat’s digital drives however, indicate with this signal an axis movement of the disabled axis (AxxS.RF=“0”) in the positive direction.

AxxS.MVNEG Axis Movement in Negative Direction

“0”= No axis movement in the negative direction

“1”= Axis movement in the negative direction

This signal does not indicate feedback device movements of a disabled analog axis if the feedback device is moved manually. Indramat’s digital drives however, indicate with this signal an axis movement of the disabled axis (AxxS.RF=“0”) in the negative direction.

- AxxS.WPn Way Point Signals**
 “0”= Actual axis position \leq specified axis parameter
 “1”= Actual axis position $>$ specified axis parameter
 n = Waypoints (0...7) corresponding to axis parameters
 (Cxx.025...Cxx.032).
 The signal AxxS.WPn has a status of “1” if the current axis position is greater than (more positive) the position that is specified in the corresponding axis waypoints parameter. The CNC generates the axis waypoints to the SPS whether the axis has been homed or not. This may require an interlock with the AxxS.HOMED signal in the SPS program to assure correct status of axis waypoints.
- AxxS.IDDS Real Time Status Bit #2**
 “0”= The status of the allocated bit in SERCOS parameter S-0-0307 is “0”.
 “1”= The status of the allocated bit in SERCOS parameter S-0-0307 is “1”.
 Indramat’s digital servo and spindle drives use the SERCOS interface when connected to the MT-CNC. SERCOS provides two real time control bits to transfer Boolean information every 2ms from the MT-CNC to the drive. The axis Gateway control signal AxxS.IDDS can be used in the SPS to read the status of the drive bit that is allocated to the real time control bit #2 (drive parameter S-0-0307).
- NOTE:** *The parameter S-0-0307 is not buffered in most digital drives (e.g., DDS, TDA, RAC, KDA) and will be lost during drive power off.*
- AxxS.LD90 Actual Spindle Load $\geq 90\%$ ^{(2),(3),(4),(5)}**
 “0”= Load is equal to or greater than 90% of the drive’s value contained in ID number S-0-0092
 “1”= Load is less than or greater than 90% of the drive’s value contained in ID number S-0-0092
 When using Indramat’s digital spindle drives:
 AxxS.LD90 reflects the status of the manufacture specific SERCOS ID number P-0-1307 (Message 90% Load) and is transferred every 2ms to the SPS of the MT-CNC.
- AxxS.SYNC Spindle is Running Synchronous** ^{(2),(3),(4),(5)}
 “0”= Deviation greater than synchronized window position (S-0-0228)
 “1”= Within synchronized window position (S-0-0228)
 The status of this signal is set when the difference between the synchronous position command value rep. velocity command value of the position feedback value resp. velocity feedback value of the synchronization spindle is within the value range of the ‘Synchronization Window Position’ (S-0-0228) with respect to the ‘Synchronization Window Velocity’ (S-0-0183).
 Refer to the digital (SERCOS) drive manual and description of ‘Synchronization Message’ (S-0-0308) for information on whether such a status bit exists.
- AxxS.N_CMD Actual Spindle Speed = Commanded Spindle Speed (N=CMD)** ^{(2),(3),(4),(5)}
 “0”= Spindle axis is not running at the commanded speed
 “1”= Spindle is running at the commanded speed
 The signal AxxS.N_CMD reflects the status of SERCOS ID number S-0-0330 (Message “nactual=ncommand”), bit 0. ID number S-0-0157 “Velocity Window” can be used to adjust a window around the commanded speed. Bit 0 of S-0-0330 and AxxS.N_CMD will have a status of “1” if the actual speed is equal to or greater than the commanded speed plus/minus the window value.
- NOTE:** *If AxxC.N_CMD is not used, motion is stopped only during acceleration and deceleration based on the command value. Steep command ramps which do not allow the drive to follow in time may allow the CNC to command axis motion at incorrect spindle speeds. Therefore, the SPS must contain logic to monitor the drive’s N=CMD and INPOS signals at all times.*
- AxxS.N_MIN Actual Spindle Speed $<$ Minimum Spindle Speed (N<MIN)** ^{(2),(3),(4),(5)}
 “0”= Actual spindle speed is greater than the specified minimum spindle speed

“1”= Actual speed is less than the specified minimum spindle speed
 The signal AxxS.N_MIN reflects the status of SERCOS ID number S-0-0331 (Message “nactual=0”) bit 0. The ID number S-0-0124 (Zero Velocity Window) can be used to specify the drive’s minimum speed.

AxxS.MD_DX Actual Torque \leq Torque Threshold ($T \geq T_x$)^{(2),(3),(4),(5)}
 “0”= Actual torque of spindle axis is less than or equal to the specified (Torque Threshold T_x) value in ID number S-x-0126 ($x = 0 \dots 7$)
 “1”= Actual torque of spindle axis is greater than the specified (Torque Threshold T_x) value in ID number S-x-0126 ($x = 0 \dots 7$)
 The signal AxxS.MD_DX reflects the status of SERCOS ID number S-0-0330 (Message “ $T \geq T_x$ ”) bit 0.

AxxS.N_MAX Actual Spindle Speed \geq Maximum Spindle Speed^{(2),(3),(4),(5)}
 (N=MAX)
 “0”= Actual velocity of the spindle axis is less than the specified value in ID number S-x-0091 (Bipolar Velocity Limit Value)
 “1”= Actual velocity of the spindle axis is greater than or equal to the specified value in ID number S-x-0091 (Bipolar Velocity Limit Value)
 The signal status AxxS.N_MAX reflects the status of SERCOS ID number S-0-0335 Message ‘ncomand > nlimit’, bit 0.

AxxS.INPOS Spindle in Position (INPOS)^{(2),(3),(4),(5)}
 “0”= Actual spindle axis position is not within the specified value in ID number S-0-0057 (Position Window) relative to the value in ID number S-0-0047 (Position Command Value)
 “1”= Actual spindle axis position is within the specified value in ID number S-0-0057 (Position Window) relative to the value in ID number S-0-0047 (Position Command Value).
 The signal status AxxC.INPOS reflects the status of SERCOS ID number S-0-0336 Message ‘In Position’, bit 0.

AxxS.P_PX Actual Spindle Power \geq Power Threshold ($P \geq P_x$)^{(2),(3),(4),(5)}
 “0”= Actual supplied power of spindle axis exceeds the specified value in ID number S-0-0158 (Power Threshold P_x)
 “1”= Actual supplied power of spindle axis does not exceed the specified value in ID number S-0-0158 (Power Threshold P_x)
 The signal status AxxS.P_PX ($xx = 1 \dots 20$) reflects the status of SERCOS ID number S-0-0337 Message ‘ $P \geq P_x$ ’, bit 0.

AxxS.POWN Axis within Position Window^{(2),(3),(4),(5)}
 “0”= Axis is outside the defined position window
 “1”= Axis is within the defined position window
 This signal can be used in the SPS program to check that the difference between the accumulated actual axis position value and the position command value is within the range of the position window (Cxx.023), (Cxx.052) or (S-0-0057).

NOTE: *When jogging the axis in continuous jog mode, AxxS.POWN has the status of ‘0’. The signal is issued when using incremental jogging. The signal also is not issued after a spindle stop via Mx05.*

AxxS.MCPOS Motion Command, Positive^{(2),(3),(4),(5)}
 “0”= No positive motion commanded for the axis
 “1”= Positive motion commanded for the axis
 AxxS.MCPOS has a status of ‘1’ if a positive motion or movement is commanded for the axis. Movement interruptions caused by a stop, reset the status of AxxS.MCPOS to ‘0’.

NOTE: The status of both, *AxxSMCPOS* and *AxxS.MCNEG* are set to '1' during axis homing, spindle positioning, circular interpolation and hand wheel mode.

AxxS.MCNEG Motion Command, Negative^{(2),(3),(4),(5)}

“0”= No negative motion commanded for the axis

“1”= Negative motion commanded for the axis

AxxS.MCNEG has a status of '1' if a negative motion or movement is commanded for the axis.

Movement interruptions caused by a stop, reset the status of *AxxS.MCNEG* to '0'.

NOTE: The status of both, *AxxSMCPOS* and *AxxS.MCNEG* are set to '1' during axis homing, spindle positioning, circular interpolation and hand wheel mode.

6.16 MT-CNC Specific APR parameters

This section describes all MT-CNC (APR) drive parameters that are available at the time of printing. The data of these ID numbers / digital drive parameters is accessed via the APR service channel using the NC-command AXD and FUNCTION_BLOCKS AXD_RD and AXD_WR.

Table 6-21: APR parameter range defined for the MT-CNC

General drive parameter structure used in NC-command AXD	Binary ID number structure	INTEGER ID number used with SPS FUNCTION_BLOCKS AXD_RD and AXD_WR
P-7-3560	1111 1101 1110 1000	65000
:	:	:
P-7-4096	1111 1111 1111 1111	65535

6.16.1 SCARA Robots

65000/P-7-3560 SCARA robot, Arm Bending Preference, Control

(read/write), (drive: X axis)

“0”= Arm bends to the left

“1”= Arm bends to the right

This APR parameter can be used to select the arm bending preference of a SCARA robot.

65001/P-7-3561 Interpolation Mode of SCARA Robot

(read/write), (drive: X axis)

“0”= Cartesian interpolation

“1”= Point to point operation with filter on drive

“2”= Point to point operation without filter

This APR parameter can be used to select the arm bending preference of a SCARA robot.

65002/P-7-3562 Velocity of SCARA Robot in Point to Point Operation

(read/write), (drive: X axis)

100% corresponds to the maximum possible velocity.

65003/P-7-3563 Interrogation of the SCARA Robot's Current Arm Position

(read only), (drive: X axis)

“0”= Left armed

“1”= Right armed

65004/P-7-3564 Interrogation of the SCARA Robot's Selected Mode

(read only), (drive: X axis)

“0”= Cartesian interpolation

“1”= Point to point operation with filter on drive

“2”= Point to point operation without filter

65006P-7-3566 Activation of SCARA lift/rotary axis

(read/write), (drive: X axis)

Bit 0:

“0”= Z axis and manual axis are mechanically separated

“1”= Lift/rotary axis

Bit 1:

“0”= Pos. direction of correction when lift/rotary axis is used

“1”= Neg. direction of correction when lift/rotary axis is used.

65007/P-7-3567 Length of SCARA Gripper Arm

(read/write), (drive: X axis)

Program in (0.1µm) increments.

65019/P-7-3579 SCARA Arm Length

(read/write)

The APR takes the SCARA arm length from the drives when the DDS is used. The arms lengths are defined in the SERCOS parameters P-0-0102. If one of the drives does not support the P-0-0102 parameter, the related arm length can be selected via the ident number 65019. The drives have the selected lengths assigned:

- X axis (1st axis APR) 1st SCARA arm.
- Y axis (2nd axis APR) 2nd SCARA arm
- Z axis (3rd axis APR) SCARA gripper arm
- W axis (4th axis APR) SCARA gripper arm

The parameter values of the SCARA gripper arm may also be selected via the ident number 65007.

65020/P-7-3580 Speed of SCARA Drive in Point to Point Operation

(read/write)

X, Y and W axis: Speed in 0.0001 rpm

Z axis: Velocity in 0.001 mm/min

65021/P-7-3581 Acceleration of SCARA Drive in Point to Point Operation

(read/write)

X, Y and W axis: Acceleration in 0.001 rad/s²

Z axis: Acceleration in 0.001 mm/s²

65039/P-7-3599 Referenced Path Velocity for Sealant Application Using SCARA Robots

(read/write), (drive: X axis)

The specified spindle speed is output to the pump motor when the Cartesian axes move at referenced path velocity. This parameter can be programmed in 1 mm/min increments.

$$VBahn = \sqrt{V_x^2 + V_y^2 + V_z^2}$$

$$1) \quad V_{Bahn} < V_{Bahnreferenz}:$$

$$NPumpe = SWert \times \left(\frac{VBahn}{VBahnreferenz} \right) \times \left(\frac{SOverride}{100} \right)$$

$$2) \quad V_{Bahn} \geq V_{Bahnreferenz}$$

$$NPumpe = SWert \times \left(\frac{SOverride}{100} \right)$$

N_{Pumpe}	Speed of Pump motor
V_{Bahn}	Current path velocity
$V_{Bahnreferenz}$	Reference path velocity
V_x	Velocity in X direction
V_y	Velocity in Y direction
V_z	Velocity in Z direction
$SWert$	Programmed speed value
$SOverride$	Spindle override

6.16.2 DDS as Spindle Drive**65008/P-7-3568 Spindle Drive “DDS Spindle Mode”**

(read/write)

“0”= No DDS used as main spindle drive

“1”= DDS used as main spindle

This parameter can be used to define an Indramat digital servo drive (DDS) as a main spindle. Once defined the DDS (main spindle) is capable of positioning, tapping, and feed per revolution. The use of the following NC-commands can also used, (M3, M4, M5, M19, G63, G64, G65, G95)

65009/P-7-3569 Reference Angle Position (DDS as Main Spindle)

(read/write)

This parameter offset can be used to shift the reference (0°) position if the DDS is defined as a main spindle drive. It is interpreted as offset in M19 S0 and is only active if ident number 65008 is set to ‘1’. This parameter can be programmed in 0.0001° increments.

6.16.3 Gear Change

65012/P-7-3572 Spindle Function During Gear Change
 (read/write)

Table 6-22: Spindle Function During Gear Change

BIT	ACTION	DEFINITION
0	0	No function
	1	Command spindle oscillation
1	0	No function
	1	Command spindle positioning
2	0	No function
	1	Command spindle speed, speed defined in 65013 / P-7-3573, acceleration defined in 65014 / P-7-3574
8	0	No function
	1	Acknowledge spindle oscillating
9	0	No function
	1	Acknowledge spindle positioning
10	0	No function
	1	Acknowledge commanded spindle speed

NOTE: This feature requires the digital main spindle drive software version to be greater than HASE 0V0.9.

65013/P-7-3573 Spindle Speed for Spindle Function ‘Cmd. Spindle Speed’
 (read/write)

The spindle speed value can be modified during activation of the spindle function “Command Spindle Speed” in increments of 0.0001 1/min to a maximum value of 1000 1/min.

NOTE: This feature requires the digital main spindle drive software version to be greater than HASE 0V0.9.

65014/P-7-3574 Spindle Acceleration for the Spindle Function ‘Command Spindle Speed’
 (read/write)

The spindle acceleration can be modified during activation of the spindle function “Command Spindle Speed”. This value is programmed in rad/S².

NOTE: This feature requires the digital main spindle drive software version to be greater than HASE 0V0.9.

65015/P-7-3575 Spindle Position for the Spindle Function ‘Command Spindle Positioning’
 (read/write)

The spindle position can be modified during activation of the spindle function “Command Spindle Positioning”. This parameter can be programmed in 0.0001° increments.

NOTE: This feature requires the digital main spindle drive software version to be greater than HASE 0V0.9.

6.16.4 Torque Reduction

65016/P-7-3576 Reduced Motor Torque of the Digital Drive

(read/write)

This feature allows the limit of bipolar (maximum) motor torque. A setting of 25% will limit the motor torque to the motor standstill (100% duty cycle) torque if the bipolar torque is set to 400% by the control. This parameter is programmed in percent.

65017/P-7-3577 Reduced Torque of the Digital Drive During Feed to Positive Stop

(read/write)

This feature allows the limit of bipolar (maximum) motor torque while searching for a positive stop (G75). A setting of 25% will limit the motor torque to the motor standstill (100% duty cycle) torque if the bipolar torque is set to 400% by the control. This parameter is programmed in percent.

65018/P-7-3578 Reduced Torque of the Digital Drive at Positive Stop

(read/write)

This feature allows the limit of bipolar (maximum) motor torque at the positive stop (G75). This value is active if it is less than the value of axis parameter Cxx.044 (Reduced Torque at Positive Stop) and is less than 25%. A setting of 10% will limit the motor torque to 40% of the motor standstill (100% duty cycle) torque if the bipolar torque is set to 400% by the control. This parameter is programmed in percent.

6.16.5 Filter

65010/P-7-3570 Filter Depth for Jerk Limitation Filter

(read/write)

“0”= Jerk limitation de-activated

“1...512”= Filter depth, jerk limitation activated

Prior to starting a motion block, the same filter depth (AXD command) must be transferred to all axes of a process. Time constant for acceleration ramp = (Filter Depth) x (2ms). The MT-CNC parameter Bxx.034 should be used instead of this parameter.

65022/P-7-358 Filter for the Command Position Values of the Specified Axis

(read/write)

This filter is automatically activated in handwheel mode. This command filter delimits velocity and acceleration of the commanded position values that are fed to the filter input. An additional jerk limitation may also be activated.

Table 6-23: Filter for the Command Position Values of the Specified Axis

BIT	ACTION	DEFINITION
0	0	Filter is inactive
	1	Filter is active
1	0	Filter 1 selected. Filter output position = Filter input position, if maximum velocity and acceleration are not exceeded.
	1	Filter 2 selected Filter with delay. At a constant velocity, the filter output position lags behind the filter input position.

65023/P-7-3583 Filter Depth for Jerk Limitation on Activation of Command Filter

(read/write)

“0”= Jerk limitation de-activated

“1...512”= Filter depth, jerk limitation active

This ID number allows to activate “Jerk” limitation for the axis if the ID number 65022 is set to “1”.

The filter depth value is accepted only while ID number 65022 is set to “0” and the path “Jerk” limitation Bxx.034 (Acceleration Time

Constant) is set to “0” ms, unless handwheel mode is activated (in manual operating mode) for the axis via the FUNCTION HNDWHELL. The acceleration time constant for the axis is determined as follows: (Filter Depth) * (2ms) = (Axis Accel. Time Constant)

NOTE: *This function is only available in MT-CNC software releases 04.13/xx and up.*

65024/P-7-3584 Acceleration Override for Command Filter

(read/write)

This ID number can be used as an override for the acceleration if the position command smoothing filter (ID number 65022=“1”) is active. A value of 100% equals the “Maximum Acceleration Rate” (Cxx.018) of the axis. The acceleration override value is accepted only while ID number 65022 is set to “0”.

65050/P-7-3610 Smoothing Time Constant for Second Feedback System^{(3),(4),(5)}

(read/write)

This time constant determines the coupling of the second feedback system with the motor feedback.

6.16.6 Synchronous Axis

65026/P-7-3586 Spindle Synchronization Mode^{(2),(3),(4),(5)}

(read/write)

Spindles can be synchronized only if they are connected in the same SERCOS fiber optic loop (APR). The first spindle address (address #1) is the master spindle, all remaining spindles are slave follower spindles. Spindle Synchronization mode can be activated only if velocity mode is active for the spindles. The mode must be selected first for the master spindle and then for the slave spindles. Correct values must be assigned to the parameters (65027...65030) before selecting the spindle synchronization mode. NC-blocks with M19 are correctly executed only if the spindle axes have been referenced via a M19 prior to selecting the spindle synchronization mode.

NOTE: *Use Machine Parameters and the corresponding Gateway signals when using software release 0X.15/00 and later.*

65027/P-7-3587 Slave Spindle, Gear Ratio Input^{(2),(3),(4),(5)}

(read/write)

Defines the input revolutions of the slave spindle when spindle synchronization mode is active.

NOTE: *Use Machine Parameters and the corresponding Gateway signals when using software release 0X.15/00 and later.*

65028P-7-3588 Slave Spindle, Gear Ratio Output^{(2),(3),(4),(5)}
(read/write)

Defines the output revolutions of the slave spindle when spindle synchronization mode is active.

NOTE: Use Machine Parameters and the corresponding Gateway signals when using software release 0X.15/00 and later.

65029/P-7-3589 Offset angle of a Slave Spindle in Synchronous Mode^{(2),(3),(4),(5)}
(read/write)

The offset angle defines the angular offset between the master spindle and the slave spindle. The offset angle may be modified during synchronous mode. The new offset angle will then be approached on the shortest way possible. If, prior to selecting synchronous mode, the offset angle is set to '-1', the current angular offset between master and slave spindles will be maintained upon changeover.

NOTE: Use Machine Parameters and the corresponding Gateway signals when using software release 0X.15/00 and later.

65030/P-7-3590 Axis Number of Master Spindle^{(2),(3),(4),(5)}
(read/write)

The number defines the axis drive number of the master spindle.

NOTE: Use Machine Parameters and the corresponding Gateway signals when using software release 0X.15/00 and later.

65042/P-7-3602 P factor for Spindle Synchronization Compensation Controller^{(3),(4),(5)}
(read/write)

The parameter defines the angular offset per torque difference between the spindles. This parameter is programmed in increments of 0.000001°/°.

NOTE: Use Machine Parameters and the corresponding Gateway signals when using software release 0X.15/00 and later.

6.16.7 Measuring Function Torque / Lag Error

65035/P-7-3595 Activate Measuring Function^{(3),(4),(5)}

(read/write)

This function can be used to measure the torque mean value, torque maximum, torque peak and the torque distribution and following error. In addition a tool broken monitor (exceeding the maximum torque peak) can be activated.

Table 6-24: Activate Measuring Function

BIT	ACTION	DEFINITION
0	0	Torque measurement inactive
	1	Torque measurement active
1	0	Torque measurement without acceleration phase
	1	Torque measurement with acceleration phase
2	0	Measuring the average torque value
	1	Measuring the maximum torque value
3	0	Tool breakage monitoring function inactive
	1	Tool breakage monitoring function active
4	0	Torque distribution measurement inactive.
	1	Torque distribution measurement active
5	0	Lag error measurement inactive
	1	Lag error measurement active

65036/P-7-3596 Measured Torque^{(3),(4),(5)} **Value**

(read/write)

Measured motor current (100% = Motor current at standstill, continuous) with a resolution of 0.1%. The measurement of this data can be started via Bit 0 of APR parameter 65035 (P-7-3595), based on Bit 1 and Bit 2, with or without acceleration consideration and RMS can be selected. . This parameter can be programmed in increments of 0.1%.

65037/P-7-3597 Peak Current (Torque) Limit^{(3),(4),(5)}

(read/write)

Peak current limit (100% = motor current at standstill, continuous) for monitoring. An error is generated if the measured peak current (65038/P-7-3598) exceeds this peak current (torque) limit while the Current Monitor (Bit 3 of 65035/P-7-3595) is active. This parameter can be programmed in increments of 0.1%.

65038/P-7-3598 Measured Peak Current (Torque)^{(3),(4),(5)}

(read/write)

The measured peak current limit (100% = Motor current at standstill, continuous) for monitoring is measured only if the Current Monitor (Bit 3 of 65035/P-7-3595) is NOT active. This parameter can be programmed in increments of 0.1%.

65040/P-7-3600 Measured Current (Torque) Distribution^{(3),(4),(5)}
(read/write)

Measured current (torque) distribution with a resolution of 0.01%. The measurement is useful in milling application only, using main spindles and Bit 4 of 65035. The measurement requires cyclical (every 2ms, check software release if KDA, RAC, TDA is used) actual torque measurement. The current distribution can be used to check on the symmetrical mill cutter intervention into the material. The more symmetrical the mill cutter edge intervention with the material, the smaller the current distribution value becomes.

NOTE: *Current distribution measurement can be performed for one APR axis at a time.*

65041/P-7-3601 Measured Maximum Following Error (LAG)^{(3),(4),(5)}
(read/write)

Measured maximum following error (lag) in 0.0001 mm (linear axis) or 0.0001° (rotary axis). The measurement is started via Bit 5 of 65035.

NOTE: *Following error (lag) measurement can be performed for one APR axis at a time.*

6.16.8 Combined Spindle / Turret Axis**65032/P-7-3592 Reference Cam Shift for Combined Spindle/Turret Axis**^{(3),(4),(5)}
(read/write)

Software reference cam shift in 0.0000° of a complete revolution for a combined spindle axis. This parameter can be used to eliminate the need of mechanical home cam shift.

65043/P-7-3603 Number of Tool Positions for Combined Spindle / Turret Axis^{(3),(4),(5)}
(read/write)

Number of tool locations for the combined spindle / turret axis.

65044/P-7-3604 Homing Required for Combined Spindle / Turret Axis^{(3),(4),(5)}
(read/write)

Table 6-25: Homing Required for Combined Spindle / Turret Axis

BIT	ACTION	DEFINITION
0	0	Homing not necessary
	1	Homing necessary

6.16.9 General**65005/P-7-3565 Actual Drive Current / Torque**^{(1),(2),(3),(4),(5)}
(read)

The motor type plate contains the information to determine the nominal (standstill) motor torque. Nominal torque (Nm) = Torque constant (Nm/A) + Standstill motor current (A).

65011/P-7-3571 Spindle Parameter

(read/write)

Set Bit 0 to '1' to display the external position feedback data on the MUI, GUI or SOT. The feature works only if the digital main spindle drive software version is greater than HASEOV0.8A.

Table 6-26: Spindle Parameter

BIT	ACTION	DEFINITION
0	0	Motor feedback displayed
	1	External feedback displayed
1	0	Drive data telegram containing actual position value, actual velocity value and actual torque.
	1	Drive data telegram containing actual position value, actual velocity value, actual torque and signal status word.

65025/P-7-3585 Polygon Course Interpolation Mode Selection^{(2),(3),(4),(5)}

(read/write)

"0"= Polygon mode is inactive

"1"= Polygon mode is active

This parameter must be set or reset for all process axes in order to use polygon interpolation mode. The polygon interpolation mode is active only while G1, G2 and G3 are active. Also, the preparatory functions G8 and G62 must be active. Polygon interpolation mode allows the execution of polygon courses. G9 should be selected in the last NC-block of the polygon course and then the polygon interpolation mode should be reset.

65031/P-7-3591 Static Monitoring Window for MT-CNC Internal Drive Monitoring^{(2),(3),(4),(5)}

(read/write)

Linear axis :Monitoring window in 0.0001 mm

Rotary axis/spindle :Monitoring window in 0.0001°

This parameter is the static monitoring window for the MT-CNC internal drive monitoring. Normally, this parameter is being set to useful values by the MT-CNC. However, if this static monitoring is too sensitive it can be made less sensitive via this parameter.

65033/P-7-3593 Digital Axis Simulation Mode^{(3),(4),(5)}

(read/write)

The simulation mode allows the execution of NC-programs containing axis movements without the need of having digital drives connected to the control. The actual axis positions equal the corresponding commanded axis positions.

Table 6-27: Digital Axis Simulation Mode

BIT	ACTION	DEFINITION
0	0	Normal operating mode
	1	Simulation mode

65034/P-7-3594 Axis Droop Compensation ON/OFF^{(3),(4),(5)}
(read/write)

This parameter allows to switch Axis Droop Compensation OFF via the user NC or SPS program. The compensation can also be switched back on if the compensation is activated via axis parameter Cxx.077.

Table 6-28: Axis Droop Compensation ON/OFF

BIT	ACTION	DEFINITION
0	0	Axis droop compensation inactive
	1	Axis droop compensation active

65045/P-7-3605 Actual Axis Position^{(1),(2),(3),(4),(5)}, **Motor Feedback 1**
(read)

Linear axis :0.0001 mm

Rotary axis :0.0001°

This parameter provides the actual axis position as an INT value.

65046/P-7-3606 Actual Axis Position, External Feedback 2
(read)

Linear axis :0.0001 mm or 0.0001°

This parameter provides the actual axis position as an INT value.

NOTE: Rotary axis currently not supported.

65047/P-7-3607 Active Feedback System^{(1),(2),(3),(4),(5)}
(read)

This parameter indicates which feedback of the drive is active.

Table 6-29: Active Feedback System

BIT	ACTION	DEFINITION
0	0	Encoder system 1 active
	1	Encoder system 2 active

65048/P-7-3608 Positive Limit of Dynamic Limit Monitoring Function^{(3),(4),(5)}
(read/write)

Linear axis :0.0001mm

This parameter is for linear axis only.

65049/P-7-3609 Negative Limit of Dynamic Limit Monitoring Function^{(3),(4),(5)}
(read/write)

Linear axis :0.0001mm

This parameter is for linear axis only.

65050/P-7-3610 Smoothing Time Constant for 2nd Encoder System^{(4),(5)}
(read/write)

This time constant determines the coupling between the 2nd encoder system and the motor encoder. This value is programmed in milliseconds.

65051/P-7-3611 Commanded Axis Position^{(4),(5)}

(read)

This parameter provides the commanded axis position as an INTEGER with a resolution of 0.0001mm.

65052/P-7-3612 Spindle Speed Limit^{(4),(5)}

(read/write)

This speed limit applies only when the spindle is in velocity mode. The limit does not apply when the spindle runs in position mode when performing spindle orient (Mx19). During spindle orient the spindle can assume higher speeds for a short period of time, when the spindle was at zero speed.

65053/P-7-3613 Table Type for Value Table on the APR^{(4),(5)}

“0”= No table defined

“1”= Torque value table depending on the axis position (only while G75 is active)

“2”= Table interpolator. Additional position offset depending on the position of a reference axis.

NOTE: Values in the APR table are not shown in the position display.

65054/P-7-3514 Length of APR Value Table^{(4),(5)}

(read/write)

The length of APR value table. The transfer of the table values is initialized when transferring this parameter via the NC-command AXD or FUNCTION_BLOCK AXD_WR.

65055/P-7-3615 APR Table Value Transfer^{(4),(5)}

(read/write)

The APR table value transfer is initiated when transferring the length of the APR value table (65054). The number of table values specified in 65054 must be transferred, whereby the first value will be at the APR table start position (65057). The value scaling depends on the specific APR Table type (65053):

65053 = 1 :Torque value in 0.1%, (e.g., 65055 = 1235 is 123%)

65053 = 2 :Position value in 0.0001mm or 0.0001° (e.g., 65055 = 12345678 is 1234.5678 mm or °)

65056/P-7-3616 APR Table Value Multiplication Factor^{(4),(5)}

(read/write)

The table value multiplication factor has a resolution of 0.0001. If (65056 = 12345) and the multiplication factor is set to (0.0001), parameter (65056) would be 1.2345. The APR parameter 65056 also depends on the specified APR table type (65053):

65053 = 1 :All table values are multiplied with the specified multiplication factor.

65053 = 2 :The multiplication factor is used only if the table mode “multiplication” is selected.

65057/P-7-3617 APR Table Start Position^{(4),(5)}

(read/write)

The start position value is scaled as either 0.0001mm or 0.0001° (e.g., 65055 = 12345678 is 1234.5678 mm or °).

65058/P-7-3618 APR Table Position Distance^{(4),(5)}

(read/write)

The table position distance value is scaled as either 0.0001mm or 0.0001°, e.g., 65055 = 12345678 is 1234.5678 mm or °. The defined distance is incrementally added to the start position value (65057) defining as many points as values defined in the APR table length parameter 65054.

65061/P-7-3621 Activate APR Table^{(4),(5)}

(read/write)

“0”= Deactivate APR table

“1”= Activate APR table

This parameter is used to activate/deactivate the APR table for the specified axis.

65063/P-7-3623 Disable Path Velocity Monitor^{(3),(4),(5)}

(read/write)

“0”= Path velocity reduction enabled

“1”= Path velocity reduction disabled

Path velocity optimization in Polar-Coordinate Transformation function (G31) can be switched off via this AXD parameter. The default setting of path velocity can cause difficulties in high speed machining. Disabling this reduction with this parameter allows for maximum path velocity, but the user is responsible for staying within the velocity and acceleration limits.

65065P-7-3625 Dynamic Limit Response Time^{(3),(4),(5)}

(read/write)

This parameter allows to program the response time of the dynamic limit monitoring in ms. the higher the response time, the more effectiveness the dynamic limit monitoring has on the system.

65066/P-7-3626 Feed in Axis Reference Offset for G^{(3),(4),(5)}

(read/write)

Reference offset of feed axis at active polar coordinate transformation (G31) in 0.0001mm resolution.

6.17 Gateway Signal Listing (Structured Data Type)

6.17.1 Axis Status Signals (iAXIS)

Table 6-30: INDRAMAT_DATA_TYPE iAXIS declaration

(*AXIS STATUS GATEWAY SIGNALS*)				
	0x.12/xx - 0x.13/xx SOFTWARE VERSION	0x.14/xx and above SOFTWARE VERSION	Type	Description
A01S	%IW0.4	A#A01S or %IW0.10	iAXIS	(*AXIS 1 STATUS*)
A02S	%IW0.12	A#A02S or %IW0.30	iAXIS	(*AXIS 2 STATUS*)
A03S	%IW0.20	A#A03S or %IW0.50	iAXIS	(*AXIS 3 STATUS*)
A04S	%IW0.28	A#A04S or %IW0.70	iAXIS	(*AXIS 4 STATUS*)
A05S	%IW0.36	A#A05S or %IW0.90	iAXIS	(*AXIS 5 STATUS*)
A06S	%IW0.44	A#A06S or %IW0.110	iAXIS	(*AXIS 6 STATUS*)
A07S	%IW0.52	A#A07S or %IW0.130	iAXIS	(*AXIS 7 STATUS*)
A08S	%IW0.60	A#A08S or %IW0.150	iAXIS	(*AXIS 8 STATUS*)
A09S	%IW0.68	A#A09S or %IW0.170	iAXIS	(*AXIS 9 STATUS*)
A10S	%IW0.76	A#A10S or %IW0.190	iAXIS	(*AXIS 10 STATUS*)
A11S	%IW0.84	A#A11S or %IW0.210	iAXIS	(*AXIS 11 STATUS*)
A12S	%IW0.92	A#A12S or %IW0.230	iAXIS	(*AXIS 12 STATUS*)
A13S	%IW0.100	A#A13S or %IW0.250	iAXIS	(*AXIS 13 STATUS*)
A14S	%IW0.108	A#A14S or %IW0.270	iAXIS	(*AXIS 14 STATUS*)
A15S	%IW0.116	A#A15S or %IW0.290	iAXIS	(*AXIS 15 STATUS*)
A16S	%IW0.124	A#A16S or %IW0.310	iAXIS	(*AXIS 16 STATUS*)
A17S	%IW0.132	A#A17S or %IW0.330	iAXIS	(*AXIS 17 STATUS*)
A18S	%IW0.140	A#A18S or %IW0.350	iAXIS	(*AXIS 18 STATUS*)
A19S	%IW0.148	A#A19S or %IW0.370	iAXIS	(*AXIS 19 STATUS*)
A20S	%IW0.156	A#A20S or %IW0.390	iAXIS	(*AXIS 20 STATUS*)

6.17.2 Process Status Signals (iPROC)

Table 6-31: INDRAMAT_DATA_TYPE iPROC declaration

(*PROCESS STATUS GATEWAY SIGNALS*)			
	0x.12/xx - 0x.13/xx SOFTWARE VERSION	0x.14/xx and above SOFTWARE VERSION	Type Description
P00S	%IW0.172	A#P00S or %IW0.420	iPROC (*PROCESS 0 STATUS*)
P01S	%IW0.196	A#P01S or %IW0.460	iPROC (*PROCESS 1 STATUS*)
P02S	%IW0.220	A#P02S or %IW0.500	iPROC (*PROCESS 2 STATUS*)
P03S	%IW0.244	A#P03S or %IW0.540	iPROC (*PROCESS 3 STATUS*)
P04S	%IW0.268	A#P04S or %IW0.580	iPROC (*PROCESS 4 STATUS*)
P05S	%IW0.292	A#P05S or %IW0.620	iPROC (*PROCESS 5 STATUS*)
P06S	%IW0.316	A#P06S or %IW0.660	iPROC (*PROCESS 6 STATUS*)

6.17.3 Local Inputs Signals (iDEA)

Table 6-32: INDRAMAT_DATA_TYPE iDEA declaration

(*LOCAL MT-CNC INPUTS*)			
	0x.12/xx - 0x.13/xx SOFTWARE VERSION	0x.14/xx and above SOFTWARE VERSION	Type Description
DEA0_I	%IW0.532	A#iDEA0 or %IW0.914	iDEA (*DEAB LOCAL INPUTS DEA0*)
DEA1_I	%IW0.540	A#iDEA1 or %IW0.934	iDEA (*DEAB LOCAL INPUTS DEA1*)
DEA2_I	%IW0.548	A#iDEA2 or %IW0.954	iDEA (*DEAB LOCAL INPUTS DEA2*)

6.17.4 External Mechanism Status Signals (IMECH)

Table 6-33: INDRAMAT_DATA_TYPE iMECH declaration

(*EXTERNAL MECHANISM STATUS GATEWAY SIGNALS*)			
	0x.12/xx - 0x.13/xx SOFTWARE VERSION	0x.14/xx and above SOFTWARE VERSION	Type Description
M07S	%IW0.332	A#M07S or %IW0.704	iMECH (*EXT. MECHANISM 7 STATUS*)
M08S	%IW0.340	A#M08S or %IW0.712	iMECH (*EXT. MECHANISM 8 STATUS*)
M09S	%IW0.348	A#M09S or %IW0.720	iMECH (*EXT. MECHANISM 9 STATUS*)
M10S	%IW0.356	A#M10S or %IW0.728	iMECH (*EXT. MECHANISM 10 STATUS*)
M11S	%IW0.364	A#M11S or %IW0.736	iMECH (*EXT. MECHANISM 11 STATUS*)
M12S	%IW0.372	A#M12S or %IW0.744	iMECH (*EXT. MECHANISM 12 STATUS*)
M13S	%IW0.380	A#M13S or %IW0.752	iMECH (*EXT. MECHANISM 13 STATUS*)
M14S	%IW0.388	A#M14S or %IW0.760	iMECH (*EXT. MECHANISM 14 STATUS*)
M15S	%IW0.396	A#M15S or %IW0.768	iMECH (*EXT. MECHANISM 15 STATUS*)
M16S	%IW0.404	A#M16S or %IW0.776	iMECH (*EXT. MECHANISM 16 STATUS*)
M17S	%IW0.412	A#M17S or %IW0.784	iMECH (*EXT. MECHANISM 17 STATUS*)
M18S	%IW0.420	A#M18S or %IW0.792	iMECH (*EXT. MECHANISM 18 STATUS*)
M19S	%IW0.428	A#M19S or %IW0.800	iMECH (*EXT. MECHANISM 19 STATUS*)
M20S	%IW0.436	A#M20S or %IW0.808	iMECH (*EXT. MECHANISM 20 STATUS*)
M21S	%IW0.444	A#M21S or %IW0.816	iMECH (*EXT. MECHANISM 21 STATUS*)
M22S	%IW0.452	A#M22S or %IW0.824	iMECH (*EXT. MECHANISM 22 STATUS*)
M23S	%IW0.460	A#M23S or %IW0.832	iMECH (*EXT. MECHANISM 23 STATUS*)
M24S	%IW0.468	A#M24S or %IW0.840	iMECH (*EXT. MECHANISM 24 STATUS*)
M25S	%IW0.476	A#M25S or %IW0.848	iMECH (*EXT. MECHANISM 25 STATUS*)
M26S	%IW0.484	A#M26S or %IW0.856	iMECH (*EXT. MECHANISM 26 STATUS*)
M27S	%IW0.492	A#M27S or %IW0.864	iMECH (*EXT. MECHANISM 27 STATUS*)
M28S	%IW0.500	A#M28S or %IW0.872	iMECH (*EXT. MECHANISM 28 STATUS*)
M29S	%IW0.508	A#M29S or %IW0.880	iMECH (*EXT. MECHANISM 29 STATUS*)
M30S	%IW0.516	A#M30S or %IW0.888	iMECH (*EXT. MECHANISM 30 STATUS*)
M31S	%IW0.524	A#M31S or %IW0.896	iMECH (*EXT. MECHANISM 31 STATUS*)

6.17.5 Axis Control Signals (qAXIS)

Table 6-34: INDRAMAT_DATA_TYPE qAXIS declaration

(*AXIS CONTROL GATEWAY SIGNALS*)			
	0x.12/xx - 0x.13/xx SOFTWARE VERSION	0x.14/xx and above SOFTWARE VERSION	Type Description
A01C	%QW0.0	A#A01C or %QW0.0	qAXIS (*AXIS 1 CONTROL*)
A02C	%QW0.8	A#A02C or %QW0.20	qAXIS (*AXIS 2 CONTROL*)
A03C	%QW0.16	A#A03C or %QW0.40	qAXIS (*AXIS 3 CONTROL*)
A04C	%QW0.24	A#A04C or %QW0.60	qAXIS (*AXIS 4 CONTROL*)
A05C	%QW0.32	A#A05C or %QW0.80	qAXIS (*AXIS 5 CONTROL*)
A06C	%QW0.40	A#A06C or %QW0.100	qAXIS (*AXIS 6 CONTROL*)
A07C	%QW0.48	A#A07C or %QW0.120	qAXIS (*AXIS 7 CONTROL*)
A08C	%QW0.56	A#A08C or %QW0.140	qAXIS (*AXIS 8 CONTROL*)
A09C	%QW0.64	A#A09C or %QW0.160	qAXIS (*AXIS 9 CONTROL*)
A10C	%QW0.72	A#A10C or %QW0.180	qAXIS (*AXIS 10 CONTROL*)
A11C	%QW0.80	A#A11C or %QW0.200	qAXIS (*AXIS 11 CONTROL*)
A12C	%QW0.88	A#A12C or %QW0.220	qAXIS (*AXIS 12 CONTROL*)
A13C	%QW0.96	A#A13C or %QW0.240	qAXIS (*AXIS 13 CONTROL*)
A14C	%QW0.104	A#A14C or %QW0.260	qAXIS (*AXIS 14 CONTROL*)
A15C	%QW0.112	A#A15C or %QW0.280	qAXIS (*AXIS 15 CONTROL*)
A16C	%QW0.120	A#A16C or %QW0.300	qAXIS (*AXIS 16 CONTROL*)
A17C	%QW0.128	A#A17C or %QW0.320	qAXIS (*AXIS 17 CONTROL*)
A18C	%QW0.136	A#A18C or %QW0.340	qAXIS (*AXIS 18 CONTROL*)
A19C	%QW0.144	A#A19C or %QW0.360	qAXIS (*AXIS 19 CONTROL*)
A20C	%QW0.152	A#A20C or %QW0.380	qAXIS (*AXIS 20 CONTROL*)

6.17.6 Process Control Signals (qPROC)

Table 6-35: INDRAMAT_DATA_TYPE qPROC declaration

(*PROCESS CONTROL GATEWAY SIGNALS*)			
	0x.12/xx - 0x.13/xx SOFTWARE VERSION	0x.14/xx and above SOFTWARE VERSION	Type Description
P00C	%QW0.160	A#P00C or %QW0.400	qPROC (*PROCESS 0 CONTROL*)
P01C	%QW0.184	A#P01C or %QW0.440	qPROC (*PROCESS 1 CONTROL*)
P02C	%QW0.208	A#P02C or %QW0.580	qPROC (*PROCESS 2 CONTROL*)
P03C	%QW0.232	A#P03C or %QW0.520	qPROC (*PROCESS 3 CONTROL*)
P04C	%QW0.256	A#P04C or %QW0.560	qPROC (*PROCESS 4 CONTROL*)
P05C	%QW0.280	A#P05C or %QW0.600	qPROC (*PROCESS 5 CONTROL*)
P06C	%QW0.304	A#P06C or %QW0.640	qPROC (*PROCESS 6 CONTROL*)

6.17.7 Local Output Signals (qDEA)

Table 6-36: INDRAMAT_DATA_TYPE qDEA declaration

(*LOCAL MT-CNC OUTPUTS*)			
	0x.12/xx - 0x.13/xx SOFTWARE VERSION	0x.14/xx and above SOFTWARE VERSION	Type Description
DEA0_Q	%QW0.528	A#qDEA0 or %QW0.904	qDEA (*DEAB LOCAL OUTPUTS DEA0*)
DEA1_Q	%QW0.536	A#qDEA1 or %QW0.924	qDEA (*DEAB LOCAL OUTPUTS DEA1*)
DEA2_Q	%QW0.544	A#qDEA2 or %QW0.944	qDEA (*DEAB LOCAL OUTPUTS DEA2*)

6.17.8 External Mechanism Control Signals (qMECH)

Table 6-37: INDRAMAT_DATA_TYPE qMECH declaration

(*EXTERNAL MECHANISM CONTROL GATEWAY SIGNALS*)			
	0x.12/xx - 0x.13/xx SOFTWARE VERSION	0x.14/xx and above SOFTWARE VERSION	Type Description
M07C	%QW0.328	A#M07C or %QW0.700	qMECH (*EXT. MECHANISM 7 CONTROL*)
M08C	%QW0.336	A#M08C or %QW0.708	qMECH (*EXT. MECHANISM 8 CONTROL*)
M09C	%QW0.344	A#M09C or %QW0.716	qMECH (*EXT. MECHANISM 9 CONTROL*)
M10C	%QW0.352	A#M10C or %QW0.724	qMECH (*EXT. MECHANISM 10 CONTROL*)
M11C	%QW0.360	A#M11C or %QW0.732	qMECH (*EXT. MECHANISM 11 CONTROL*)
M12C	%QW0.368	A#M12C or %QW0.740	qMECH (*EXT. MECHANISM 12 CONTROL*)
M13C	%QW0.376	A#M13C or %QW0.748	qMECH (*EXT. MECHANISM 13 CONTROL*)
M14C	%QW0.384	A#M14C or %QW0.756	qMECH (*EXT. MECHANISM 14 CONTROL*)
M15C	%QW0.392	A#M15C or %QW0.764	qMECH (*EXT. MECHANISM 15 CONTROL*)
M16C	%QW0.400	A#M16C or %QW0.772	qMECH (*EXT. MECHANISM 16 CONTROL*)
M17C	%QW0.408	A#M17C or %QW0.780	qMECH (*EXT. MECHANISM 17 CONTROL*)
M18C	%QW0.416	A#M18C or %QW0.788	qMECH (*EXT. MECHANISM 18 CONTROL*)
M19C	%QW0.424	A#M19C or %QW0.796	qMECH (*EXT. MECHANISM 19 CONTROL*)
M20C	%QW0.432	A#M20C or %QW0.804	qMECH (*EXT. MECHANISM 20 CONTROL*)
M21C	%QW0.440	A#M21C or %QW0.812	qMECH (*EXT. MECHANISM 21 CONTROL*)
M22C	%QW0.448	A#M22C or %QW0.820	qMECH (*EXT. MECHANISM 22 CONTROL*)
M23C	%QW0.456	A#M23C or %QW0.828	qMECH (*EXT. MECHANISM 23 CONTROL*)
M24C	%QW0.464	A#M24C or %QW0.836	qMECH (*EXT. MECHANISM 24 CONTROL*)
M25C	%QW0.472	A#M25C or %QW0.844	qMECH (*EXT. MECHANISM 25 CONTROL*)
M26C	%QW0.480	A#M26C or %QW0.852	qMECH (*EXT. MECHANISM 26 CONTROL*)
M27C	%QW0.488	A#M27C or %QW0.860	qMECH (*EXT. MECHANISM 27 CONTROL*)
M28C	%QW0.496	A#M28C or %QW0.868	qMECH (*EXT. MECHANISM 28 CONTROL*)
M29C	%QW0.504	A#M29C or %QW0.876	qMECH (*EXT. MECHANISM 29 CONTROL*)
M30C	%QW0.512	A#M30C or %QW0.884	qMECH (*EXT. MECHANISM 30 CONTROL*)
M31C	%QW0.520	A#M31C or %QW0.892	qMECH (*EXT. MECHANISM 31 CONTROL*)

7. TOOL MANAGEMENT

7.1 Introduction to Tool Management

Tool management provides the compensation for tool length, tool nose radius and mill radius path, monitoring of tool life, warning limits, and automatic tool wear compensation.

Each process (0...6) is capable of controlling one magazine or turret and the tool changes between the available spindles (1...4), grippers (1...4) and tool storage (magazine/turret). System and process parameter setup defines these capabilities. Tool storage movements are asynchronous to the axis movements of a process.

The tool management of the MT-CNC handles numerical (servo) as well as programmable controller (SPS) controlled magazines and turrets.

7.2 Tool Setup List

A tool entry into the Setup List must be made for each tool that is used within the NC-program(s) of a process. The tool data requirements that have been specified during the NC-program generation are loaded together with the NC-program package into the MT-CNC.

The actual tool data can be prepared and loaded via the MUI. Tool data can also be entered, erased, and modified online in the MT-CNC, using functions of the MUI and SOT.

7.2.1 Tool Lists and Setup Lists

The tool management performs automatic tool checks whereby it compares the Active Tool List with the active Setup List. Tools contained in the Setup list must be present in the Active Tool List and fulfill the Setup List requirements.

7.2.2 Basic Tool Data of Setup List

7.2.2.1 Tool Name (ID)

The tool name is part of the Tool Identification (ID). The Tool Name can consist of up to any 28 characters. Tools are displayed in the MUI and SOT menus with their Tool Name as well as their T-number.

Two different tools are not allowed to have the same Tool Name. Tools which could replace each other can be comprised of a tool family by using the same Tool Name. The tools within the tool family are distinguished in the Tool Index.

7.2.2.2 T-Number (T)

Tools or pockets can be assigned in the NC-program by using the T-word which is part of the NC-command set. The T-word represents the T-number and consists of the letter "T", followed by the tool's or pocket's number (T0...T999999).

The relationship between the T-number and tool is valid within the process and its NC-programs if the Setup lists are station specific. It is only valid for the particular NC-program if the Setup Lists are NC-program specific.

7.2.2.3 Amount of Tool Edges

Up to nine tool edges can be assigned to each tool. The system parameter Axx.054 can be used to limit the maximum amount of Tool Edges for all processes.

7.2.2.4 Time Unit

The time units "minute" [MIN] and "cycles" [CYCL] can be used.

Tool life data, with the exception of "Remain. Usage Life" (%) and "Used Life" (%), is registered and updated based on the selected time unit.

7.2.2.5 Length Unit

The geometric tool data can be entered with the units "millimeter" [MM] or "inch" [INCH]. The length unit in the Setup List does not have to be the same as the one specified in the Tool List. Geometric tool data is always converted into the basic process unit (Process Parameter Bxx.001) when loading the Tool List and Setup List into the MT-CNC.

7.2.2.6 Comment

The system parameter Axx.057 enables a comment field for each tool in the Setup List. The comment can consist of up to 5 lines, 76 characters each.

NOTE: *The comment is available in the PC (MUI) only!*

7.2.3 Tool Edge Data

The tool edge data contains geometric, life, status and user data.

7.2.3.1 Tool Edge Orientation

Tool edge orientation allows the measuring of tools with correction type 3(turning tool) and 4 (right angle mill) relative to their theoretical Edge "P", without having inaccuracies during the tool usage. The theoretical Edge "P" is used by the CNC if radius compensation for tool nose or mill is active (G42/G43). The theoretical Edge "P" moves then on the programmed path.

7.2.3.2 Maximum Tool Life

The maximum tool life is the tool's maximum machining time, minutes or NC-program (cutting) cycles. The machining time is usually the time in which the resharpener tool cuts under steady conditions until it is worn-out.

NOTE: *The tool life monitoring is switched off if a value of "0" is entered. The updating of the Maximum Tool Life is then suppressed.*

7.2.3.3 Tool Life

The tool life is defined as the time during which the tool is cutting the part.

7.2.3.4 Maximum and Minimum Tool Length

Geometric length limitation values are used to perform tool length checks. The length of all tools within a tool family that are in the Active Tool List must be within the length limits specified for the same tool in the Setup List. Only tools that fulfill this requirement will be used by the tool management for machining.

NOTE: *Setup List tool length limits are not considered for tool corrections.*

7.2.3.5 Maximum and Minimum Tool Radius

Radius limitation values are used to perform tool radius checks. The radius of all tools within a tool family that are in the Active Tool List must be within the radius limits specified for the same tool in the Setup List. Only tools that fulfill this requirement will be used by the tool management for machining.

NOTE: *Setup List tool length limits are considered for tool corrections.*

7.2.3.6 Length Tool Wear Factor (L1, L2, L3)

The length wear factor can be used to compensate tool's length wear. Tool length wear can be activated via G48 and G49. The length compensation value is calculated by multiplying the tool's usage time with its length wear factor. The usage time is determined while G1, G2, G3, G63, G64, G65, and G33 are active if the length wear factor has the unit "MM/MIN" or "INCH/MIN". The usage time is one cycle and the compensation is the amount specified as length wear factor, if the unit of the length wear factor is "MM/CYCL" or "INCH/CYCL".

Tool Management updates the usage time and compensates the length wear automatically if:

- A different tool edge is selected (NC-command E)
- The same tool edge is selected (NC-command E)
- The tool is put back into the tool magazine
- The tool is taken out of the machining position
- T0 (no tool) is selected in the case of a tool turret or no tool storage (magazine/turret) at all

7.2.3.7 Radius Tool Wear Factor

The radius wear factor can be used to compensate tool's radius wear, which can be activated via G41 and G42. The radius compensation value is calculated by multiplying the tool's usage time with its radius wear factor. The usage time is determined while G1, G2, G3, G63, G64, G65, and G33 are active if the radius wear factor has the unit "MM/MIN" or "INCH/MIN". The usage time is one cycle and the compensation is the amount specified as radius wear factor, if the unit of the radius wear factor is "MM/CYCL" or "INCH/CYCL".

Tool Management updates the usage time and compensates the length wear automatically if:

- A different tool edge is selected (NC-command E)
- The same tool edge is selected (NC-command E)
- The tool is put back into the tool magazine
- The tool is taken out of the machining position
- T0 (no tool) is selected in the case of a tool turret or no tool storage (magazine/turret) at all

7.2.4 Setup List Organization

Setup Lists can be organized either as station (process) or NC-program specific. The System Parameter Axx.053 is used to select type of Setup List organization.

Station (process) specific:

This Setup List selection creates only one Setup List per process in the NC-program package. Up to 99 NC-programs per process share this same Setup List.

NC-program Specific:

This Setup List selection creates one Setup List for each NC-program of a NC-program package.

7.2.5 Setup List Within the MT-CNC

Tool Setup Lists (SL) can be generated or modified by using the MUI, independent of the active NC-program. The MUI main menu "2 = NC-program Administration" contains the NC-program index that allows NC-program modifications. The key <F5> allows access to the (SL) for the process or NC-program.

The (SL) are always transferred together with the NC-program package into the MT-CNC memory A or B via the MUI. The (SL) will be stored together with the NC-program since they are part of the NC-program package. Also the (SL) data is not modified while the (SL) is active in the MT-CNC memory.

7.3 Tool List

Tool Lists are used to prepare and store tool data sets. Besides the basic tool data, the Tool List also contains the tool Edge data (Edge ID, geometry, tool life and wear, and user data) of all defined tools.

CAUTION *Changes, such as inserting/deleting or moving tools, in the Active Tool List must reflect the real facts in the tool storage (magazine/turret) and vice versa to prevent damage to man and machine.*

7.3.1 Tool List Data

The Tool List consists of the summary of all tool entries in the tool storage (magazine/turret). The tool data in the Tool List consists of the basic tool data and the data of different tool Edges.

7.3.2 Basic Tool Data of Setup List

7.3.2.1 Tool Name

The tool name is part of the Tool Identification (ID). The Tool Name can consist of up to any 28 characters. Tools are displayed in the MUI and SOT menus with their Tool Name as well as their T-number.

Two different tools are not allowed to have the same Tool Name. Tools which could replace each other can be comprised of a tool family by using the same Tool Name. The tools within the tool family are distinguished in the Tool Index.

7.3.2.2 Tool Storage

The tool storage data of the Tool List is not completely available since only the tool storage (tool storage = 0) and the spindles (tool storage = 1) are registered.

7.3.2.3 Tool Location

The "tool location" data identifies within the tool storage the tool pocket or spindle number that contains the tool. Using the Tool List, all tool locations in the tool storage as well as all available spindles can be prepared with tool data.

CAUTION *After loading the Tool List into the MT-CNC it becomes an Active Tool List. Changes, such as inserting/deleting or moving tools, in the Active Tool List must reflect the real facts in the tool storage(magazine/turret) and vice versa to prevent damage to man and machine.*

7.3.2.4 Tool Index

The Tool Index is used to:

- Identify spare tools within a tool family (since tools of a tool family have the same tool name (ID))
- Determine the sequence of their usage

Tools of a tool family are used according to their tool index number. Tools that are not worn out and have a lower index number are used first, before tools with a higher index number.

All tools of a tool family are called with the same T-number that is specified with the tool name. Spare tools are used whenever the T-number is programmed and the previous tool has been locked out.

7.3.2.5 Tool Correction Type

The Tool Correction Type specifies the amount of tool's corrections and their orientations.

- Correction Type 1:** (drilling tool)
Tools of this correction type have only one length correction (L3), that is always perpendicular to the plane.
- Correction Type 2:** (milling tool)
Tools of this correction type have only one length correction (L3), that is always perpendicular to the active plane. A radius correction (R) in the active plane is also performed.
- Correction Type 3:** (turning tool)
Tools of this correction type have two length corrections (L1, L2) and a radius correction within the active plane.
- Correction Type 4:** (right angle milling tool)
Tools of this correction type can have length correction (L1, L2, L3) for all three main axis (X, Y, Z) and a radius correction (R) in the active plane. The length L3 is always perpendicular to the active plane, while L1 and L2 reside in the active plane.
- Correction Type 5:** (tool gripper)
Tool of this correction type can perform length corrections in all directions of the main axes (X,Y,Z) via the length correction (L1,L2,L3). Length L3 works always perpendicular to the active plane, while length L1 and L2 always work in the active plane. The correction type 5 corresponds to correction type 4 with the difference that it has no radius R correction.

7.3.2.6 Amount of Tool Edges

Up to nine tool edges can be assigned to each tool. The system parameter Axx.054 can be used to limit the maximum amount of Tool Edges for all processes.

7.3.2.7 Half Pocket Overlap

Tools whose width covers more than one pocket are identified via the "Half Pocket Overlap" entry.

- 0 : The tool has a normal width and symmetrically covers one location (pocket)
- 1 : The tool symmetrically covers half of the left and half of the right (location) pocket
- 2 : The tool symmetrically covers the complete (2 half) left and right location (pocket)
- 3 : The tool symmetrically covers one and a half locations (pockets) to the left and to the right.
- 4 : The tool symmetrically covers the two complete (4 half) left and right location (pockets)

Location (pockets) covered by oversized tools are identified in the Setup Lists and Tool Lists via dashed lines.

7.3.2.8 Old Location

The basic tool data "Old Pocket" is not displayed in the Tool List. However, it is necessary for tools in the spindle, that require a fixed pocket and is therefore contained in the data.

7.3.2.9 Length Unit

The geometric tool data can be entered with the units "millimeter" [MM] or "inch" [INCH].

The length unit in the Setup List does not have to be the same as the one specified in the Tool List. Geometric tool data is always converted into the basic process unit (Process Parameter Bxx.001) when loading the Tool List and Setup List into the MT-CNC.

7.3.2.10 User Tool Data (1...9)

User tool data (1...9) can be used to attach user specific data to a tool. The user tool data can be defined in the System Parameters (Axx.061...Axx.069) by entering a name of up to 13 characters. These names and their associated data are then displayed in the MUI and SOT via the Tool List.

7.3.3 Tool Edge Data

7.3.3.1 Tool Edge Orientation

Tool edge orientation allows the measuring of tools with correction type 3 (turning tool) and 4 (right angle mill) relative to their theoretical Edge "P", without having inaccuracies during the tool usage. The theoretical Edge "P" is used by the CNC if radius compensation for tool nose or mill is active (G42/G43). The theoretical Edge "P" moves then on the programmed path.

7.3.3.2 Remaining Tool Life

The Remaining Tool Life defines the tool's wear status in %, independent of the tool/workpiece material combination.

$$\text{Remain. Tool Life [\%]} = \frac{\text{Tool Life [MIN] or [CYCL]}}{\text{Max. Tool Life [MIN] or [CYCL]}} * 100\%$$

A new or resharpened tool has a Remaining Tool Life of 100%. The Remaining Tool Life of a worn out tool is 0%. Using the Remaining Tool Life (%), the tool management controls the wear status of the tools, independent of the tool/workpiece material combinations.

The Remaining Tool Life is updated:

- During the automatic tool check
- When changing the tool Edge (E1, then E3)
- When calling the same tool Edge (E1, then E3)
- When the tool is put back into the tool storage (magazine)
- When the turret swivels a new tool into machining position
- When the tool "T0" is selected (tool storage = turret or no tool storage present)

NOTE: *The tool management uses "1 cycle" for Tool Life if the unit [CYCL] for Tool Life was selected for the tool in the Setup List.*

Each time the tool management updates tool data it checks the remaining Tool Life of all tools. The status of the Gateway signal PxxS.MGTWO is set to "1" if the remaining Tool Life of all tools within a tool family is zero.

7.3.3.3 Warning Limit

The tool management monitors the remaining Tool Life (%) and uses the Gateway process status signal PxxS.MGWRN to indicate that the remaining Tool Life time (%) is below the specified limit. The status of PxxS.MGWRN is set to "1" at the time of tool check if the last tool within the tool family exceeded its warning limit. The warning limit is a percentage value of the Max. Tool Life which is 100%.

7.3.3.4 Geometric Data

The tool dimensions are automatically compensated by using the geometric data if the compensation is activated.

Table 7-1: Geometry Data Groups

Geometry	Wear	Offset
Length L1	Wear L1	Offset L1
Length L2	Wear L2	Offset L2
Length L3	Wear L3	Offset L3
Radius R	Wear R	Offset R

The wear and offset registers can be selected as an option via the System Parameters Axx.055 and Axx.056.

- Geometry Registers:** Are used as program independent storage, used to compensate the tool dimensions.
- Wear Registers:** Wear registers are used by the tool management to compensate for the tool wear that occurs over time or cycles.
- The "Reset" function of the MT-CNC will reset the tool's wear registers to "0" and the remaining tool life to 100%.
- Offset Registers:** Offset registers are not influenced by tool management. Like wear registers, they can be used to compensate for dimensional deviations. The deviations are either determined by the user or an automatic gauging system.

NOTE: *Wear and offset registers are internally set to "0" if they are not selected via the System Parameters Axx.055 and Axx.056.*

7.3.3.5 Tool Length Correction

The length "L1", "L2", and "L3" of a tool Edge is calculated as follows:

$$\text{Length Correction L1} = \text{Length L1} + \text{Wear L1} + \text{Offset L1}$$

$$\text{Length Correction L2} = \text{Length L2} + \text{Wear L2} + \text{Offset L2}$$

$$\text{Length Correction L3} = \text{Length L3} + \text{Wear L3} + \text{Offset L3}$$

7.3.3.6 Tool Radius Correction

The radius "R" correction of a tool Edge is calculated as follows:

$$\text{Length Correction R} = \text{Length R} + \text{Wear R} + \text{Offset}$$

7.3.3.7 User Edge Data

The user Edge data (part of Edge data) enables user specific data to be assigned to each Edge. User Edge data registers are provided in the Tool List (MUI and SOT) after declaring names (up to 13 characters) in the System Parameters (Axx.070...Axx.074). User Edge data can be prepared in the Tool List like any other data.

7.3.4 Tool List within the Control System

MUI menu item 6 "Tool Management" and its sub menu "Tool Lists" allow to generate up to 99 Tool Lists for each process. The Tool List can be created, erased, copied, saved, and modified independently from the active NC-program.

7.4 Active Tool List

The Active Tool List of a process can be viewed in ONLINE mode in the MUI as well as the SOT. In addition to the actual tool data, the corresponding Setup List data is displayed also in the Active Tool List. This provides all information to assure problem recognition and recovery in the case of a tool data mismatch or error.

CAUTION *After loading the Tool List into the MT-CNC it becomes an Active Tool List. Changes, such as inserting/deleting or moving tools, in the Active Tool List must reflect the real facts in the tool storage (magazine/turret) and vice versa to prevent damage to man and machine.*

7.4.1 Data of the Active Tool List

The Active Tool List consists of the summary of all tools that are present in the tool storage (magazine/turret), tool spindles and grippers. The following are descriptions of the data contained in an Active Tool List:

Location.	Tool storage location
T-number	Tool number in the NC-program
Name	Tool identification
Index	Index to distinguish tools of a family
Status	The following tool status bits are displayed in the status column of the Active Tool List:

Table 7-2: Tool Status Bits

STATUS	DESCRIPTION
(1...8)	User tool status bit (1...8). Characters are defined by the user in System Parameters (Axx.075...Axx.082)
C	Tool has a fixed location (pocket/position)
L	Tool is locked (can not be used)
p	Primary tool
s	Spare tool of tool family
w	Warning limit exceeded (tool life)
d	Tool worn out
B	Location is blocked (can not be used)
f	Incorrect tool Edge data (L1, L2, L3, R)
e	Incorrect number of tool edges
t	Incorrect correction type
?	Tool not required? (when sorted by location or T-number)
!	Tool not present (when sorted by T-number only)

The MUI and SOT display of the Active Tool List should be used to identify the problem if one of the following process error(s) is active:

- #314 "Programmed tool not found"
- #315 "No empty tool location found"
- #316 "No acceptable tool found"
- #318 "Tool List doesn't fulfill Setup List requirements"
- #325 "One or more tools worn out"
- #383 "Correction value deviation"

Table 7-3: Basic Tool Data of Active Tool List

Description	Value Range	Unit	Short ID	Opt.	Setup List	Tool List
Basic Tool Data						
Index Address	Hexadecimal long word with 32 Bit (read only)	-	E0.D01			
Tool Name (ID)	Up to 28 case sensitive characters	-	E0.D02		X	X
Tool Storage	(0...2) (0 = Mag./Turret, 1 = Spindle, 2 = Gripper)	-	E0.D03			X
Tool location	(0...999)	-	E0.D04			
T-Number	(1...9999999)	-	E0.D05		X	
Index Number	(1...999)	-	E0.D06			X
Correction Type	(1...4)	-	E0.D07		X	X
Amount of Edges	(1...9)	-	E0.D08		X	X
Tool Status	0/1 (32 Status Bits)	-	E0.D09			X
Pocket Data						
Half pocket overlap	(0...4)	-	E0.D10			X
Old pocket	(1...999)	-	E0.D11			X
Next E/T memory	(0...2) (0 = Mag./Turret, 1 = Spindle, 2 = Gripper)	-	E0.D12			
Next E/T location	(1...999)	-	E0.D13			
Prev. E/T memory	(0...2) (0: Mag./Turret, 1 = Spindle, 2: Gripper)	-	E0.D14			
Prev. E/T Location	(1...999)	-	E0.D15			
Units						
Time Unit	0/1 (0: {MIN}, 1: [CYCL])	-	E0.D16		X	
Length	0/1 (0: {MM}, 1: [INCH])	-	E0.D17		X	X
User Tool Data						
User tool data (1...9)	(+/- 1.2*10 ⁻³⁸ ... +/- 3.4*10 ⁺³⁸ and 0)	any	E0.D18 : E0.D26	X X		X X
Comment						
	Up to 5 lines, 76 characters each	-		X	X	

Table 7-4: Tool Edge Data of Active Tool List

Description	Value Range	Unit	Short ID	Opt	Setup List	Tool List
Edge ID						
Edge Orientation	(0...8)	-	En.D01		X	X
Edge Status	0/1 (16 status bits)	-	En.Do2			X
Tool Life Data						
Remain usage. life	(0.00...100.00)	%	En.D03	X		X
Warning Limit	(0.00...100.00)	%	En.D04	X		X
Max. Useful Life	(0...9999999) (0= Tool life monitoring disabled)	MIN / CYCL	En.D05	X	X	
Used Life	(0...9999.999)	MIN / CYCL	En.D06	X	X	
Geometric Data						
Length L1			En.D07			X
Length L2			En.D08			X
Length L3			En.D09			X
Radius R			En.D10			X
Wear L1	-9999.9999...9999.9999 or	MM /	En.D11	X		X
Wear L2	-999.99999...999.99999	INCH	En.D12	X		X
Wear L3			En.D13	X		X
Wear R			En.D14	X		X
Offset L1			En.D15	X		X
Offset L2			En.D16	X		X
Offset L3			En.D17	X		X
Offset R			En.D18	X		X
Geometry Limits						
L1_min			En.D19	X	X	
L1_max			En.D20	X	X	
L2_min	-9999.9999...9999.9999 or	MM /	En.D21	X	X	
L2_max	-999.99999...999.99999	INCH	En.D22	X	X	
L3_min			En.D23	X	X	
L3_max			En.D24	X	X	
R_min			En.D25	X	X	
R_max			En.D26	X	X	
Tool wear factor						
Wear Factor L1		MM /	En.D27	X	X	
Wear Factor L2	-9999.9999...9999.9999 or	INCH	En.D28	X	X	
Wear Factor L3	-999.99999...999.99999	MIN /	En.D29	X	X	
Wear Factor R		CYCL	En.D30	X	X	
User Edge Data						
User edge data 1	(+/- 1.2*10 ⁻³⁸ ...		En.D31	X		X
:	+/- 3.4*10 ⁺³⁸ and 0)	any	:	:		:
User edge data 5			En.D35	X		X

Table 7-5: Tool Status Bit Overview

Group	Definition	Sym	Write by	Type	Bit	Val	Remark
Presence	Tool not present / Tool present	!	Tool Mgmt	Setup List	1	1 0	Tool Missing
	Tool not required? / Tool required	?	Tool Mgmt	Setup List	2	1 0	Tool not required?
Correction Type Incorrect	Incorrect correction type / correction type OK	t	Tool Mgmt	Setup List	3	1 0	Correction type does not meet reqrmt.
Incorrect Nr. of tool Edges	Incorrect Nr. of Edges / Number of Edges correct	e	Tool Mgmt	Setup List	4	1 0	Number of tool Edge does not meet reqrmt.
Incorrect Edge Data	Edge data incorrect / Edge data correct	f	Tool Mgmt	Setup List	5	1 0	Tool Edge data does not meet requirements
Spare	-	-	-	-	6	-	-
Spare	-	-	-	-	7	-	-
Spare	-	-	-	-	8	-	-
Block location	Location blocked / Location not blocked	B	User Prg/Op	Locat Status	9	1 0	e.g., location block since damaged
	Upper half Location. blocked / Upper half Location. not blocked		Tool Mgmt	Locat. Status	10	1 0	Blocked for fixed Location. tool in grp. or spindle
	Lower half location. blocked / Lower half location. not blocked		Tool Mgmt.	Locat. Status	11	1 0	Blocked for fixed location. tool in grp. or spindle
Location Reservation	Upper half location. reserved / Upper half location. not reserved		User Prgm	Locat. Status	12	1 0	e.g., to channel a tool in
	Lower half location. reserved / Lower half location. not reserved		User Prgm	Locat. Status	13	1 0	e.g., to channel a tool in
Location Occupancy	Upper half location. occupied / Upper half location. not occupied		Tool Mgmt	Locat. Status	14	1 0	A tool occupies the upper half location
	Lower half location. occupied / Lower half location. not occupied		Tool Mgmt	Locat. Status	15	1 0	A tool occupies the lower half location
	Location occupied/ Locat not occupied		Tool Mgmt	Locat. Status	16	1 0	The location is occ. by tool

Table 7-6: Tool Status Bit Overview (cont.)

Group	Definition	Sym	Write by	Type	Bit	Val	Remark
Tool Wear Status	Tool worn out / Tool not worn out	d	Tool Mgmt.	Tool Status	17	1 0	Tool worn out (exchange)
	Tool warning limit exceeded / Tool warning limit not exceeded	w	Tool Mgmt.	Tool Status	18	1 0	Warning limit of tool life is exceeded
Spare Tools (tool family)	Primary tool / Not a primary tool	p	Tool Mgmt.	Tool Status	19	1 0	Primary tools of tool family
	Spare tool / Not a spare tool	s	Tool Mgmt.	Tool Status	20	1 0	Spare tools replace worn primary tool
Fixed Locat.	tool fixed location. / tool no fixed location.	C	User Prg/Op	Tool Status	21	1 0	Tool location. in tool mgmt fixed
Tool Locking	Tool is locked / Tool is not locked	L	User Prg/Op	Tool Status	22	1 0	Tools can be locked
Spare	-	-	-	-	23	-	-
Spare	-	-	-	-	24	-	-
User Tool Status (1...8)	User tool status bit (1...8) set / User tool status bit (1...8) reset	any	User Prg/Op	Tool Status	25 : 32	1 0	any

Explanations:

Tool Mgmt. : Tool Management
 User Prg. : User Programs of Programmable Controller (SPS) and NC
 Op : Operator
 Locat. Status : Location specific status bit

- NOTE:**
- The group definitions are for display purpose only. The groups are not recognized as such in the MT-CNC.
 - Status bits that are identified with upper case letters can be influenced by the user.
 - Status bits that are identified via lower case letters can not be influenced by user.

7.4.2 Tool Edge Status Bits

Table 7-7: Tool Edge Status Bit Overview

Group	Definition	Sym	Write by	Type	Bit	Val	Remark
Invalid Edge Orientation	Edge orientation invalid / Edge orientation valid	e	Tool Mgmt.	Setup List	1	1 0	1: Tool Edge orient. differs from one in Setup List
Invalid Tool Length L1	Tool length L1 invalid / Tool length L1 valid	1	Tool Mgmt.	Setup List	2	1 0	1: Tool length differs from necessary dimensions
Invalid Tool Length L2	Tool length L2 invalid / Tool length L2 valid	2	Tool Mgmt.	Setup List	3	1 0	1: Tool length differs from necessary dimensions
Invalid Tool Length L3	Tool length L3 invalid / Tool length L3 valid	3	Tool Mgmt.	Setup List	4	1 0	1: Tool length differs from necessary dimensions
Invalid Tool Radius R	Tool radius R invalid / Tool radius R valid	r	Tool Mgmt.	Setup List	5	1 0	1: Tool radius differs from necessary dimensions
Spare	-	-	-	-	6	-	-
Spare	-	-	-	-	7	-	-
Spare	-	-	-	-	8	-	-
Edge Wear Status	Edge worn out / Edge not worn out	d	Tool Mgmt.	Setup List	9	1 0	1: Remaining tool life is zero
	Edge warning limit exceeded / Edge warning limit not exceeded	w	Tool Mgmt.	Setup List	10	1 0	Remaining tool life exceeded warning limit value
Spare	-	-	-	-	11	-	-
Spare	-	-	-	-	-	-	-
User Edge Status (1...4)	User edge status bit (1...4) set / User edge status bit (1...4) reset	any	User Prg/Op	Tool Status	16	1 0	any

Explanations:

Tool Mgmt. : Tool Management
 User Prg. : User Programs of Programmable Controller (SPS) and NC
 Op : Operator
 Locat. Status : Location specific status bit

7.5 Tool Management Parameters

7.5.1 System Parameters

A00.052 Tool Management

This parameter should be set to “yes” if any process will be using tool management. As soon as Tool Management is enabled, further system parameters (A00.053...A00.086, A00.091...A00.096) appear which configure the tool data sets within the system. Also Setup lists can be generated for the NC-program or process (depending on Axx.053) and Tool Management MUI menu item #6, SOT menu item #3, and Tool Management GUI is activated.

A00.053 Organization of the Setup List Station Specific

The Setup List (part of the NC-program package) is common for all 99 NC-programs per process.

Program Specific

The Setup List is unique for each of the 99 NC-programs per process.

A00.054 Maximum Amount of Tool Edges

This parameter specifies the maximum number of cutting edges per tool that can be monitored. Each tool can have up to nine edges.

A00.055 Wear Factor Register A00.056 Offset Register

The “Wear Factor Register” parameter allows for tool wear data to be included in the tool data set, and the “Offset Register” parameter allows for offset data to be included in the tool data set. If “yes” is selected, these registers appear in the Tool Management menu of the MUI and SOT, and they include entry fields for the lengths L1, L2, L3, and radius R.

If Wear Factors are enabled (system parameter A00.058) the MT-CNC uses the wear records to automatically compensate for the wear of the tools. If the Wear factors are not chosen, the wear records are used as a resettable tool compensation.

The wear records are also influenced by the MUI “Reset” function. Besides resetting the status time to 100%, the reset - function resets all wear records belonging to the tool to zero. The offset register however, is not reset by the MUI “Reset” function.

A00.057 Comment (Assembly Instruction)

This parameter opens up a field of up to 5 lines with a maximum of 75 characters each in the Setup List. The comments are only available in the MUI Setup List. They are not available in the SOT.

A00.058**Wear Factors**

This parameter activates the automatic Wear compensation feature of the MT-CNC Tool Management. The Wear compensation is activated if a value other than zero is entered in the tool “W.Factor L1”, “W.Factor L2”, “W.Factor L3”, “W.Factor R” field, and this parameter is selected. If this parameter is not selected, entry fields for wear factors are not displayed in the Setup List, but can still be entered in the Wear register of the Active Tool List if the system parameter A00.055 is set to “Yes”.

Wear L1=Wear L1+Processing duration Length wear element L1
 Wear L2=Wear L2+Processing duration Length wear element L2
 Wear L3=Wear L3+Processing duration Length wear element L3
 Wear R=Wear R+Processing duration Length wear element R

The Tool Management updates the wear under these circumstances:

- When changing to another tool Edge
- When calling the same edge again
- If the tool is taken from the magazine (magazine type tool storage)
- If the tool is swung to the machining position(turret type tool storage)
- If the tool becomes de-selected with “T0” (turret type tool storage, or no tool storage available)

A00.059**Tool Life Data**

Determines whether the Tool Life data on the MUI and the SOT’s is available. If “no” is selected, these functions will not be available in the Setup and Tool Lists:

- Remaining tool life
- Warning limit
- Maximum tool life

If status time supervision is not desired, the corresponding supervision mechanisms can be switched off through the de-selection of this parameter.

A00.060**Tool Geometry Limits**

Allows geometry limiting values to be entered in the Tool List on the MUI and the SOT(s). Tool geometry limits make it possible to check a tool in the tool storage with regard to their suitability for immediate use.

$L1_min < L1 < L1_max$
 $L2_min < L2 < L2_max$
 $L3_min < L3 < L3_max$
 $R_min < R < R_max$

A00.061**Description of User Tool Data (1..9)**

:

A00.069

The user data (1..9) allow arbitrary user specific data (13 characters; any ASCII text except /, !, or #) to be given to each tool in the tool set. The desired user data is entered in each tool data set, on the MUI, SOT, or NC-command TLD.

A00.070 **Description of User Edge Data (1...10)**^{(4),(5)[A00.092...A00.096]}
 :
A00.096 Depending on the setting of system parameter A00.091 “Tool Technology” the following system parameters have different functionality:

A00.091 set to ‘Grinding’

A00.070 Defined as min. spindle speed [S-min]
 A00.071 Defined as max. spindle speed [S-max]
 A00.072 Defined as max. grind Disk Circumference Speed [DCS-max]
 A00.073 Defined as slant angle [Slant Angle]
 A00.074 Defined as associated follower edge [ass.foll-E.]

A00.091 set to “Turn./Mill”

The user data (1...10) allow arbitrary user specific edge data (13 characters; any ASCII text except /, !, or #) to be given to each tool in the tool set. The desired user edge data is entered in each tool data set, on the MUI, SOT, or NC-command TLD

A00.075 **Symbol for User Tool Status Bit (1...8)**
 :
A00.082 These parameters allow arbitrary user specific status information (1 character; any ASCII text except /, !, or #) for each tool to be given via user - tool status bits (1...8). The desired user data is entered in each tool data set, on the MUI, SOT, or NC-command TLD.

A00.083 **Symbol for User Tool Edge Status Bit (1...4)**
 :
A00.086 These parameters allow arbitrary user specific status information (1 character; any ASCII text except /, !, or #) for each tool edge to be given via user - tool edge status bits (1...4). The desired user edge symbol is entered in each tool data set, on the MUI, SOT indicated and includes a value entry for the corresponding bit.

NOTE: *All user status bits and data are denoted in **Upper Case Letters**, while the internal status bits and data are **lower case**.*

A00.091 **Tool Technology**^{(4),(5)}
 This parameter can be used to specify special functions for the ‘User Tool Edge Data 1...5’ (A00.070...A00.074), used for grinding. If set to ‘Turn./Mill’, the user can freely specify (A00.070...A00.074).

Bxx.014 **Tool Management**
 Once set to “yes”, further process parameters (Bxx.015...Bxx.028) appear which configure the tool data sets within the process. Also, menu items to generate and administrate tool Setup Lists are available in MUI menu item #2.

Bxx.015 **Type of Tool Storage**
 Either magazines or turrets can be used as tool storage. Magazines necessitate both a physical as well as logical tool transfer between the available spindles and the magazine, using grippers if applicable. A turret keeps the tools in the same relative position. The tools are brought in for use solely through rotation of the turret to the proper position.

Bxx.016 **Endless Rotating Tool Storage**
 This parameter allows for magazines and turrets to rotate in the same direction endlessly. It is important to insure that the distance between the last and first tool storage places is equal to all others and that the first place follows the last place when rotating the tool storage unit.

- Bxx.017** **Number of Tool Locations in Tool Storage**
This parameter specifies the number of tool locations (pockets of magazine, position of turret) in the tool storage. It is only used if tool storage is needed.
- Bxx.018** **Number of Spindles with Tool**
This parameter specifies the number of Tool spindles used with this process. Tool spindles are defined as spindles that turn tools whose data is managed by the tool management of the MT-CNC. During processing only one tool and tool spindle can be active per process, for which the Tool Management updates the tool corrections as well as the status time and wear data.
- Bxx.019** **Number of Grippers**
This parameter is used when the tool changer in the process uses grippers to exchange the tool between the spindle and the magazine.
- Bxx.020** **Axis Number for Magazine Axis**
Specifies the MT-CNC axis number to be used to move the tool storage axis. This number correlates with the system parameters (A00.032...A00.051). This parameter is only used if the MT-CNC controls a servo tool storage axis for this process.
- NOTE:** *The axis defined in this parameter can NOT be switched to a secondary process (Cxx.000)! It must remain in its primary process, since the tool storage (management) can NOT be switched.*
- Bxx.021** **Position (1...4) (Offset to the Reference)**
:
Bxx.024 This parameter allows the definition of up to four arbitrary positions for the tool storage unit which can be used for tool change, exchange, or processing. These positions are in relation to the tool change reference position. The reference is defined as tool pocket or position "1" and is the position the magazine goes to when executing the NC-command MHP. The offsets relative to the reference are counted in the direction of ascending position number, in positive direction.
- Bxx.025** **Max. Entry for Length Wear**
Bxx.026 **Max. Entry for Radius Wear**
Bxx.027 **Max. Entry for Length Offset**
Bxx.028 **Max. Entry for Radius Offset**

These parameters limit the wear and offset entry into the Active Tool List, for both the tool length and the tool radius.

The values in these fields are used for the absolute and the incremental entries through the MUI and SOT. Values entered via the NC-command TLD will be checked during the next automatic tool check only!

Bxx.043 Activate Tool Correction^{(2),(3),(4),(5)}

Tool correction can be activated one of three ways:

Immediately After Stop

A tool data modification is only possible when a process stop is active (PxxS.STOP=1) or after deselecting a tool edge (deactivating a tool). The correction value will be processed with a new process advance, reverse or after another tool edge selection (tool activation).

Next Activation

A tool data modification is possible any time. The modified correction values is considered at the next activation of the tool. The old correction values remain active until the tool is deactivated.

Next Possible NC-block

A tool data modification is possible any time. The modified correction values are considered in the NC-block that is being loaded into the NC-block preparation memory next.

Bxx.044 Asynchronous Turret Movement⁽³⁾

Normally, a tool storage turret is not moved asynchronously in the process. However if an application would require this, this parameter could be set to 'yes'.

Bxx.057 Activating Tool Correction for Turret^{(4),(5)}

This parameter is only displayed if the following parameters have been set to the following values:

A00.052 Tool Management = Yes

Bxx.014 Tool Management = Yes

Bxx.015 Tool Management = Turret

With NC and PLC controlled turrets that index asynchronously to NC-program execution (Bxx.044 'Asynchronous Turret Movement'='Yes', and the related REV_SYNC standard function is inactive), this parameter permits tool compensation to be activated either at the beginning of the index movement (with first movement of the axis following a tool storage movement command 'MTP' or 'MMP') or at the end of the index movement (with the first movement of the axis following 'MRY').

NOTE: *Due to the index movement of the tool axis that is synchronous with NC-program execution, this function remains without effect for combined spindle/turret axes (dual function spindle and tool axis).*

7.6 NC Tool Management Control Functions

The MT-CNC provides NC-commands, SPS FUNCTIONS, and FUNCTION_BLOCKS to influence the tool management. They are used to influence the tool correction and life data mechanisms as well as the tool movements within the machine.

7.6.1 Requirements**7.6.1.1 Defined Plane**

Proper usage of tool correction requires a Cartesian coordinate system. At least one of the three main axis must exist physically.

Depending on the application,

- One main axis (i.e., drill unit)
- Two main axes (i.e., turning unit), or
- Three main axes (i.e., milling unit),

can exist physically.

The XY (G17), ZX (G18) and YZ (G19) plane can be selected independently of the available main axes.

The tool length correction "Length L3" always works perpendicular to the selected plane. The length correction, "Length L1", "Length L2", tool nose and mill nose radius compensation, on the other hand work always in the selected plane.

7.6.1.2 Tool Selection

Syntax: T<x> :x = (0...9999999), only (0...999) when using MMP
:x = Mathematical expression

The T-word is used to select the next tool or tool location. The NC-command must be used to select the desired tool spindle if more than one spindle is operated by the process. The positioning of the tool storage and the tool transfer itself are initiated via special NC-commands.

The tool corrections are deactivated after programming "T0" (no tool) and performing the tool transfer. The T-word contains "0" after MT-CNC power on which means "T0" is selected.

Each tool in used in the NC-program must be defined in the tool Setup List for the process or program. A process error occurs (No acceptable tool found) if the tool is not defined but programmed. The tool number is assigned to the corresponding tool in the Active Tool List at NC-program start. A selected tool number (T) stays active until a different tool number is programmed. This allows the calling of tools in the NC-program and use of tool change subroutines.

The T-word function depends on the type of tool storage device:

- Tool storage is of the type **Tool Magazine:**
Used in conjunction with a NC-command MTP and MMP, the T-word results in the positioning of the selected tool in the defined tool change position. Corresponding tool data is provided with the **next tool change NC-command** TSM, TSM, and TCH.

An empty magazine pocket is moved into tool change position if "T0" is programmed and the NC-command MTP is executed. The tool data of the currently active tool becomes inactive after executing the next tool change NC-command.
- Tool storage is of the type **Tool Turret:**
The corresponding tool is moved into position if a T-word is programmed and the NC-command MTP and MMP is executed. The tool's data becomes active **at the same time**.

The NC-command MTP and "T0" deactivate the tool data that is currently in machining position. No turret movement occurs.
- Tool storage is of the type **Tool Spindle** only:
The T-word and NC-command E activates the data of the tool that resides in the selected tool spindle if no tool magazine or turret exists and the tool management is active. "T0" can be used to deactivate the tool's data. No other tool change NC-commands are required.

7.6.1.3 Tool Edge Selection (E)

Syntax: E<x> :x = (1...9), (defined in system parameter Axx.054)

The NC-command E selects the defined tool Edge of the currently active tool if a tool magazine or turret exists. It automatically activates the data of the tool in the currently selected tool spindle if a tool spindle exists.

The Edge number 1 (default) simplifies the use of most tools, since no tool Edge must be selected via the NC-program.

The tool Edge number 1 is selected and its data is provided:

- at each tool transfer into the spindle (magazine),
- when rotating the tool into machining position (turret), and
- if the T-word is used to activate the tool in the selected tool spindle, if no tool magazine or turret exists.

The life and wear data of the active tool is updated whenever there is a NC-command E executed in the NC-program. This allows to monitor tool life and wear data during long machining cycles with the same tool.

- 2: move sel. location to change pos. 2
- 3: move sel. location to change pos. 3
- 4: move sel. location to change pos. 4

<direction>=0: shortest distance (any direction)

- 1: positive direction
- 2: negative direction
- 3: longest distance (any direction)^{(4),(5)}

Chip removal can be performed via direction option '3' when using NC-command MTP. When this MTP is used with without an activated T-number, the turret performs a 360° rotation.

The NC-command MTP will cause the tool storage (turret/magazine) to move the location (pocket/position) that contains the selected tool (T-word) to the specified tool change position. Usually, MTP is programmed together with the tool selection(T-word) right after changing the currently active tool. The next tool is then moved into the specified change position while machining with the current tool.

MMP

Move Location into Position

Syntax: MMP,

MMP(<position>) or,

MMP(<position>,<direction>)

Option: <position>, <constant>, <NC-variable>: (0,1,2,3,4)
<direction>, <constant>, <NC-variable>: (0,1,2)

Explanation:<position> = 1: move sel. location to change pos. 1
2: move sel. location to change pos. 2
3: move sel. location to change pos. 3
4: move sel. location to change pos. 4

<direction>=0: shortest distance (any direction)

- 1: positive direction
- 2: negative direction
- 3: longest distance (any direction)^{(4),(5)}

Chip removal can be performed via direction option '3' when using NC-command MMP. When this MMP is used with without an activated T-number, the turret performs a 360° rotation.

The NC-command MMP will cause the tool storage (turret/magazine) to move the **location** (magazine pocket / turret position, **not tool number**) that is programmed via the T-word to the specified tool change position.

NOTE: *The constants of the T-word is used as the location and not the tool (T) number, if the NC-command MMP is executed. Tools of tool families are used by MMP only if their location number has been programmed previously in the T-word (value range (0...999)).*

MFP

Move Free Pocket into Position

Syntax: MFP,

MFP(<position>) or,

MFP(<position>,<direction>)

Option: <position>, <constant>, <NC-variable>: (0,1,2,3,4)
<direction>, <constant>, <NC-variable>: (0,1,2)

Explanation:<position> = 1: move next free loc. to chg. pos. 1

- 2: move next free loc. to chg. pos. 2
- 3: move next free loc. to chg. pos. 3
- 4: move next free loc. to chg. pos. 4

<direction>=0: any direction (shortest dist.)

- 1: positive direction
- 2: negative direction

The NC-command MFP causes a tool storage (magazine) movement that will bring the closest and empty location (pocket) into the specified tool change position. This command is used to put the tool that is currently in the gripper or spindle back into the next empty magazine pocket.

MOP

Move Old Pocket into Position

Syntax: MOP,

MOP(<position>) or,

MOP(<position>,<direction>)

Option: <position>, <constant>, <NC-variable>: (0,1,2,3,4)

<direction>, <constant>, <NC-variable>: (0,1,2)

<tool Spdl>, <constant>, <NC-variable>: (1,2,3)

Explanation:<position> = 1: move next free loc. to chg. pos. 1

2: move next free loc. to chg. pos. 2

3: move next free loc. to chg. pos. 3

4: move next free loc. to chg. pos. 4

<direction>=0: any direction (shortest dist.)

1: positive direction

2: negative direction

<tool spdl>= 1: old loc. for tool in spdl. 1 (S,S1)

2: old loc. for tool in spdl. 2 (S2)

3: old loc. for tool in spdl. 3 (S3)

The NC-command MOP will cause the tool storage (magazine) to move the corresponding location (pocket) into the specified tool change position from which the active tool (currently in the specified spindle) was removed. This NC-command makes it possible to maintain the tool order in tool magazines.

MRY

Tool Storage Ready

Syntax: MRY

This NC-command stops the NC-program execution until the active tool storage movement is completed. It influences the NC-program execution only. NC-program execution and tool storage movements (asynchronous) can be synchronized using this NC-command.

MEN

Tool Storage Enable for Manual Mode

The NC-command MEN enables the manual tool storage mode. The SPS can read the status of the Gateway signal PxxS.MGMAN and select the manual tool storage mode via Gateway signal PxxC.MGMAN.

7.6.3 Tool Change NC-commands

Tool Change NC-commands are only required if the tool storage device is a magazine and the tools must be exchanged with a tool spindle and vice versa. The transfer may be accomplished directly between tool storage and tool spindle or via grippers. Tool change NC-commands are not required if the tool storage is defined in the process parameter as a turret.

TCH Perform a Complete Tool Change

Syntax: TCH,
 TCH(<position>) or,
 TCH(<position>,<tool spdl>)
 Option: <position>, <constant>, <NC-variable>: (0,1,2,3,4)
 <tool spdl>, <constant>, <NC-variable>: (1,2,3)

Explanation:<position> = 1: exchange tool in change position 1
 2: exchange tool in change position 2
 3: exchange tool in change position 3
 4: exchange tool in change position 4

<tool spdl>= 1: with tool in spindle 1 (S, S1)
 2: with tool in spindle 2 (S2)
 3: with tool in spindle 3 (S3)

The NC-command TCH initiates the exchange of the tool that is located in the selected tool spindle with the tool that is in the selected tool change position. This requires that the pocket containing the selected tool must be moved into the selected tool change position prior to the NC-command TCH. The tool transfer itself is controlled by the (SPS). The NC-command TCH is mostly used in the conjunction with tool changer mechanisms that have two or more grippers.

TMS Tool Change from Magazine to Spindle

Syntax: TMS,
 TMS(<position>) or,
 TMS(<position>,<tool spdl>)
 Option: <position>, <constant>, <NC-variable>: (0,1,2,3,4)
 <tool spdl>, <constant>, <NC-variable>: (1,2,3)

Explanation:<position> = 1: change tool in change position 1
 2: change tool in change position 2
 3: change tool in change position 3
 4: change tool in change position 4

<tool spdl>= 1: to spindle 1 (S, S1)
 2: to spindle 2 (S2)
 3: to spindle 3 (S3)

The NC-command TMS initiates the transfer of the tool, that is in the selected change position, into the selected tool spindle. This requires that the pocket containing the selected tool must be moved into the selected tool change position prior to the NC-command TMS and that the selected tool spindle does not contain a tool.

TSM Tool Change from Spindle to Magazine

Syntax: TSM,
 TSM(<position>) or,
 TSM(<position>,<tool spdl>)
 Option: <position>, <constant>, <NC-variable>: (0,1,2,3,4)
 <tool spdl>, <constant>, <NC-variable>: (1,2,3)

Explanation:<position> = 1: change tool in change position 1
 2: change tool in change position 2
 3: change tool in change position 3

4: change tool in change position 4

<tool spdl>= 1: to spindle 1 (S, S1)
2: to spindle 2 (S2)
3: to spindle 3 (S3)

The NC-command TSM initiates the transfer of the tool that is located in the specified tool spindle to the empty magazine pocket in the selected tool change position. The old location, the one that contained the active tool, or next available empty pocket must be moved into the selected tool change position prior to the NC-command TSM and the selected tool spindle must contain a tool. The tool transfer itself is controlled by the (SPS).

- BSE** **Branch if Spindle Empty**
Syntax: BSE_label
A branch to the NC-block, starting with the defined label is performed if the tool “spindle 1” (S, S1) is empty (contains no tool).
- BTE** **Branch if Tool T0 Selected**
Syntax: BTE_label :refer to NC-labels section
A branch to the NC-block starting with the defined label is performed if the tool T0 (no tool) has been programmed and the NC-command BTE is executed. T0 indicates that the tool that is in the currently selected spindle should be removed and put back into a tool storage location.
- TPE** **Tool Pocket Empty**
Syntax: TPE
Checks whether the tool storage location (pocket), currently in the tool change **position “1”**, is empty or not. The program execution continues if the location in change **position “1”**, is empty. NC-program execution is stopped with a process error message displayed if the location at tool change **position “1”**, is not empty.
- TSE** **Tool Spindle Empty**
Syntax: TSE
The NC-command TSE checks whether there is a tool in the **spindle “1”** (S, S1) or not. If there is no tool in the **spindle “1”**, NC-program execution continues. A tool in the **spindle “1”** will cause a process error with message and interrupt NC-program execution.

TLD**Tool Data Read / Write**

Syntax: TLD ([<process>], [<address>], <s / t>, <l / i>, [<edge>], [<element>], [<status>])

<process>	(none, 0...6)	none: (0...6): Process in which TLD is used
<address>	(none, 0,1)	none: The next two parameters are interpreted as “tool storage” and “location”. 0: The next two parameters are interpreted as “tool storage” and “location”. 1: The next two parameters are interpreted as “T-number” and tool index number”.
<s / t>	(0,1,2) if <address>=0 or 1	0: Tool storage (magazine/turret) 1: Spindle 2: Gripper (1...999999) if <address>=2 (1...999999): Tool number (T-number)
<l / i>	(1...999) if <address>=0 or 1 and <s / t>=0 (1,2,3) if <address>=0 or 1 and <s / t>=1	(1...999): Magazine pocket, turret pos. 1: Tool spindle “S” or “S1” 2: Tool spindle “S2” 3: Tool spindle “S3”
	(1...4) if <address>=0 or 1 and <s / t>=2	(1...4): Gripper number
	(none, 1...9999) if <address>=2	none: Uses index number of the tool family’s (selected via T- number) primary tool (1...9999): Tool of tool family with this tool index number
<edge>	(none, 0...9)	none: Basic tool data 0: Basic tool data (1...9): Data of tool edge number (1...9)
<element>	(3...26) if <edge>=0 (1...35) if <edge>=(1...9)	(3...26): Basic tool data element (3...26) (1...35): Tool Edge data element (1...35)
<status>	(1...32) if <edge>=0 (1...16) if <edge>=(1...16)	(1...32): Basic tool status bit (1...32) (1...16): Tool Edge status bit (1...16)

The NC-command can be TLD can be used to read tool Edge data from and write (limited) it to the Active Tool List of the process. NC-variables can be used instead of constants, however no mathematical expressions can be used for the parameters (<...>) of the NC-command TLD.

Optional NC-command parameters are identified by square brackets “[” and “]” and must not be programmed. This requires the programming of the separator “,” at all times even though the parameter is not programmed.

7.7 SPS Tool Management Control Functions

The programmable controller (SPS) program must not contain additional logic, other than the FUNCTIONs alone, to move the tool storage if the tool storage movement is NC-controlled via a servo axis. Only the tool storage / change Gateway signals must be implemented. SPS controlled tool storage's require additional logic to control the reference and move sequences of the tool storage.

7.7.1 Standard Data Types

TLBD

Standard data type for Basic Tool Data^{(4),(5)}

```
STRUCT TLBD (*Basic tool data*)
  TOOL_NAME: CHAR28; (*E0.D02: Tool Name (ID)*)
  T_NR: DINT; (*E0.D05: T-number*)
  INDEX_NR: INT; (*E0.D06: Index number*)
  CORR_TYP: USINT; (*E0.D07: Correction Type*)
  EDGES: USINT; (*E0.D08: Amount of Edges*)
  STATUS: DWORD; (*E0.D09: Tool Status*)
  HALF_POCK: USINT; (*E0.D10: half pocket overlap*)
  F1: USINT; (*not used*)
  OLD_PLACE: INT; (*E0.D11: old pocket*)
  TIME_UNIT: USINT; (*E0.D16: time unit*)
  LEN_UNIT: USINT; (*E0.D17: length unit*)
  CODE: USINT; (*E0.D18: Function / feature*)
  F2: USINT; (*not used*)
  DISPLTYP: INT; (*E0.D19: Display type, tool*)
  USERDAT1: REAL; (*E0.D20: User tool data 1*)
  USERDAT2: REAL; (*E0.D21: User tool data 2*)
  USERDAT3: REAL; (*E0.D22: User tool data 3*)
  USERDAT4: REAL; (*E0.D23: User tool data 4*)
  USERDAT5: REAL; (*E0.D24: User tool data 5*)
  USERDAT6: REAL; (*E0.D25: User tool data 6*)
  USERDAT7: REAL; (*E0.D26: User tool data 7*)
  USERDAT8: REAL; (*E0.D27: User tool data 8*)
  USERDAT9: REAL; (*E0.D28: User tool data 9*)
END_STRUCT
```

TLED

Standard data type for Tool Edge Data^{(4),(5)}

```
STRUCT TLED (*Edge tool data*)
  ORIENT: USINT; (*En.D01: Edge Orientation*)
  STATUS: WORD; (*En.D02: Edge Status*)
  F1: USINT; (*not used*)
  REM_LIFE: REAL; (*En.D04: Remain. Tool Life*)
  WARN_LIM: REAL; (*En.D04: Warning Limit*)
  MAX_LIFE: REAL; (*En.D05: Max. Tool Life*)
  LENGTH_L1: DINT; (*En.D07: Length L1*)
  LENGTH_L2: DINT; (*En.D08: Length L2*)
  LENGTH_L3: DINT; (*En.D09: Length L3*)
  RADIUS_R: DINT; (*En.D10: Radius R*)
  WEAR_L1: DINT; (*En.D11: Wear L1*)
  WEAR_L2: DINT; (*En.D12: Wear L2*)
  WEAR_L3: DINT; (*En.D13: Wear L3*)
  WEAR_R: DINT; (*En.D14: Wear R*)
  OFFSET_L1: DINT; (*En.D15: Offset L1*)
```

```

OFFSET_L2: DINT; (*En.D16: Offset L2*)
OFFSET_L3: DINT; (*En.D17: Offset L3*)
OFFSET_R: DINT; (*En.D18: Offset R*)
WEARFCTL1: DINT; (*En.D27: Wear Factor L1*)
WEARFCTL2: DINT; (*En.D28: Wear Factor L2*)
WEARFCTL3: DINT; (*En.D29: Wear Factor L3*)
WEARFCTR: DINT; (*En.D30: Wear Factor R*)
USERDAT1: REAL; (*En.D31: User Edge Data 1*)
USERDAT2: REAL; (*En.D32: User Edge Data 2*)
USERDAT3: REAL; (*En.D33: User Edge Data 3*)
USERDAT4: REAL; (*En.D34: User Edge Data 4*)
USERDAT5: REAL; (*En.D35: User Edge Data 5*)
USERDAT6: DINT; (*En.D36: User Edge Data 6*)
USERDAT7: DINT; (*En.D37: User Edge Data 7*)
USERDAT8: DINT; (*En.D38: User Edge Data 8*)
USERDAT9: DINT; (*En.D39: User Edge Data 9*)
USERDAT10: DINT; (*En.D40: User Edge Data 10*)
END_STRUCT
    
```

7.7.2 Tool Storage Reference

MRF

Reference Tool Storage

Is used in the (SPS) program to recognize the NC-command MRF that starts the reference sequence of the tool storage. The FUNCTION MRF_Q must be used to acknowledge the completion of the reference sequence.

Figure 7-1: Read Tool Storage NC-command MRF

Basic Structure	ACTIVE	Result
	0: MRF FUNCTION not active 1: MRF FUNCTION active	0: NC-command MRF not active or Active = 0 1: NC-command MRF active

PROC = Process number (0...6)

MRF_Q

Acknowledge Tool Storage at Reference

Is used in the programmable controller (SPS) program to acknowledge the completion of the tool storage reference sequence. The CNC stops NC-program execution until it receives this acknowledgment if the NC-command MRF_Q is programmed after the NC-command MRF.

Figure 7-2: Acknowledge Tool Storage NC-command MRF

Basic Structure	QUIT	Result
	0: MRF_Q FUNCTION not active 1: Acknowledges comp. of the reference sequence	0: MRF_Q FUNCTION not active 1: Acknowledges comp. of the reference sequence

PROC = Process number (0...6)

7.7.3 Position Tool Storage

The tool storage must move the pocket with the selected tool and empty pockets into the tool change position in order to use different tools during a NC-program cycle.

MMV

Move Tool Storage

Used in the programmable controller (SPS) program to recognize tool storage move NC-commands (MHP, MMP, MTP, MFP, MOP) that starts the reference sequence of the tool storage. The FUNCTION MMV_Q must be used to acknowledge the completion of the tool storage movement.

Figure 7-3: Read Tool Storage NC-command MMV

Basic Structure	ACTIVE	Result
	0: MMV FUNCTION not active 1: MMV FUNCTION active	0: NC-command MMV not active or Active = 0 1: NC-command MMV active

PROC = Process number (0...6)

MMV_Q

Acknowledges Tool Storage Move

Is used in the (SPS) program to acknowledge the completion of the initiated (NC-commands MHP, MMP, MTP, MFP, MOP) tool storage move. The CNC stops NC-program execution until it receives this acknowledgment if the NC-command MHP, MMP, MTP, MFP, or MOP is active.

Figure 7-4: Acknowledge Tool Storage NC-command MMV

Basic Structure	QUIT	Result
	0: MMV_Q FUNCTION not active 1: Acknowledges comp. of the tool storage move.	0: MMV_Q FUNCTION not active 1: Acknowledges comp. of the tool storage move.

PROC = Process number (0...6)

7.7.4 Standard Tool Change FUNCTIONS

Tool change FUNCTIONS are required only if the tool storage is of the type magazine and if tools must be exchanged between spindle, magazine, and eventually grippers.

All tool change sequences are initiated via the tool change NC-commands and NC-block execution is halted. The tool change is then recognized and performed by the (SPS). Completed tool change sequences are acknowledged to the CNC which allows the CNC to continue NC-block execution.

TCH

Complete Tool Change

The status of the NC-command TCH must be read in the SPS program via the FUNCTION TCH. The NC-command TCH initiates in the SPS program the user tool change sequencer. The NC-program execution stops until the NC-command TCH is acknowledged in the SPS program via FUNCTION TCH_Q after completion of the tool transfer sequence.

Figure 7-5: Read Tool Change NC-command TCH

Basic Structure	ACTIVE	Result
	0: TCH FUNCTION not active 1: TCH FUNCTION active	0: NC-command TCH not active or Active = 0 1: NC-command TCH active

PROC = Process number (0...6)

TCH_Q

Complete Tool Change Acknowledge

The FUNCTION TCH_Q must be used in the SPS to acknowledge the completion of the tool change that was initiated via the NC-command TCH.

Figure 7-6: Acknowledge Tool Change Command TCH

Basic Structure	QUIT	Result
	0: TCH_Q FUNCTION not active 1: Acknowledges comp. of the complete tool change.	0: TCH_Q FUNCTION not active 1: Acknowledges comp. of the complete tool tool change

PROC = Process number (0...6)

7.7.5 Tool Change Between Magazine and Spindle

Beside the general tool change NC-command TCH, more specific tool change NC-commands can be used. By using these commands (TMS, TSM) the NC-command can contain smaller tool change steps, i.e. clamp / unclamp, via auxiliary M-functions, motion NC-blocks to the magazine and spindle orientation to unlock or lock the tool in the magazine pocket.

TMS

Tool from Magazine to Spindle

This NC-command initiates tool change sequence for a tool transfer from the specified tool change position to the specified tool spindle. The NC-program execution stops until the NC-command TMS is acknowledged in the SPS program via FUNCTION TMS_Q after completion of the tool transfer sequence.

Figure 7-7: Read Tool Change NC-command TMS

Basic Structure	ACTIVE	Result
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> TMS ACTIVE PROC POS SPINDLE </div> Bool — INT — INT — INT —	0: TMS FUNCTION not active 1: TMS FUNCTION active	0: NC-command TMS not active or Active = 0 1: NC-command TMS Active

PROC: Process number (0...6)

POS: Tool Change Position (1...4)

SPINDLE: 1 = Tool Spindle 1 (S, S1)

SPINDLE: 2 = Tool Spindle 2 (S2)

SPINDLE: 3 = Tool Spindle 3 (S3)

TMS_Q

Acknowledge Tool from Magazine to Spindle

This FUNCTION must be used in the SPS program to acknowledge the tool change from the magazine to the spindle, that was initiated via the NC-command TMS.

Figure 7-8: Acknowledge Tool from Magazine to Spindle

Basic Structure	QUIT	Result
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> TMS_Q QUIT PROC POS SPINDLE </div> Bool — INT — INT — INT —	0: TMS_Q FUNCTION not active 1: Acknowledges tool change from the magazine to spindle	0: TMS_Q FUNCTION not active 1: Acknowledges tool change from the magazine to spindle

PROC: Process number (0...6)

POS: Tool Change Position (1...4)

SPINDLE: 1 = Tool Spindle 1 (S, S1)

SPINDLE: 2 = Tool Spindle 2 (S2)

SPINDLE: 3 = Tool Spindle 3 (S3)

TSM

Read Tool Change NC-command TSM

This command initiates in the SPS program the tool change sequence for a tool transfer from the specified tool spindle to the specified tool change position. NC-program execution stops until the NC-command TSM is acknowledged in the SPS program via FUNCTION TSM_Q after completion of the tool transfer.

Figure 7-9: Read Tool Change NC command TSM

Basic Structure	ACTIVE	Result
	0: TSM FUNCTION not active 1: TSM FUNCTION active	0: NC-command TSM not active or Active = 0 1: NC-command TSM Active

PROC: Process number (0...6)
 POS: Tool Change Position (1...4)
 SPINDLE: 1 = Tool Spindle 1 (S, S1)
 SPINDLE: 2 = Tool Spindle 2 (S2)
 SPINDLE: 3 = Tool Spindle 3 (S3)

TSM_Q

Acknowledge Tool from Spindle to Magazine

This function must be used in the SPS to acknowledge the tool change from the spindle to the magazine, that was initiated via the NC-command TSM.

Figure 7-10: Acknowledge Tool Change NC-command TSM

Basic Structure	QUIT	Result
	0: TSM_Q FUNCTION not active 1: Acknowledges tool change from the spindle to magazine	0: TSM_Q FUNCTION not active 1: Acknowledges tool change from the spindle to magazine

PROC: Process number (0...6)
 POS: Tool Change Position (1...4)
 SPINDLE: 1 = Tool Spindle 1 (S, S1)
 SPINDLE: 2 = Tool Spindle 2 (S2)
 SPINDLE: 3 = Tool Spindle 3 (S3)

7.7.6 Tool Transfer FUNCTIONS

Tool Transfer FUNCTIONS are used in the SPS program to perform the logical tool transfer in the tool management while simultaneously and sequentially performing the physical tool change.

XMS

Magazine to Spindle Tool Transfer Request

The SPS uses this function to request a logical tool transfer from the magazine pocket in the specified tool change position to the specified tool spindle.

Figure 7-11: Magazine to Spindle Transfer Request

Basic Structure	INIT	Result
	0: XSM FUNCTION not active 1: XSM FUNCTION active	0: NC-command XSM not active or Active = 0 1: Tool transfer request magazine to spindle

PROC: Process number (0...6)
 POS: Tool Change Position (1...4)
 SPINDLE: 1 = Tool Spindle 1 (S, S1)
 SPINDLE: 2 = Tool Spindle 2 (S2)
 SPINDLE: 3 = Tool Spindle 3 (S3)

XMS_PA

Magazine to Spindle Tool Transfer Positive Ack.

A positive acknowledgement of a tool transfer XMS is recognized via a status of “1” at the FUNCTION XMS_PA. The tool management logically transfers the tool in this case from the pocket in the tool change position to the spindle.

Figure 7-12: Magazine to Spindle Tool Transfer Positive Ack.

Basic Structure	ACTIVE	Result
	0: Not reading ack. status 1: Read positive ack. status	0: No positive ack. or Active = 0 1: Transfer positive ack. (allowed)

PROC: Process number (0...6)
 POS: Tool Change Position (1...4)
 SPINDLE: 1 = Tool Spindle 1 (S, S1)
 SPINDLE: 2 = Tool Spindle 2 (S2)
 SPINDLE: 3 = Tool Spindle 3 (S3)

XMS_NA Magazine to Spindle Tool Transfer Negative Ack.
 A negative acknowledge of a tool transfer request XMS is recognized via a status of “1” at the FUNCTION XMS_NA. The tool management in this case prevents a logical tool transfer.

Figure 7-13: Magazine to Spindle Tool Transfer Negative Ack.

Basic Structure	ACTIVE	Result
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> Bool — INT — INT — INT — </div> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> XMS_NA ACTIVE PROC POS SPINDLE </div> <div style="margin-left: 10px;"> —Result </div> </div>	0: Not reading ack. status 1: Read negative ack. status	0: No negative ack. or Active = 0 1: Transfer negative ack. (not allowed)

PROC: Process number (0...6)
 POS: Tool Change Position (1...4)
 SPINDLE: 1 = Tool Spindle 1 (S, S1)
 SPINDLE: 2 = Tool Spindle 2 (S2)
 SPINDLE: 3 = Tool Spindle 3 (S3)

XMS_Q Magazine to Spindle Tool Transfer Ack.
 A positive tool transfer acknowledge of a tool transfer request is recognized via a status of “1” at the FUNCTION XMS_PA. If such a case a status of “1” is applied to the FUNCTION XMS_Q, logical tool transfer in the tool management, from the pocket in the tool change position to the spindle, is performed.

NOTE: *The tool in the pocket that is in the tool change position will be logically moved by tool management to the spindle! The tool must be physically moved from the magazine to the spindle when it is moved in the tool management to prevent damage to machine!*

Figure 7-14: Magazine to Spindle Tool Transfer Ack.

Basic Structure	QUIT	Result
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> Bool — INT — INT — INT — </div> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> XSM_Q QUIT PROC POS SPINDLE </div> <div style="margin-left: 10px;"> —Result </div> </div>	0: XSM_Q FUNCTION not active 1: Acknowledges , physical tool xfer magazine to spindle performed, perform logical xfer	0: XSM_Q FUNCTION not active 1: Acknowledges , physical tool xfer magazine to spindle performed, perform logical xfer

PROC: Process number (0...6)
 POS: Tool Change Position (1...4)
 SPINDLE: 1 = Tool Spindle 1 (S, S1)
 SPINDLE: 2 = Tool Spindle 2 (S2)
 SPINDLE: 3 = Tool Spindle 3 (S3)

XMS_CA Magazine to Spindle Tool Transfer Cancel

This function can be used to cancel a tool transfer that was requested via the FUNCTION XMS if the transfer is not allowed by tool management. A negative acknowledge of a tool transfer request XMS is recognized via a status of "1" at the FUNCTION XMS_NA. In this case tool management prevents a logical tool transfer.

Figure 7-15: Magazine to Spindle Tool Transfer Cancel

Basic Structure	CANCEL	Result
	0: XSM_CA FUNCTION not active 1: Cancel active tool transfer request XMS	0: XSM_CA FUNCTION not active 1: Cancel active tool transfer request XMS

PROC: Process number (0...6)
 POS: Tool Change Position (1...4)
 SPINDLE: 1 = Tool Spindle 1 (S, S1)
 SPINDLE: 2 = Tool Spindle 2 (S2)
 SPINDLE: 3 = Tool Spindle 3 (S3)

XFER_CHK Disable Tool Transfer Error Check

The CNC normally activates the process error #320 "Tool transfer instruction without tool change command" if a tool transfer request is initiated in the SPS program and no tool change NC-command is active. To prevent this error when performing tool transfers via the SPS program only, the status of the FUNCTION XFER_CHK can be set to "1".

Figure 7-16: Disable Tool Transfer Error Check

Basic Structure	QUIT	Result
	0: XFER_CHK FUNCTION not active 1: Disable error check for active tool change NC-command request XMS	0: XFER_CHK FUNCTION not active 1: Disable error check for active tool change NC-command request XMS

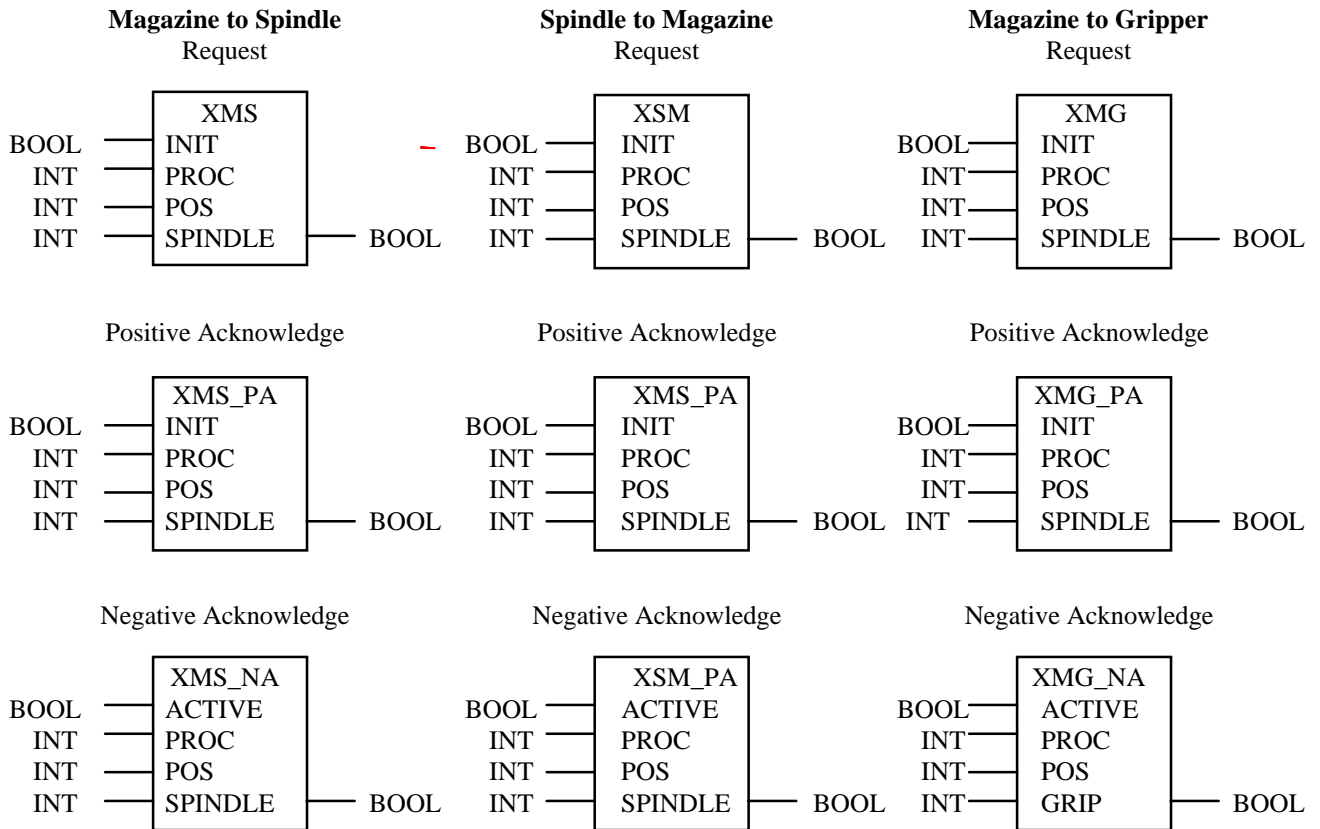
PROC = Process number (0...6)

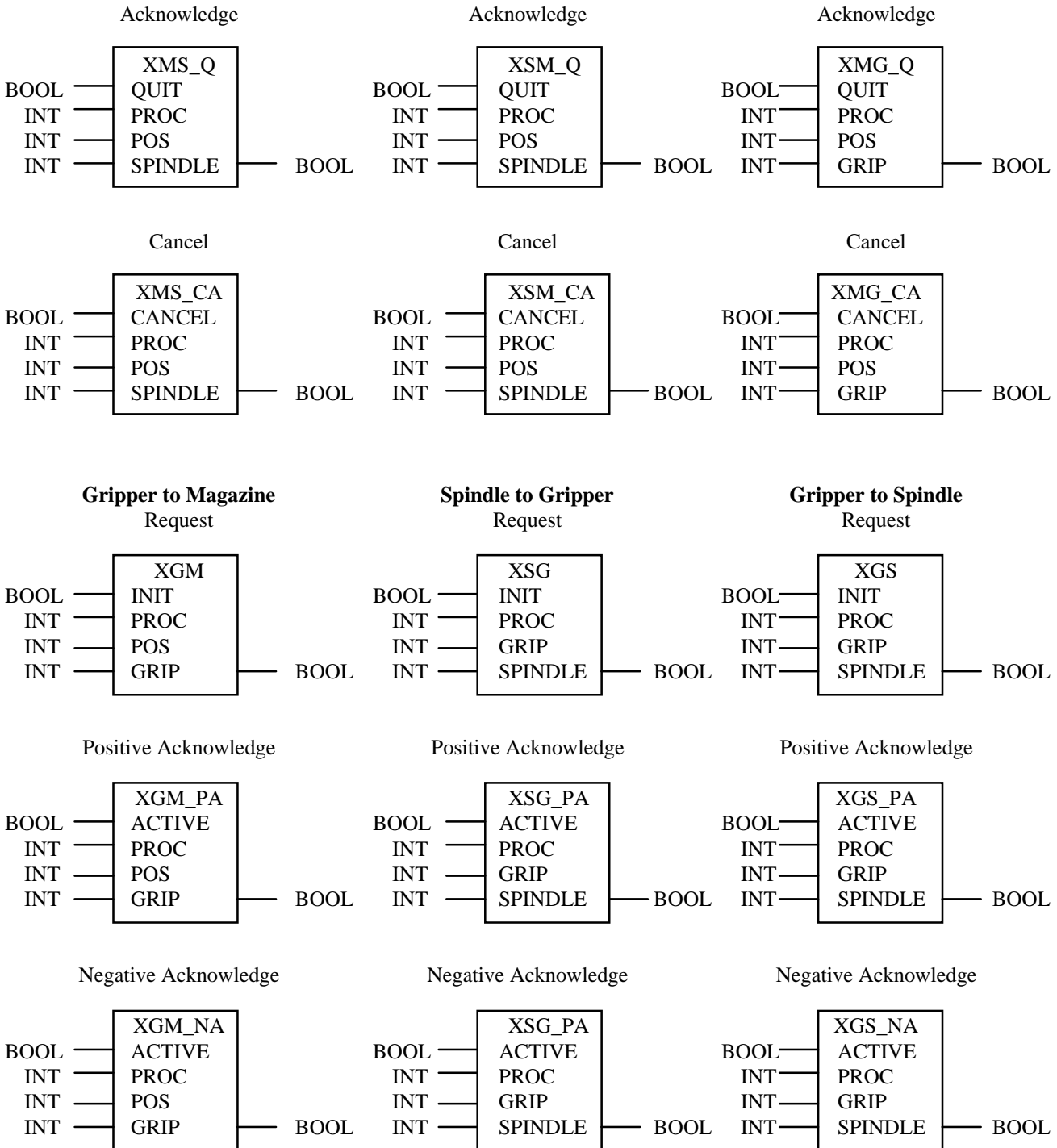
7.7.7 Tool Transfer FUNCTION Overview

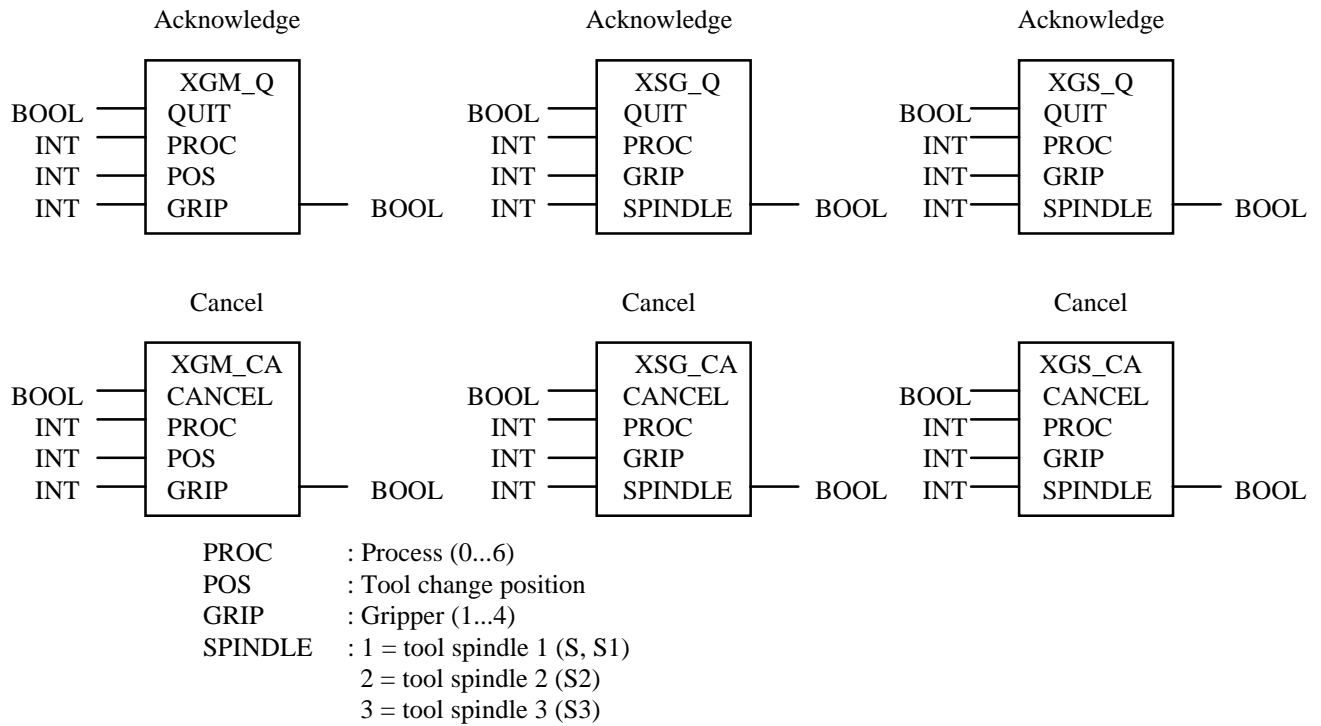
The operational sequence for tool transfers between,

- Magazine and Gripper
- Spindle and Magazine
- Spindle and Gripper
- Gripper and Magazine
- Gripper and Spindle

are identical to the one between the magazine and the spindle using the corresponding FUNCTIONS.







7.7.8 Combined Spindle / Turret Axis

The MT-CNC system parameters allow to specify a combined spindle turret axis. Such an axis type allows to use one drive to move the tool turret and spindle for live tooling. The CNC automatically switches between turret and spindle mode, depending on the executed NC-command.

MAG_ACT Read Turret Mode

The CNC selects turret mode if a tool storage movement NC-command is programmed and a combined spindle / turret axis is defined for the process.

Figure 7-17: Read Turret Mode

Basic Structure	ACTIVE	Result
	0: MAG_ACT FUNCTION not active 1: Read turret mode status	0: MAG_ACT FUNCTION not active 1: Read turret mode status

PROC = Process number (0...6)

MAG_Q Acknowledge the Turret Mode

Once the combined spindle / turret axis is mechanically coupled to move the turret, the standard FUNCTION MAG_Q must be used to acknowledge this status to the CNC. This acknowledgment must remain active while the turret mode is active, otherwise a process error, #321 “Comb. Spindle/Turret axis acknowledgment missing”, is activated.

Figure 7-18: Acknowledge Turret Mode

Basic Structure		QUIT	Result
Bool	MAG_Q QUIT	0: MAG_Q FUNCTION not active	0: MAG_Q FUNCTION not active
INT	PROC	1: Acknowledges turret mode	1: Acknowledges turret mode

PROC = Process number (0...6)

SPDL_ACT

Read Spindle Mode

The CNC selects the spindle mode if NO tool storage movement NC-command is programmed and a combined spindle / turret axis is defined for the process. The process then automatically has a turret tool storage driven by a servo or spindle drive (depending on their capabilities) that can also be used as a spindle for live tooling.

Figure 7-19: Read Spindle Mode

Basic Structure		ACTIVE	Result
Bool	SPDL_ACT ACTIVE	0: SPDL_ACT FUNCTION not active	0: SPDL_ACT FUNCTION not active
INT	PROC	1: Read spindle mode status	1: Read spindle mode status

PROC = Process number (0...6)

SPDL_Q

Acknowledge Spindle Mode

Once the combined spindle / turret axis is mechanically coupled to move the tool, this FUNCTION must be used to acknowledge this status to the CNC. The acknowledgment must remain active while the spindle mode is active, otherwise a process error, #321 “ Comb. Spindle/Turret axis acknowledgment missing”, is activated.

Figure 7-20: Acknowledge Spindle Mode

Basic Structure		QUIT	Result
Bool	SPDL_Q QUIT	0: SPDL_Q FUNCTION not active	0: SPDL_Q FUNCTION not active
INT	PROC	1: Acknowledges spindle mode	1: Acknowledges spindle mode

PROC = Process number (0...6)

REV_SYNC **Asynchronous Turret Movement^{(4),(5)}**

When setting Bxx.044 'Asynchronous Turret Movement' to 'Yes', the OEM can synchronize the turret movement with the NC-block execution by activating the standard FUNCTION REV_SYNC.

Figure 7-21: Asynchronous Turret Movement

```

+-----+
|REV_SYNC|
|ACTIVE   | +-BOOL
|PROC     |
+-----+

```

Variable Description:

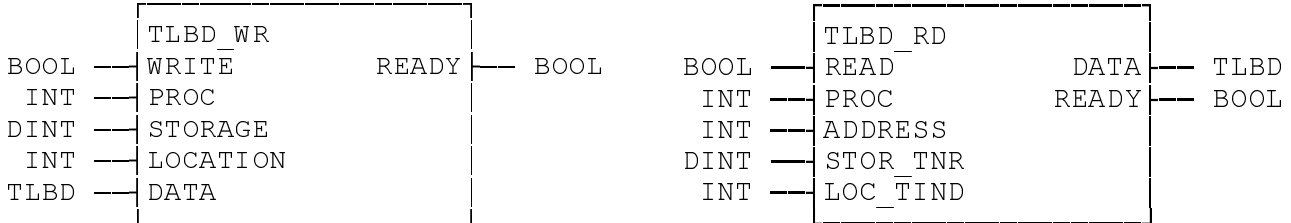
ACTIVE	'0' REV_SYNC not active
	'1' REV_SYNC active
PROC	Defines process number
BOOL	'0' REV_SYNC not active
	'1' REV_SYNC active

7.7.9 Standard FUNCTION_BLOCKS for Tool Management

The SPS provides access to the individual tool data elements and a simple way to access all tool information via the following FUNCTION_BLOCKS:

7.7.9.1 Standard FUNCTION_BLOCKS for Basic Tool Data

Figure 7-22: Standard FUNCTION_BLOCKS for Basic Tool Data^{(4),(5)}



Variable Description:

WRITE	0 ⇒ 1 transition writes basic tool data
READ	0 ⇒ 1 transition reads basic tool data
PROC	Defines process number
STORAGE	Tool storage
ADDRESS	`0` Next 2 inputs are interpreted as tool STORAGE and LOCation information
	`1` Next two inputs are interpreted as T-NumbeR and Tool INDeX number
LOCATION	Tool location number
STOR_TNR	`0` Magazine / Turret , if ADDRESS = `0`
	`1` Tool spindle , if ADDRESS = `0`
	`2` ;Tool gripper , if ADDRESS = `0`
	`1...9999999` Tool number (T-number) , if ADDRESS = `1`
TLBD	Basic tool data
LOC_TIND	`0...999` Magazine pocket / Turret position , if ADDRESS = `0`, STOR_IND = `0`
	`0..3` Tool spindle , if ADDRESS = `0`, STOR_IND = `0`
	`0..4` Tool gripper , if ADDRESS = `0`, STOR_IND = `0`
	`0..999` Tool index number , if ADDRESS = `1`
READY	Acknowledgment of Write/Read
DATA	Basic tool data

7.7.9.2 Standard FUNCTION_BLOCKS for Tool Edge Data

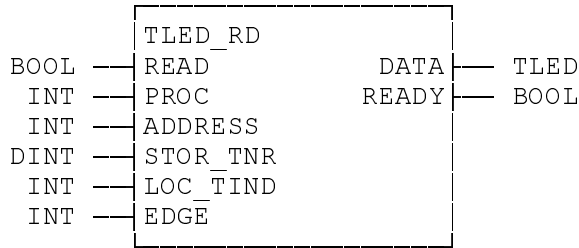
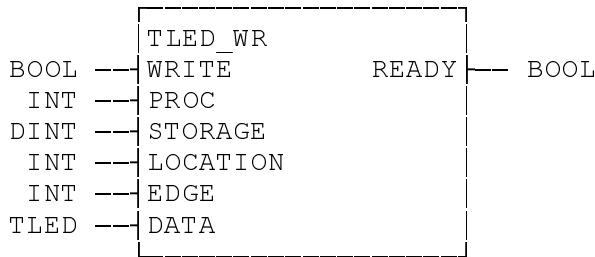


Figure 7-23: Standard FUNCTION_BLOCKS for Tool Edge Data^{(4),(5)}

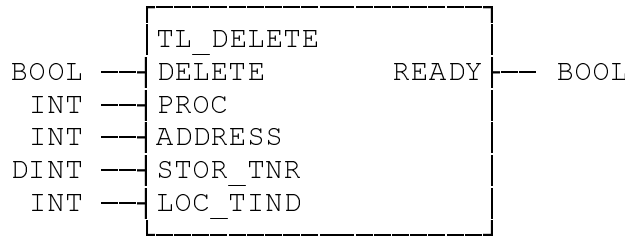


Variable Description:

WRITE	0 ⇒ 1 transition writes tool edge data
READ	0 ⇒ 1 transition reads tool edge data
PROC	Defines process number
STORAGE	Tool storage number
ADDRESS	`0` Next 2 inputs are interpreted as tool STORAge and LOCation information `1` Next two inputs are interpreted as T-NumbeR and Tool INDeX number
LOCATION	Tool location number
STOR_TNR	`0` Magazine / Turret , if ADDRESS = `0` `1` Tool spindle , if ADDRESS = `0` `2` ;Tool gripper , if ADDRESS = `0` `1...999999` Tool number (T-number) , if ADDRESS = `1`
EDGE	`0` Basic tool data `1..9` Tool edge (^0..9`) data
LOC_TIND	`0...999` Magazine pocket / Turret position , if ADDRESS = `0`, STOR_IND = `0` `0..3` Tool spindle , if ADDRESS = `0`, STOR_IND = `0` `0..4` Tool gripper , if ADDRESS = `0`, STOR_IND = `0` `0..999` Tool index number , if ADDRESS = `1`
READY	Acknowledgment of Write/Read
DATA	Tool edge data

7.7.9.3 Standard FUNCTION_BLOCKS for Tool Data

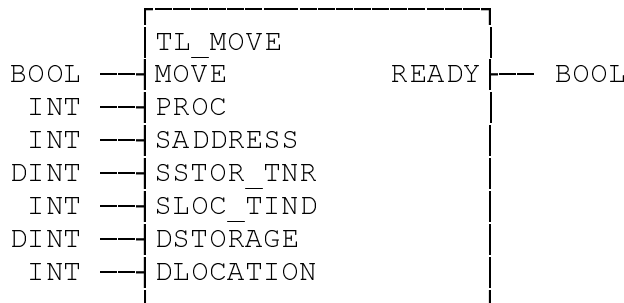
Figure 7-24: Standard FUNCTION_BLOCK to Delete a Tool^{(4),(5)}



Variable Description:

DELETE	0 ⇒ 1 transition deletes tool
PROC	Defines process number
ADDRESS	`0` Next 2 inputs are interpreted as tool STORAge and LOCation information `1` Next two inputs are interpreted as T-NumbeR and Tool INDeX number
STOR_TNR	`0` Magazine / Turret , if ADDRESS = `0` `1` Tool spindle , if ADDRESS = `0` `2` ;Tool gripper , if ADDRESS = `0` `1...999999` Tool number (T-number) , if ADDRESS = `1`
LOC_TIND	`0...999` Magazine pocket / Turret position , if ADDRESS = `0`, STOR_IND = `0` `0..3` Tool spindle , if ADDRESS = `0`, STOR_IND = `0` `0..4` Tool gripper , if ADDRESS = `0`, STOR_IND = `0` `0..999` Tool index number , if ADDRESS = `1`
READY	Acknowledgment of Tool Delete

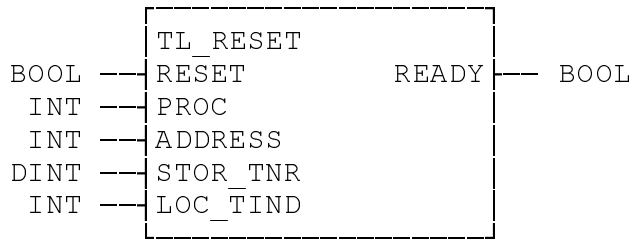
Figure 7-25: Standard FUNCTION_BLOCK to Move a Tool^{(4),(5)}



Variable Description:

MOVE	0 ⇒ 1 transition moves tool
PROC	Defines process number
SADDRESS	Addressing source tool
SSTOR_TNR	Tool storage / T-number source tool
SLOC_TIND	Tool Loc. / Tool index nr. source tool
DSTORAGE	Tool storage destination
DLOCATION	Tool location destination
READY	Acknowledgment of Tool Move

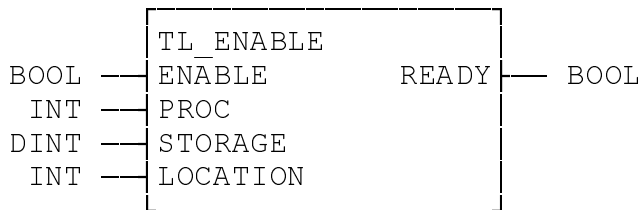
Figure 7-26: Standard FUNCTION_BLOCK to Reset a Tool^{(4),(5)}



Variable Description:

RESET	0 ⇒ 1 transition resets tool
PROC	Defines process number
ADDRESS	`0` Next 2 inputs are interpreted as tool STORAGE and LOCATION information
	`1` Next two inputs are interpreted as T-Number and Tool INDEX number
STOR_TNR	`0` Magazine / Turret , if ADDRESS = `0`
	`1` Tool spindle , if ADDRESS = `0`
	`2` ;Tool gripper , if ADDRESS = `0`
	`1...999999` Tool number (T-number) , if ADDRESS = `1`
LOC_TIND	`0...999` Magazine pocket / Turret position , if ADDRESS = `0`, STOR_IND = `0`
	`0..3` Tool spindle , if ADDRESS = `0`, STOR_IND = `0`
	`0..4` Tool gripper , if ADDRESS = `0`, STOR_IND = `0`
	`0..999` Tool index number , if ADDRESS = `1`
READY	Acknowledgment of Tool Delete

Figure 7-27: Standard FUNCTION_BLOCK to Enable Tool Data Set^{(4),(5)}

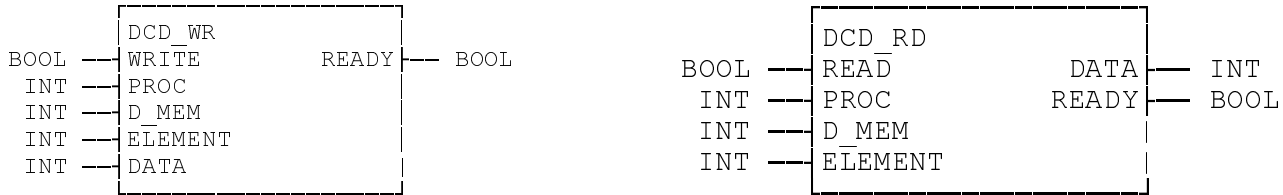


Variable Description:

ENABLE	`1` Enables tool
PROC	Defines process number
STORAGE	Tool storage number
LOCATION	Tool location number
READY	Acknowledgment of Tool Enabled

7.7.9.4 Standard FUNCTION_BLOCK for D-Corrections

Figure 7-28: Standard FUNCTION_BLOCK for D-Corrections^{(4),(5)}



Variable Description:

WRITE	0 ⇒ 1 transition writes the D-Correction
READ	0 ⇒ 1 transition reads the D-Correction
PROC	Defines process number
D_MEM	D-Correction number
ELEMENT	Element number
DATA	D-Correction value
READY	Acknowledgment of D-Correction

7.8 Tool Management Gateway Signals

The magazine control is independent from the type of tool storage axis used (SPS or NC-controlled). Tool storage movements can be initiated via NC-program, MDI mode, and the SPS program. The MT-CNC provides a Gateway control and status signal set for each process to control manual tool storage movements.

7.8.1 Tool Management Control Signals (SPS⇒CNC)

PxxC.MGENA Enable Tool Storage

“0”= Tool storage not enabled by SPS

“1”= Tool storage enabled

This signal must have a status of “1” to enable any tool storage movements. The tool storage must be enabled for its manual mode via SPS and CNC mode.

NOTE: *If the tool storage movement is SPS controlled PxxS.MGPLC=“1”, the SPS logic that causes tool storage movement must be combined with the signal PxxS.MGENA. To prevent damage and injury, the interlock must prevent any movement if the status of PxxS.MGENA is “0”!*

PxxC.MGMAN Manual Tool Storage Mode

“0”= Tool storage is controlled by CNC program

“1”= Manual (SPS) tool storage mode (request)

The tool storage can be moved manually independent of the active process operating mode, i.e. while the NC-program is executed. The NC-command (MEN) enables the manual magazine mode. The manual mode is also enabled if the NC-program execution is not active.

PxxC.MGHOM Home Tool Storage

“0”= Homing not commanded

“0 ⇒1”= Start of tool storage homing sequence

The tool management transfers the homing command to the controlling axis processor (APR) if the tool storage is servo controlled. The NC-command MRF can be read in the SPS via FUNCTION MRF. The result of FUNCTION MRF must then be assigned to PxxC.MGHOM if the tool storage is SPS controlled.

NOTE: *The tool storage home position is specified by its mechanical reference switch and cam, if the tool storage is SPS controlled. The zero marker of the encoder, reference cam, and switch specify the home position of a servo controlled tool storage axis.*

The magazine home position must not be identical with the magazine base position! The axis parameter Cxx.037 (analog axis) or digital drive parameter S-0-0052, S-0-0054 specifies the difference between reference and base position.

PxxC.MGBP Tool Storage to Base Position

“0”= Command not active

“0⇒1”= Moves the tool storage axis to its base position

The base position of a servo controlled tool storage axis must be specified in its axis parameter Bxx.037 (analog axis) and S-0-0052, S-0-0054 (digital axis). The tool storage movement is controlled by the SPS if the process parameter Bxx.020 is “0” (no NC-controlled servo tool storage). The tool management then transfers the base position (pocket #1) in PxxS.MGCP to the SPS and sets the status of the FUNCTION MMV to “1”.

PxxC.MGPOS Move Tool Storage Positive

“0”= Do not move.

“0⇒1”= Move tool storage 1 location in position dir.

A signal change from “0” to “1” causes the tool management to command the tool storage location with the next higher location number into the reference position. The tool management transfers the new position to the Axes Processor that controls the servo tool storage axis. The data word PxxS.MGCP contains the commanded location number that must be moved to the reference position if the tool storage is SPS controlled.

PxxC.MGNEG Move Tool Storage Negative

“0”= Do not move

“0⇒1”= Move tool storage 1 location in negative dir.

A signal change from “0” to “1” causes the tool management to command the tool storage location with the next lower location number into the reference position. The tool management transfers the new position to the Axes Processor that controls the servo tool storage axis. The data word PxxS.MGCP contains the commanded location number that must be moved to the reference position if the tool storage is SPS controlled.

PxxC.MGAP Actual Tool Storage Position

The contents of this integer data is necessary only for a SPS controlled tool storage. The SPS must write the number of the location that is currently in the reference position into this data word to keep the tool management informed.

NOTE: *The tool management does NOT check that the correct location is at the reference position if the SPS confirms the movement completion via FUNCTION MMV_Q. The tool management does not compare PxxS.MGCP and PxxC.MGAP! However, the location number specified in PxxC.MGAP is used in consecutive tool transfer FUNCTIONS!*

PxxC.MGNTC No Automatic Tool Check ^{(2),(3),(4),(5)}

“0”= Tool check is performed

“1”= Tool check is not performed

Tool management does not perform an automatic tool check upon a restart if this signal is set to ‘1’.

Tool management will not take existing setup list into account, only general management tasks will be performed.

7.8.2 Tool Management Status Signals (CNC⇒SPS)

The tool management provides information about the tool storage and status, this information is contained in the tool storage status Gateway signals. The status is independent of the type of tool storage drive (servo or SPS controlled).

PxxS.MGENA Tool Storage Enabled

“0”= Tool storage movements disabled

“1”= Tool storage movements enabled

The tool management uses this signal to enable or disable the tool storage movement if it is SPS controlled, and not NC servo controlled.

NOTE: *The signal PxxS.MGENA must be included in any logic within the SPS program that controls tool storage movements. A signal status of “0” must be used to prevent damage to man and machine!*

PxxS.MGREQ CNC Tool Storage Request

“0”= CNC does not use or request the tool storage

“1”= CNC uses or request the tool storage

This signal is set to “1” whenever a NC-command MTP, MMP, MFP, MOP, MHP, or MRF is executed. This signal is reset to “0” if the NC-commands MEN, BST, or RET is executed. A “Control Reset” or a process stop also reset the status of PxxS.MGREQ to “0”.

PxxS.MGMOV Tool Storage is Moving

“0”= No tool storage movement (standstill)

“1”= Tool storage is moving

The status of this signal is “1” whenever the tool storage moves. The signal is valid for SPS and servo controlled tool storages. The movement can be commanded by either the CNC or the SPS.

PxxS.MGMAN Manual Tool Storage Mode Status

“0”= Tool storage not in manual mode

“1”= Tool storage is in manual mode

The tool management sets this signal to “1” only if the SPS requests manual tool storage mode (PxxC.MGMAN=“1”) and the CNC does not request or use the tool storage. The NC-command MEN can be used to allow SPS manual mode access.

PxxS.MGBP Tool Storage in Base Position

“0”= Tool storage in base position

“1”= Tool storage is in base position

This signal is only necessary for SPS controlled tool storages. Its status is set to “1” if the tool storage is at its base position. This means that location number 1 is at the tool storage reference position.

PxxS.MGCP Commanded Tool Storage Position

The data contains the location number (+/- 0...999) that must be moved to the **tool storage reference** position. The contents of this signal is only required when using a SPS controlled tool storage device (Magazine/Turret).

PxxS.MGCP contains the tool location number that must be moved **to the reference position** in order to locate the programmed tool in the selected tool change position.

PxxS.MGTWO Tool Worn Out

“0”= The necessary tool can be used

“1”= Last available tool is worn out

The Status of PxxS.MGTWO is set to “1” when, one or more tool Edges of the last tool within the tool family is worn out and,

- activating another tool Edge (NC-command E).
- activating the already active tool Edge again (NC-command E).
- transferring the tool back into the tool storage location (magazine pocket).
- moving the tool of a turret out of machining position.
- when de-selecting the tool via T-number T0.

Tool management updates the data of the Active Tool List during the automatic tool check (performed whenever a different Setup List or Tool List is loaded into the MT-CNC or when activating a different NC-memory (A,B).

PxxS.MGWRN Tool Warning Limit

“0”= No tool has reached its Warning Limit

“1”= At least one tool has reached its Warning Limits

The status of PxxS.MGWRN is set to “1” to indicate that the last usable tool of a tool family exceeded its warning limit and,

- activating another tool Edge (NC-command E).
- activating the already active tool Edge again (NC-command E).
- transferring the tool back into the tool storage location (Magazine pocket).
- moving the tool of a turret out of the machining position.
- When de-selecting the tool via T-number T0.

Tool management updates the data of the Active Tool List during the automatic tool check (performed whenever a different Setup List or Tool List is loaded into the MT-CNC or when activating a different NC-memory (A,B).

PxxS.MGERR Tool Storage / Tool Error

“0”= All required tools are O.K. (no error)

“1”= Tool storage or tool error active

This signal is set to a status of “1” if

- the last usable tool or at least one tool family is worn out
- the Active Tool List does not fulfill the Setup List requirements
- a tool that was programmed via the T-word (T) can not be found
- the signal PxxS.MGTWO has a status of “1” at the end of NC-program execution (BST, RET)
- tool management recognizes potential tool collisions

Tool management updates the data of the Active Tool List during the automatic tool check (performed whenever a different Setup List or Tool List is loaded into the MT-CNC or when activating a different NC-memory (A,B).

8. STATUS, DIAGNOSTICS and SETUP

8.1 NC Status and Diagnostics

The MUI can display up to 600 messages for each process with further information. Some of these messages are MT-CNC activated messages. The user can activate up to 150 Messages in the SPS program per process or slave mechanism.

Each message is activated via a message number. The MUI and SOT displays one message at a time.

The text of the status message numbers are displayed as normal text. Text of error message numbers are displayed in reverse video. Error message numbers also activate the text "Fault in Mechanism" in the MUI and "Mech. Error" in the SOT.

Messages can be activated in the SPS program via the FUNCTION MSG_WR. The currently active message number can be read via the FUNCTION MSG_RD.

Table 8-1: MT-CNC Message Number Listing

MSG #	Activation type	MUI message text display	SOT message text display
600 : 551	User (SPS) error messages	FAULT IN MECHANISM and station name in REVERSE VIDEO	**MECHAN.ERROR** and message text in REVERSE VIDEO
550 : 251	Basic process messages	Status / Error	Status / Error
250 : 201	User (SPS) error messages	FAULT IN MECHANISM and station name in REVERSE VIDEO	**MECHAN.ERROR** and message text in REVERSE VIDEO
200 : 151	User (SPS) status messages	normal text	normal text
150 : 1	Basic process messages	Status / Error	Status / Error
0	Message: No message text available for error code "0"	normal text	normal text

NOTE: Each system diagnostic includes further information which includes a paragraph containing the cause, and a paragraph containing a recovery of the present fault. Further information for SPS activated faults are dependent on the programmer.

8.2 Fault List

The MT-CNC has a feature which will log all faults that occur in the MT-CNC. In order to make use of this function the LOGLIST.EXE file must be activated. To activate this file the MT-RUN.BAT file must be edited. A (/I or /L) must be appended to the line TSRPG25I in MT-RUN.BAT. Once the file is edited the word "LOG" will appear in the upper right hand corner in the MUI the next time MT-RUN.BAT file is executed.

To (display / print) logged errors the following hot keys must be used:

<Main Menu>, <F7>, <CTRL>+<F2>

/T To display Diagnostic message text
/I To display Basic Messages only (0...150, 251...550)
/U To display User (SPS) activated Messages only
(151...250, 551...999)

/N Display mechanism type number
/C Comma-separate values for Spreadsheet Import

Output to: | Print (output to printer)
Output to: | More (show one screen at a time)

8.3 Hardware and Software Requirements

8.3.1 Computer Requirements (PC)

The MUI/GUI software package is designed to operate on an IBM or compatible MS-DOS based computer. Due to the many computers on the market, the following minimum hardware is recommended:

- MS-DOS \geq 6.20 operating system:

486DX processor, running at 66 MHz or faster

8 Mbytes RAM or More (all allocated as extended Memory)

EGA, or VGA display. (LCD screens should be VGA compatible)

3.5" HD Floppy disk Drive. (1.44 MB)

80 MB Hard Disk Drive, **min. 35MByte free for new installations and 15mbyte fee for upgrades.**

- Windows NT 3.51 operating system (under a FAT disk):

486DX processor, running at \geq 100 MHz or Pentium at \geq 90 MHz

\geq 16 Mbytes RAM

EGA, or VGA display. (LCD screens should be VGA compatible)

3.5" HD Floppy disk Drive. (1.44 MB)

250 MB Hard Disk Drive, **min. 35MByte free for new installations and 15mbyte fee for upgrades.**

The MUI/GUI software may be run on Laptop and portable computers provided that the LCD, or Plasma screen drivers are VGA compatible.

8.3.2 MS-DOS Software Configuration Recommendations for compatible PCs

Different hardware installed in a computer may require non-standard device drivers and/or TSR programs (Terminate and Stay Resident) to be loaded into memory. Some of these programs may interfere with the operation of the MUI/GUI software.

The MUI/GUI software needs **589 Kbytes of RAM** to operate before it is loaded into memory.

A system with extended memory would allow TSRs and disk caches to be loaded into high memory, while taking up only a small amount of the lower 640K (DOS 6.0 and up). The MUI/GUI software can only be run in the lower 640K of memory, it will make use of extended memory !

The following pages list a typical configuration for compatible PCs and Indramat's BTV 1.xx. This is not a required system configuration, but simply an example of a working configuration. The example also shows the way that the programs are loaded into memory.

To change the configuration, refer to your PC's and the DOS user manual and read the following :

8.3.2.1 Example of the CONFIG.SYS file of compatible PCs

```
DEVICEHIGH=C:\DOS\SETVER.EXE
DEVICE=C:\DOS\HIMEM.SYS
DEVICEHIGH=C:\DOS\EMM386.EXE 2048 RAM
DEVICEHIGH=C:\DOS\ANSI.SYS /X
DOS=HIGH,UMB
STACKS=0,0
SHELL=COMMAND.COM /E:500 /P
FILES = 40
BUFFERS = 40
BREAK = OFF
```

8.3.2.2 The CONFIG.SYS file for Indramat's BTV 01

```
REM *****
REM ***** CONFIG.SYS                               01.09.1993 *****
REM ***** -----
REM ***** (C) INDRAMAT GmbH *****
REM *****
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM 2048
BUFFERS=40,0
FILES=40
DOS=UMB
LASTDRIVE=E
FCBS=4,0
DEVICEHIGH /L:1,12240 =C:\DOS\SETVER.EXE
DOS=HIGH
COUNTRY=049,850,C:\DOS\COUNTRY.SYS
BREAK = OFF
```

DOS commands are described briefly, refer to your DOS manual for detailed information.

The CONFIG.SYS file is used by the personal computer to load special device drivers, and to set up hardware options. The most common commands are the FILES and the BUFFERS commands. The FILES command should be set to 30 or greater. This command specifies the number of files that can be open at any given time, the MUI/GUI software uses multiple files in its operation.

The BUFFERS command should be set to 30 or greater. This is necessary because the MUI/GUI software is so disk intensive. This setting is dependent upon the size of hard disk that is installed in the computer and whether SMARTDRV.EXE is used in CONFIG.SYS.

In the case of 486 and Pentium computers with extended memory installed, the extended or expanded memory manager is usually installed via the DEVICE command, e.g.,

```
DEVICEHIGH=C:\DOS\EMM386.EXE RAM  
DEVICEHIGH=C:\DOS\ANSI.SYS /X
```

If WINDOWS is installed on your PC, you may also use the following statements:

```
DEVICE=C:\WINDOWS\HIMEM.SYS  
DEVICEHIGH=C:\WINDOWS\EMM386.EXE RAM
```

8.3.2.3 Example of the AUTOEXEC.BAT File for compatible PCs

```
@ECHO OFF  
PROMPT $P$G  
SET MOUSE=C:\LMOUSE\LMOUSE  
PATH=C:\DOS;\WINDOWS;C:  
SET TEMP=C:\WINDOWS\TEMP  
SMARTDRV 512 512  
lh Doskey  
MT-RUN  
C:\MT_TEXTE\KEYBIN
```

Include the statement KEYBIN if the Indramat keyboard BTM 02 is used. This will map a normal (101) keyboard so that eventually the driver must be loaded via a separate Batch file.

8.3.2.4 The AUTOEXEC.BAT File of Indramat's BTV 01

```
@ECHO OFF  
REM *****  
REM *****AUTOEXEC.BAT 01.09.1993 *****  
REM *****-----*****  
REM ***** (C) INDRAMAT GmbH *****  
REM *****  
PROMPT $p$g  
PATH C:\DOS  
SET TEMP=C:\DOS  
LH /L:1,16416 KEYB us, ,C:\DOS\Keyboard.SYS  
LH /L:2,6528 DOSKEY  
LH /L:0;1,42896 /S C:\DOS\SMARTDRV.EXE C
```

The PATH= statement is used in the AUTOEXEC.BAT to set up the default file search paths for various software packages. If the MT-CNC sub-directory is entered into the path statement, the MT-CNC software could be started from any DOS prompt. The MUI/GUI software may be started at the end of PC boot up by adding the line MT-RUN at the end of AUTOEXEC.BAT. DOS commands are described briefly, refer your DOS manual for detail information.

8.4 MT-RUN.BAT File

The MT-RUN.BAT file is generated during MUI/GUI installation and resides in DOS root directory of the drive containing the MUI/GUI software installation. Standard MT-RUN.BAT configuration for drive C and MUI/GUI directory MT-CNC:

```
ECHO OFF
REM INDRAMAT MT-CNC (c) Copyright 1994
REM -----
SET DPMIMEM=MAXMEM 4096
C:
  CD C:\MT-CNC
  TSRPG25I
  MT-CNC %1 %2
  CD\
REM -----
```

8.4.1 Software Options Which Can Appended to (TSRPG25I) Line of MT-RUN.BAT

```
/?          HELP SCREEN DISPLAY
/A          SWITCH ON DISPLAY FOR DETAILED EXE-FILE-INFORMATION
/B          SWITCH OFF CRTL-ALT-DEL
/C1 [/C]   TSR-PROGRAM WITH WIN-NT-DOS-BOX AND DPR_VDD.DLL
/C2        TSR-PROGRAM WITH WIN-NT-DOS-BOX AND INT_VDD.DLL
/D=$xxxx,$xxxx DPR-ADDRESS
/G          CREATE FILE "\GLOBAL.DAT"
/I=$xx,$yy PROGRAMMABLE INTERRUPTS : $xx = TSR-IDENTIFIER $yy = DPR-ADDRESS
/J          TSR-STOP BY TIMEOUT
/K          TSR-PROGRAM WILL BE RUN WITHOUT BOF-ENVIROMENT
/L          SWITCH ON LOG-FUNCTION
/O=xxxx    TIMER-DEVICE WILL BE PROGRAMMED
           1:55,5ms  2:27,7ms  4:14ms  8:7ms  16:3,5ms  32:1,7ms  64:868µs  128:434µs
/P          TSR-TEXT LOADED FROM DISK
/Q          COMMUNICATION-CONTROL BY DPR TO IPC1
/R          SWITCH ON IPC1-DPR-DRIVER WITH BYTE-SWAPPING
/R0        SWITCH ON IPC1-DPR-DRIVER WITHOUT BYTE-SWAPPING
/T          SWITCH OFF TIME-SYNCHRONIZATION
/U          TSR-PROGRAM IN DEBUGGER-MODE
/U0        MT-CNC-INTERFACE : TEST-DISPLAY ON
/W          NO SWAPPING FOR MAIN-MENUE
/X          CHANGE MODEM CONTROL SIGNALS DTR/RTS
/Y          SWITCH ON TELE-BYTE-FUNCTION AND LISTING IN "\TELEGRAM.TXT"
/Z          CHANGING DATA-PROTOCOL (CHECK-SUM FOR TRANSMIT-COUNTER)
```

8.4.2 Software Options Which Can be Appended to the MT-CNC Line of MT-RUN.BAT

```
/?          DOS-parameter switch Help display
/B=XX      MUI boots into the specified main menu item xx, e.g., /B=10 = GUI
/D=DIR     Allows to define a different work directory, e.g. ANLAGExx\*. * under
           another
           drive (g:\stal)
/E         Displays the developers name when viewing the MUI Program
           Administration
/F         Disables minimum Hard Disk memory warning function
/H         None MUI/GUI EXE-files may be started, using in GUI the EXE-calls for
           EXE-
           files defined in the file ..\MT_TEXTE\MAIN_EXE.DAT
/I         Use original Indramat diagnostic text and NOT the custom user
           diagnostic text
/M         Disables the minimum MUI/GUI RAM check at start
/N         New initial. of the MUI/GUI software, delete \MT_TEXTE\MT_START.DAT
/S         Start into Setup
/V         Inserts version of MT-CNC.EXE into EXE-HEADER
/Z         Displays the software license no.
/T         Disable password levels
```

8.5 Hot Keys

By pressing <ALT>+<H> in any menu in MUI, a list of hot keys for the active screen will appear.

8.6 Windows NT3.51 Software Configuration Recommendations for compatible PCs

GUI/MUI version 16 and up can run under the Windows NT 3.51 operating system. Communication via the Function Interface is required. The Function Interface allows 8 channels to communicate to the MT-CNC in Windows NT 3.51 only.

NOTE: The MT-CNC User Interface program was developed under MS DOS. The ability of the Function Interface's 8 channels of communications was not taken into account. **Thus, it is possible to open more than one MUI application at once.** *Indramat does not assume warranty for a malfunction resulting from the above mentioned condition.*

This description applies to the following:

Operating system version: Windows NT 3.51

MUI/GUI: 16VRS

Function Interface version: 02VRS

The following software products make up the Function Interface:

■ **SWA-PC*WNT-FLB-02VRS-EN-C1,44**

This software is required for linking an independent application software to the Function Interface.

■ **SWA-PC*WNT-DRV-02VRS-NN-C1,44**

This software is required for linking the MT-CNC MUI/GUI to the Function Interface.

(Interface Drivers)

■ **SWA-PC*WNT-DDE-01VRS-EN-C1,44**

This software is required for linking an application software to the DDE mechanism of the Function Interface.

The Function Interface software products are pre-installed if ordered with Indramat's BTV industrial computer. If not ordered with the BTV or if using a computer other than Indramat's BTV then Function Interface files can be purchased separately. If purchased separately, the files must be copied to the directory: **C:\MT-CNC\IND_DRV**

Except for the files for the Windows NT interface drivers, these files are handled separately as described in the next section.

8.6.1 Windows NT Interface Drivers

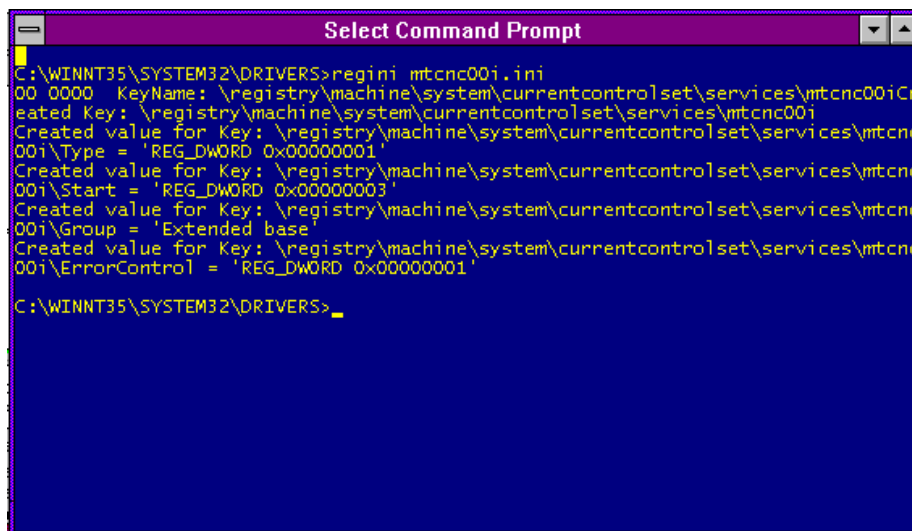
In order to ensure the best possible access and response speed for the Function Interface under Windows NT, it is absolutely necessary to reserve a serial interface. An Indramat driver for Windows NT is provided.

Installation procedure for the Interface Driver:

- **SWA-PC*WNT-DRV-02VRS-NN-C1,44** contains the files:

MTCNC00I.INI
MTCNC00I.SYS
REGINI.EXE

- These files must be copied to:
C:\WINNT35\SYSTEM32\DRIVERS
- Open MS-DOS prompt in the 'Main' group.
- Change to prompt to the directory **C:\WINNT35\SYSTEM32\DRIVERS**
- Copy the above 3 files to the directory **C:\WINNT35\SYSTEM32\DRIVERS**
- Call program REGINI with the file name MTCNC00I.INI
C:\WINNT35\SYSTEM32\DRIVERS> REGINI MTCNC00I.INI <enter>



```
C:\WINNT35\SYSTEM32\DRIVERS>regini mtcnc00i.ini
00 0000 KeyName: \registry\machine\system\currentcontrolset\services\mtcnc00i
Created Key: \registry\machine\system\currentcontrolset\services\mtcnc00i
Created value for Key: \registry\machine\system\currentcontrolset\services\mtcnc00i
Type = 'REG_DWORD 0x00000001'
Created value for Key: \registry\machine\system\currentcontrolset\services\mtcnc00i
Start = 'REG_DWORD 0x00000003'
Created value for Key: \registry\machine\system\currentcontrolset\services\mtcnc00i
Group = 'Extended base'
Created value for Key: \registry\machine\system\currentcontrolset\services\mtcnc00i
ErrorControl = 'REG_DWORD 0x00000001'

C:\WINNT35\SYSTEM32\DRIVERS>_
```

Figure 8-1: Registration Editor Call

- Close MS-DOS window by typing EXIT after execution and reboot Windows NT.
- After reboot of the operating system, select the icon 'Devices' from the 'Control Panel'. Device MTCNC00I must be displayed in the list of devices. The device start up type is automatic.

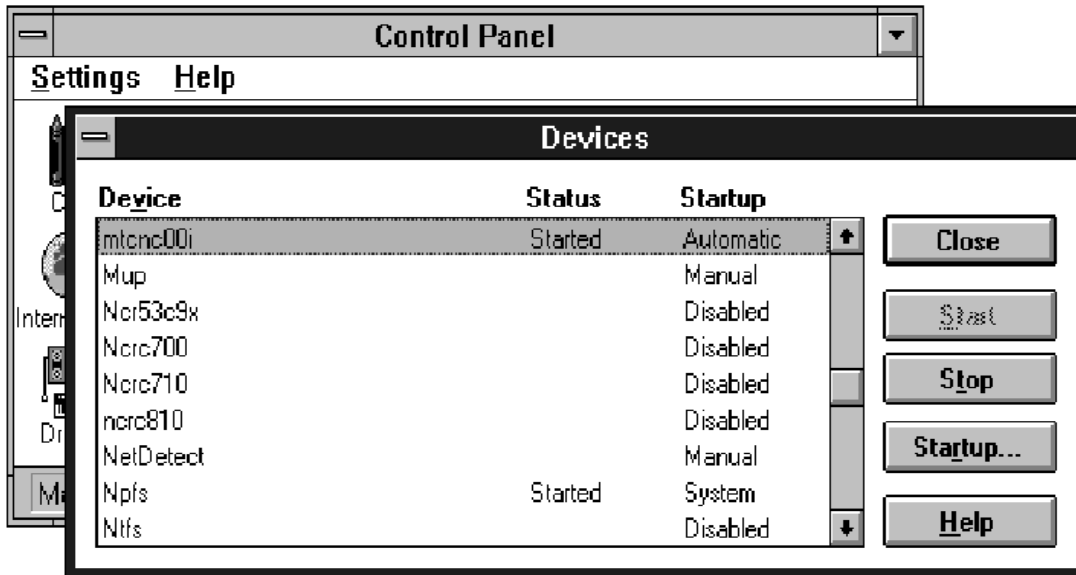


Fig. 8-2: Devices list in Windows NT showing Function Interface driver, mtcnc00i.sys.

8.6.2 Windows NT Task Response Time

To ensure the best response time for the Function Interface under all circumstances, the response time for foreground and background applications is set equally.



Figure 8-2: Windows NT System Setting 'Task Response Time'

8.6.3 PIF File Setting for the Window NT MS DOS Window

The Program Information File (PIF) is used to optimize application performance of non-Windows NT applications.

PIF Editor:

The PIF for the global MS DOS, _DEFAULT.PIF, window can be used to open the MS DOS based MT-CNC User Interface program.

- Open the PIF editor located in the Program Manager, Main Window.



Figure 8-3: PIF editor location in Windows NT Main Program Group

- From the PIF editor menu bar, select File and Open the _DEFAULT.PIF file located in the Windows NT directory.

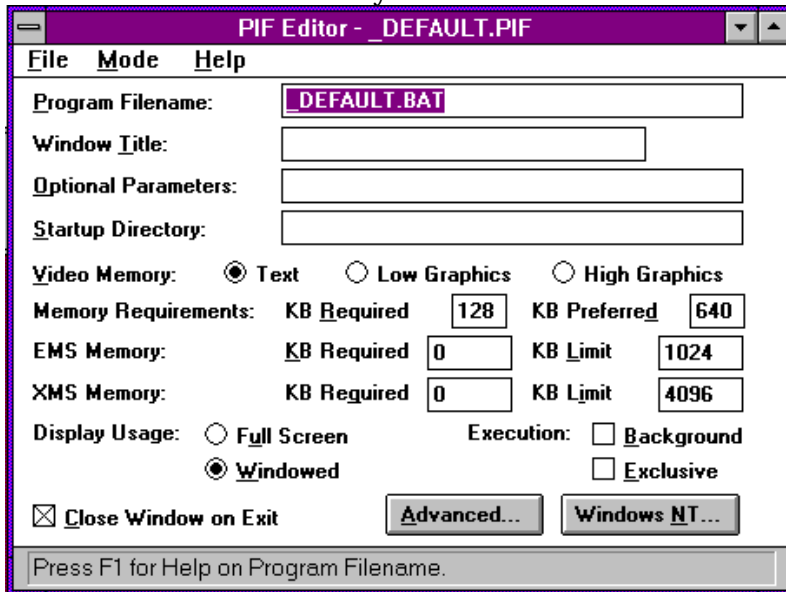


Figure 8-4: PIF Editor Main Menu

- Data input to the input fields determine how to call up the program.

- Video Memory: Windows NT can reserve a certain amount of memory for a screen built up, depending on the mode (text or graphics) of the running application. Concerning the MT-CNC User Interface, activate the option **TEXT**.
- Memory Requirements: 128 required, 640 KB limit
- EMS Memory: 0 required, 1024 KB limit
- XMS Memory: 0 required, 4096 KB limit

- Close Window on Exit: This option must be activated otherwise the MS DOS Window remains open when the MT-CNC User Interface is closed.

- Advance Options: Clicking the 'Advanced Options' button opens the 'Advanced Options' dialog window.

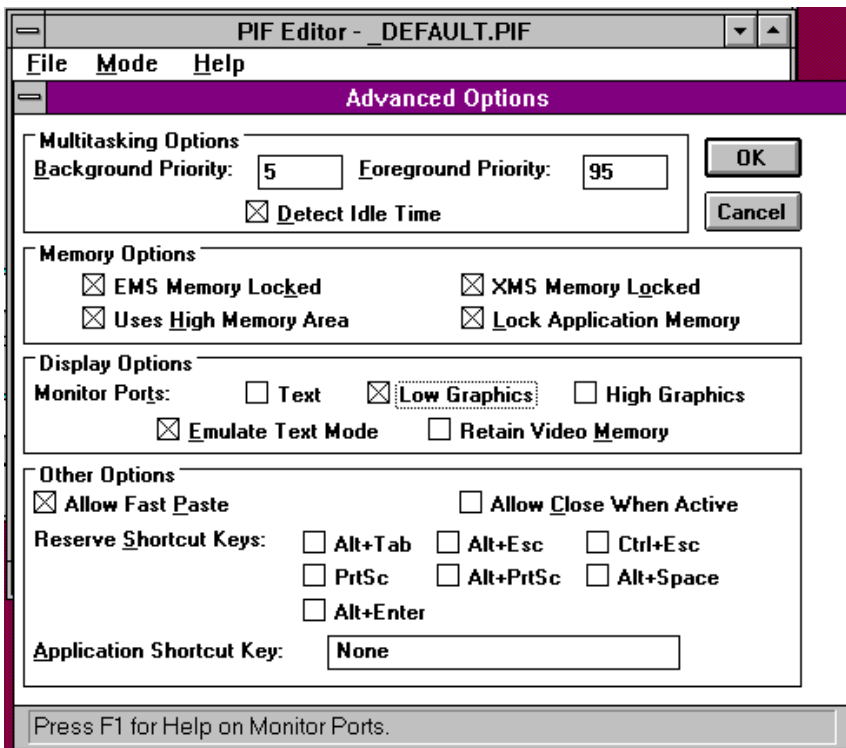


Figure 8-5: PIF Editor Dialog Window 'Advanced Options'

- Multitasking Options:
To determine the task Background Priority. If the 'Detect Idle Time' is selected, Windows NT does not allocate computing time to the application as long as it does not request any computing time. Select this option for MT-CNC User Interface.
- Additional Options:
Memory Options: Further data can be set in order to determine how memory is allocated for the application.
Display Options: Settings for the display
Other Options: Shortcut keys can be reserved for the application which normally initiates a Windows function. Fast Paste refers to inserting from temporary storage.
The above settings are required for the MT-CNC User Interface.

- Windows NT button: Clicking the 'Windows NT' button opens the 'Windows NT Option' dialog window.

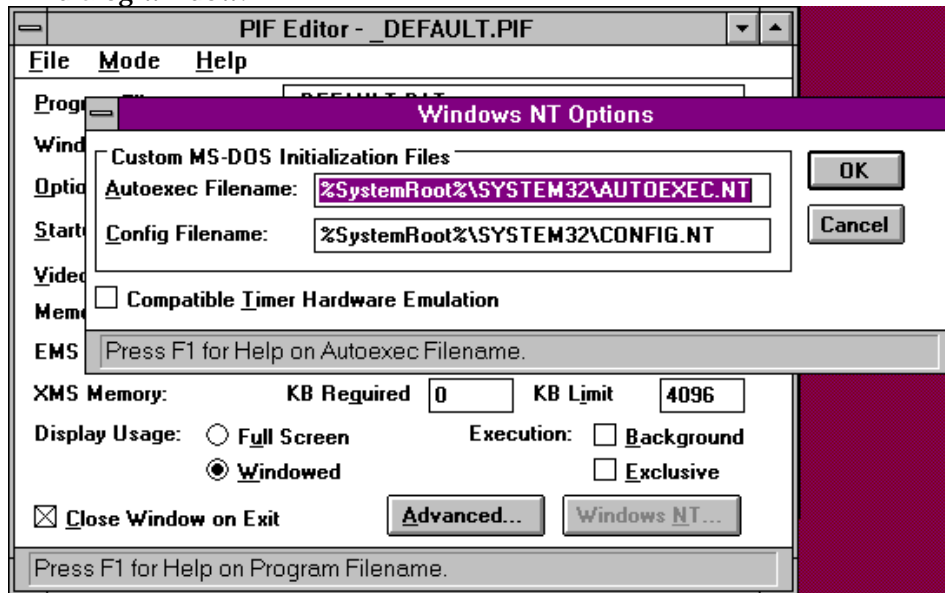


Figure 8-6: PIF Editor Dialog Window 'Windows NT Options'

Set as shown.

Individual start files can be determined in this dialog window. The option 'Compatible Timer Hardware Emulation' can be selected for applications which access the hardware time directly.

8.6.4 Batch File to Call Up the MT-CNC User Interface Software

MT-CNC User Interface version 16 and up have a second batch file called:

MT-RUNNT.BAT

Use this batch file when opening the MT-CNC User Interface software in Windows NT. The difference between MT-RUNNT.BAT and MT-RUN.BAT are two additional switches applied to the TSR program, /C2 and /O=16.

```

ECHO OFF
REM INDRAMAT      (c) Copyright 1996
REM INDRAMAT WINNT 3.5x  INSTALLATION
REM -----
SET DPMIMEM=MAXMEM 4096
C:
CD C:\MT-CNC
TSRPG25I /C2 /O=16
MT-CNC /H
CD \
REM -----

```


Figure 8-7: Batch File to call up the MT-CNC User Interface in Windows NT

NOTE: The batch file MT-RUN.BAT cannot be executed in the Windows NT operating system.

The additional switch /U0 on the TSR program line (TSRPG25I.EXE) causes the output of a diagnosis window for the Function Interface. The Function Interface's version identification and the capacity of the communications task is displayed in the diagnosis window.

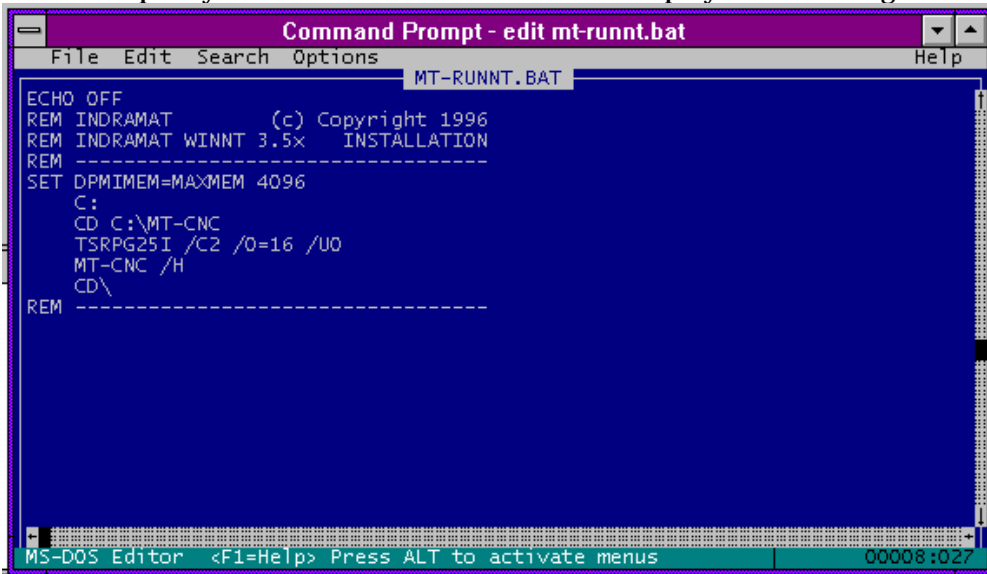


Figure 8-8: Batch File to call the MT-CNC User Interface under Windows NT with Function Interface Diagnosis Windows.

CNC-INTERFACE# 07-16T25 10-04-96	LOG-INTERFACE# 07-16T25 10-04-96	USR-INTERFACE# 0
Kanäle: 1 ----- (1:150)	08 00 53 08 00 00 00 87 1E 00 03 01 EA 00 2D 43 4E 43 20	
KOM: 1 ----- (50% 2373:1491)	LOGTASK1	
CNCTASK busy		
TIMER idle		

Figure 8-9: Function Interface Diagnosis Windows

8.6.5 Create MT-CNC User Interface Program Icon

A new program icon in a pre selected program group can be created by clicking on 'File', 'New' in the Program Manager. The new icon will be generated by clicking on 'Program' and then the 'OK' button. The 'Program Item Properties' dialog is displayed.

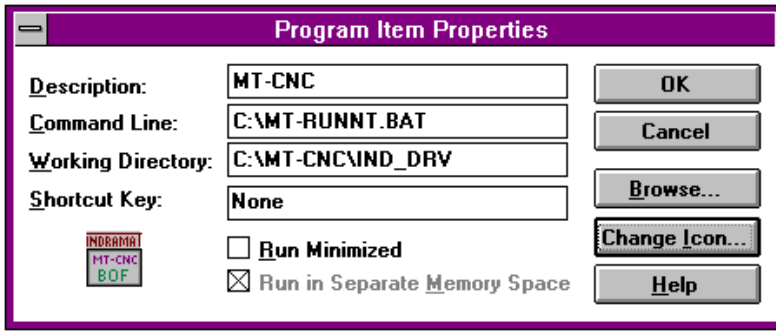


Figure 8-10: 'Program Item Properties' Dialog Window

The program item properties are accepted by confirming the text displayed in the figure above. The icon is stored as: **MT-CNC.ICO**
Located in the directory: **C:\MT-CNC\IND_DRV**

The icon can be changed by clicking on the 'Change Icon...' button.
The program icon will be completed by clicking the 'OK' button.

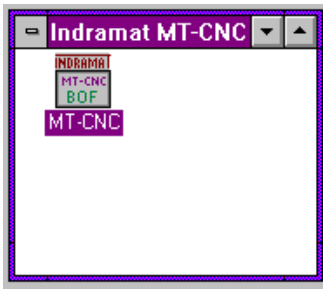


Figure 8-11: 'Indramat MT-CNC' Program Group with 'MT-CNC' Program Icon

8.7 DDE Server Deployment Conditions

8.7.1 Required Software for the DDE Server

The INDRAMAT DDE server is a software product which must be ordered separately by the following type code:

SWA-PC*WNT-DDE-01VRS-EN-C1,44.

The floppy disk contains software required for the link of a DDE client.

8.7.2 Cooperation with other Software Components

Correct performance of the DDE server is only guaranteed, if the following software components are installed correctly and are located in the directories listed below.

File	Directory	Purpose
COMINTFC.EXE	Located in the directory (driver directory) described in the file C:\INDRAMAT.DIR	Communication Interface to MT-CNC control
LOGINTFC.EXE	Located in the directory (driver directory) described in the file C:\INDRAMAT.DIR	Logic Interface between User Tasks and Communication Interface.
USRBEF.DAT	Located in the directory (driver directory) described in the file C:\INDRAMAT.DIR	Function Interface definition file
DDEXLI.EXE	Located in the directory (driver directory) described in the file C:\INDRAMAT.DIR	INDRAMAT DDE Server

File	Directory	Purpose
PASSWORD.DDE	Located in the directory (driver directory) described in the file C:\INDRAMAT.DIR	DDE Server password file
CLIENT.EXE	C:\MT-CNC\IND_DRV	Microsoft DDE client.
MTCNC00I.INI	Located in the operating system directory DRV:\%SystemRoot%\SYSTEM32\DRIVERS\	Windows NT interface driver registration data
MTCNC00I.SYS	Located in the operating system directory DRV:\%SystemRoot%\SYSTEM32\DRIVERS\	Windows NT interface driver
REGINI.EXE	Located in the operating system directory DRV:\%SystemRoot%\SYSTEM32\DRIVERS\	NT interface driver registration auxiliary program
INDRTASK.EXE	Located in the directory (driver directory) described in the file C:\INDRAMAT.DIR	INDRAMAT Program Manager to start multiple applications in a predetermined sequence
TASKDATA.INI	Located in the directory (driver directory) described in the file C:\MT-CNC\IND_DRV	INDRAMAT Programm Manager script file.
MFC.LZH	Decompress this file to the directory C:\MT-CNC\IND_DRV	INDRAMAT DDE-Server's Visual C++ library
LHA.EXE		Program for decompressing MFC.LZH.

The files listed in the table above as well as the following description relate to DDE server version 01VRS.

8.8 General

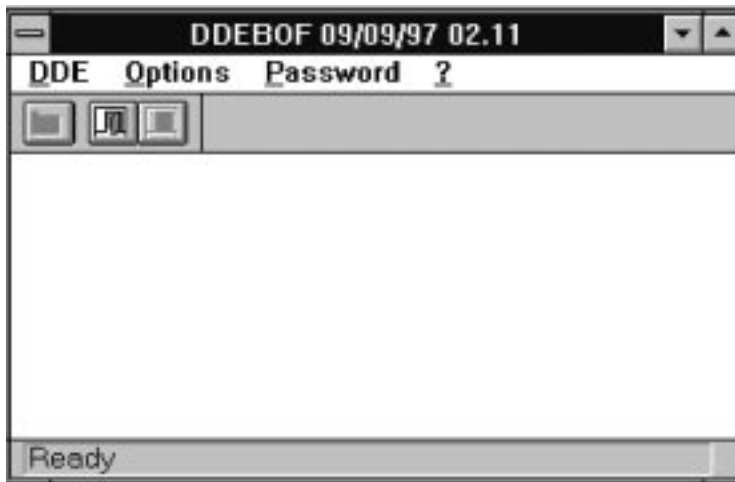


Figure 8-12: DDE Server

The DDE server user interface is shown in the figure above. Operation is performed via tool bar or pull down menus.

Care must be taken that only **one** client can communicate with the server. The link between the server and the user interface can only open one communication channel. This problem can be avoided if one instance of the server is started for each client served.

8.9 Program Call

The server is started by calling up the program DDEXLI.EXE. This can be done via file manager in Windows NT. An icon can be created in a program group. The DDE server is

started automatically when opened. Access to read and write data or to stop or close the DDE server is determined by password administration in the DDE software itself. Additional parameters switches can be added to the command line.

Parameter	Meaning
/d=f	The DDE server is started with trace recording
/c=t	Function Interface diagnostic windows

Fig. 8-1: Call Parameter

Example:

DDEXLI.EXE /c=t /d=f DDE server automatic start with Function Interface diagnostics window output and trace recording in DDE window.

8.10 Password Administration

As of DDE server 'DDEBOF' version 01.10, the DDE server contains a password administration. The password administration requires the file 'password.dde' located in directory
C:\MT-CNC\IND_DRV.

8.10.1 First Execution of Password Administration

In order to call the password administration, the user needs to log in first. This is done via menu 'Password -> Login'

For a first time Log In use the following User name and Password :

- Username: **INDEECNC**
- Password: **CNC**

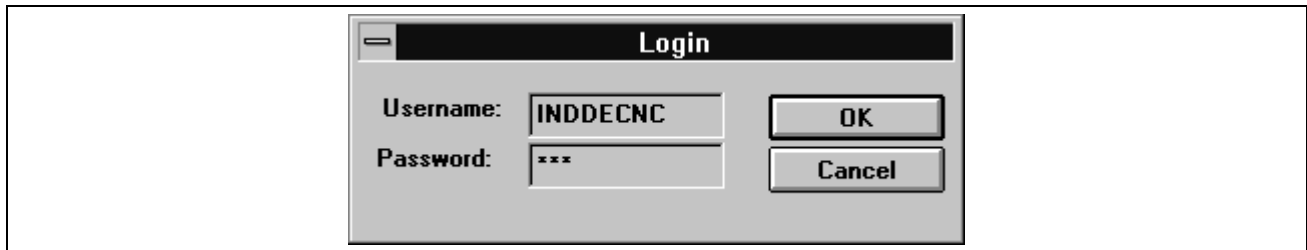


Figure 8-13: DDE Server Password Administration 'Login Menu'

Password administration is called by the menu item 'Password → New Password'. The following dialog appears:

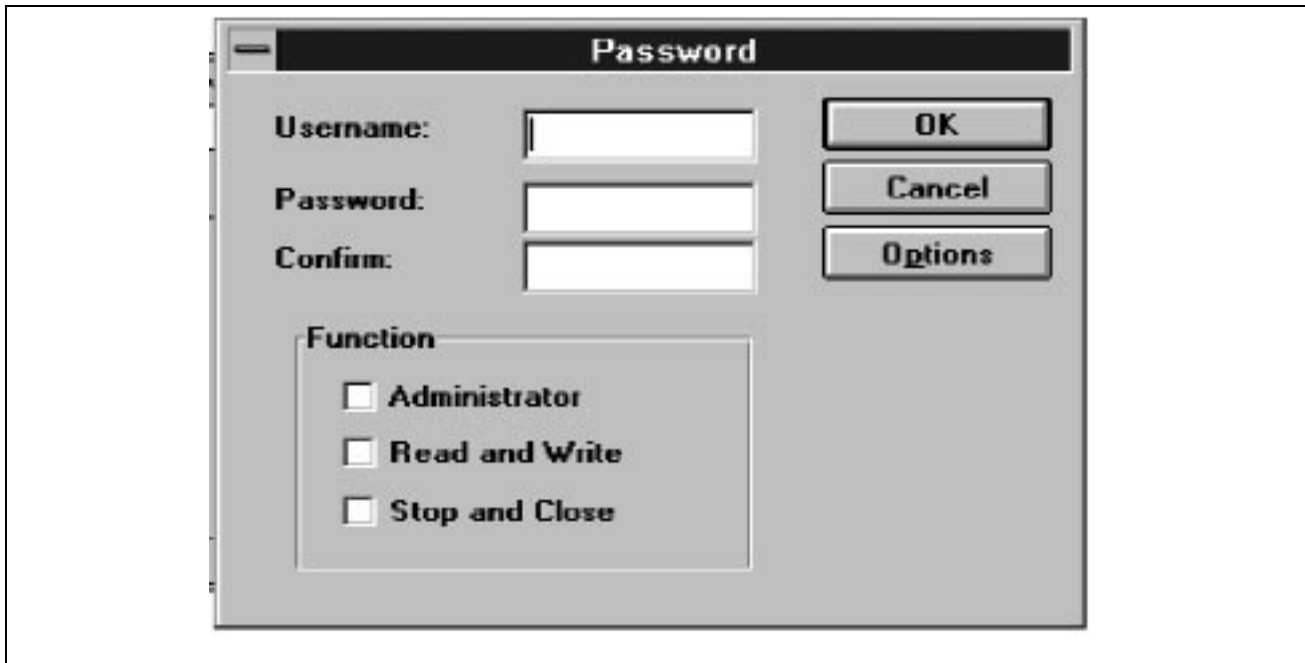


Figure 8-14: DDE Server Password Administration ‘Enter Password’

The Password Administration consists of four statements:

Designation	Fieldname	Example
User name	Username	INDRAMAT (20 characters max.)
Password	Password	MTC200 [*****] (20 characters max.)
Password confirmation	Confirm	MTC200 [*****] (20 characters max.)

The first user must be assigned as administrator.

Only the administrator can assign other users to the DDE server.

Multiple functions can be selected from the list of rights (for example Function *Read*, *Write*, *Stop* and *Close*).

Click the ‘OK’ button to accept the selection. After terminating the server or logging out by the menu item ‘LogOut’, it will not be possible to log in with user name ‘INDDECNC’ and password ‘CNC’.

An administrator can view current users of the DDE by clicking on the ‘Options ‘ button from the ‘Password’ window. The administrator can delete users from this window.

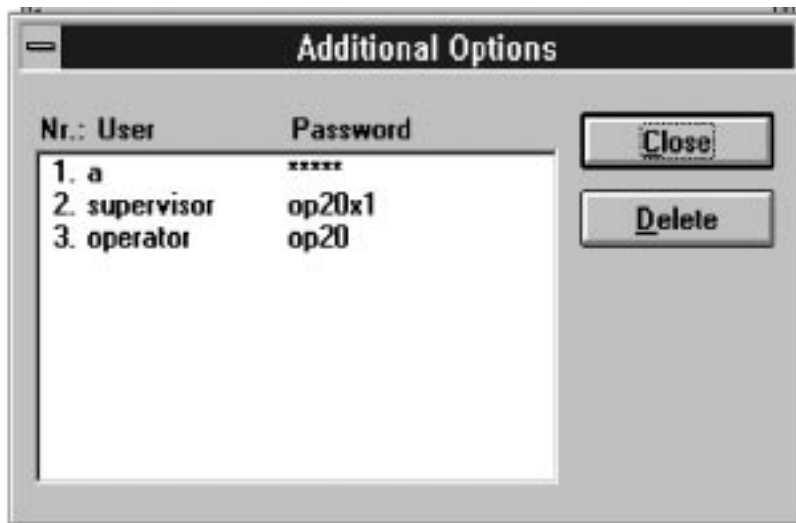


Figure 8-15: DDE Server Password Administration 'User list window'.

8.10.2 Register New User

To assign a new user name and passwords, the administrator user must Log In. The above dialog is displayed and a new user can be entered in.

8.11 DDE Client Connect to Indramat's DDE Server

For a DDE client to connect to the Indramat DDE server the following Application name and Topic name must be used:

APPLICATION: **INDDECNC**

TOPIC: **CNC**

8.11.1 Automatic Start

The DDE server is started automatically when it is launched or opened by executing the file DDEXLI.EXE via icon or from the file manager of Windows NT. If the ability to write data to the MT-CNC from a DDE client by the DDE server or to stop or close the DDE server is necessary, then the user with the rights must Log In.

Only the function 'Read' is assigned to the user without a Log In.

8.11.2 Data Access Function Commands

CNC data, CNC variables, axis position, active NC program and more can be directly accessed from the MT-CNC.

PLC variables, arrays and structures are accessed directly by an identifier due to the given structures of IEC-1131. Variables must be determined by the complete identifier. Entire arrays can be accessed by their complete identifier or single array elements by the concerning element identifier. These conditions apply also to reading and writing of structures.

To read or write to a PLC Variable the Function Command **PVS_{variable name}** is used.

For a complete list of Function commands, refer to the MT-CNC Function Interface V02 manual.

When accessing entire arrays or entire data structures, a special file, located in the directory

\\MT-CNC\IND_DRV called PVCTYPEN.INI can be modified to define the data type of the array. Then the Function Command used in conjunction with the PVCTYPEN.INI file is

PVC_{variable name}.

The file PVCTYPEN.INI will contain the name of the array or data structure used in a PLC program. In the square brackets is the name of the array or data structure to be accessed from the SPS (PLC) in the MT-CNC. The next line (TYPEN=) is how the data will appear in the host software (MSClient, Wonderware or other) when accessed via DDE. The different data types are listed below in the example.

The information can be separated by a given character or line feed defined in the next line, SEPARATOR=xx. Equate this to an ascii character's decimal value and this character will be used as the separator between each value of the array information.

A separate line must be created for each array or data structure needing a data conversion.

Example:

```
[PLCARRAY]
TYPEN=BOOL,BOOL,BOOL,BOOL,BYTE,WORD,DWORD,SINT,INT,DINT,USINT,UINT,UDINT,CHAR,STRING[10],STRING[5]
SEPARATOR=47
```

```
[DATASTR]
TYPEN=BOOL,BOOL,BOOL,BOOL,BYTE,WORD,DWORD,SINT,INT,DINT,USINT,UINT,UDINT,CHAR,STRING[10],STRING[5]
SEPARATOR=47
```

If you write in PVCTYPEN.INI SEPARATOR=10 (LF) or SEPARATOR=13 (CR), then it makes a pair CRLF (control, line feed).

INDEX

1

10^ 4-21

2

2^ 4-21

6

65000	6-39
65001	6-39
65002	6-39
65003	6-40
65004	6-40
65005	6-47
65006	6-40
65007	6-40
65008	6-41
65009	6-41
65010	6-43
65011	6-48
65012	6-42
65013	6-42
65014	6-42
65015	6-42
65016	6-43
65017	6-43
65018	6-43
65019	6-40
65020	6-40
65021	6-40
65022	6-43
65023	6-44
65024	6-44
65025	6-48
65026	6-44
65027	6-44
65028	6-45
65030	6-45
65031	6-48
65032	6-47
65033	6-48
65034	6-49
65035	6-46
65036	6-46
65038	6-46
65039	6-41
65040	6-47
65041	6-47
65042	6-45
65043	6-47
65044	6-47
65045	6-49

65046	6-49
65047	6-49
65048	6-49
65049	6-49
65050	6-44, 6-49
65051	6-50
65052	6-50
65053	6-50
65054	6-50
65055	6-50
65056	6-50
65057	6-50
65058	6-50
65061	6-51
65063	6-51
65065	6-51
65066	6-51

A

A00.000	3-1
A00.001-A00.006	3-1
A00.007-A00.031	3-1
A00.032-A00.51	3-1
A00.052	3-1, 7-14
A00.053	7-14
A00.053 - A00.083	3-1
A00.054	7-14
A00.055	7-14, 7-15
A00.056	7-14
A00.057	7-14
A00.058	7-14, 7-15
A00.059	7-15
A00.060	7-15
A00.061	7-15
A00.069	7-15
A00.070	7-16
A00.075	7-16
A00.082	7-16
A00.083	7-16
A00.086	7-16
A00.087	3-1, 3-3
A00.088	3-1, 3-4
A00.089	3-1
A00.090	3-1
A00.091	7-16
A00.096	7-16
ABS	4-20
ABS_INT	5-27
Absolute value	4-20
ACC	4-25
ACC_EFF	4-28
Accel	3-3, 3-6
Accel. Rate, Max	3-6
Accel. Rate, Max. (rotary)	3-12
Accel. Spdl.	3-10, 3-13
Acceleration Override	6-44

Acceleration, path	3-2	Axis Signals.....	6-27
Acceleration, Path Limit	4-25	Axis Status Signals (CNC⇒SPS).....	6-36
ACD_COMP.....	4-29	Axis Switching	4-24
Acknowledge Mechanism Execution	6-26	AxxC.M5.....	6-33
ACOS.....	4-21	AxxC.ENABL	6-28, 6-30
Active NC-program, GUI (Indramat default).....	2-18	AxxC.GEAR1	6-32
Active User Data Archive, GUI (Indramat default)	2-8	AxxC.GEAR2	6-32
<u>A</u> ctual <u>D</u> ata for NC- <u>C</u> OMPiler	4-29	AxxC.HOME.....	6-30
ADD		AxxC.HOMLS.....	6-29
Instruction.....	5-5	AxxC.JGNEG	6-30, 6-31
Operation.....	5-12	AxxC.JGPOS	6-30
ADTRC.....	4-28	AxxC.M19.....	6-33
Advance NC Program Req.....	6-11	AxxC.M3.....	6-32
Advance Process.....	6-21	AxxC.M4.....	6-33
Advance Program.....	4-1, 6-6, 6-7	AxxC.MHOLD	6-34
Advance Program Start	6-26	AxxC.MTAS	6-4, 6-28
Advance Program.....	6-11	AxxC.N_CMD	6-24, 6-34
AND		AxxC.OTRVL	6-4, 6-27
Instruction.....	5-3	AxxC.OVRD.....	6-18, 6-31
Operation.....	5-11	AxxC.QDDS.....	6-31
AND(AxxC.RAPID	6-34
Instruction.....	5-3	AxxC.READY	6-28, 6-30
ANDN		AxxC.SPHLT	6-24, 6-35
Instruction.....	5-3	AxxC.SPRST.....	6-35
ANDN(AxxC.SPSTE.....	6-35
Instruction.....	5-3	AxxC.SPSTP	6-23, 6-35
AP	4-14	AxxC.STRBP	6-29
<u>A</u> pproach <u>D</u> istance for <u>T</u> ool <u>R</u> adius <u>C</u> omp.....	4-28	AxxS.BBDIG.....	6-36
APR parameters	6-39	AxxS.HOMED	6-36
Arc Cosine	4-21	AxxS.IDDS.....	6-37
Arc Sine	4-21	AxxS.INPOS	6-38
Arc Tangent	4-21	AxxS.LD90.....	6-37
Arithmetic instructions.....	5-5	AxxS.MCNEG.....	6-39
Arithmetic operations.....	5-12	AxxS.MCPOS	6-39
Arrays.....	5-1	AxxS.MD_DX.....	6-38
ASIN.....	4-21	AxxS.MVNEG	6-36
Associated Processes	3-4, 3-9, 3-11	AxxS.MVPOS	6-36
Asynchronous Turret Movement	7-39	AxxS.N_CMD	6-37
ATAN	4-21	AxxS.N_MAX.....	6-38
Auto Mode, PxxC.ADV.....	6-11	AxxS.N_MIN	6-38
AUTOEXEC.Bat	8-4	AxxS.P_PX	6-38
AUTOMATIC MODE.....	6-6	AxxS.POWN.....	6-38
Automatic Spindle Gear Range Selection.....	3-11, 3-15	AxxS.RF.....	6-30, 6-36
Aux. Functions		AxxS.SYNC	6-37
M-Function.....	4-12	AxxS.WPn.....	6-37
S-Function	4-12		
AXD.....	4-22		
Axis Control Signals (SPS⇒CNC).....	6-28		
Axis Designation 1.....	3-5, 3-12		
Axis Designation 2.....	3-8, 3-15		
Axis Designation, Spdl	3-9, 3-12		
Axis Droop Compensation.....	3-8, 6-49		
Axis Enable.....	6-28		
Axis Movement in Neg. Dir.....	6-36		
Axis Movement in Pos. Dir.....	6-36		
Axis Number of Assigned Rotary Axis.....	3-11		
Axis Parameters	3-4		
Axis Position.....	6-49		
axis processors (APR).....	3-1		
Axis Ready to Operate	6-28		
Axis Regulator Release	6-36		

B

Backlash, Compensation	3-7
Backup all User Data, GUI (Indramat default)	2-6
BACKUP, GUI (Indramat default).....	2-6
Ballscrew lead.....	3-5
Basic elements of LAD.....	5-10
Bb1 signal.....	6-3
BBTPC	4-28
BCD_INT	5-17
BEQ.....	4-17
BER.....	4-17
BES	4-17
BEV.....	4-19

Bit string functions.....	5-29
Block Active	6-15
Block Buffer for Tool Path Compensation.....	4-28
Block Comment, NC.....	4-3
Block Numbers, NC.....	4-2
BMI.....	4-17
BNE	4-17
BPL.....	4-17
BRA	4-15
Branch Absolute.....	4-15
Branch Commands.....	4-17
Branch if <= Zero.....	4-17
Branch if = Zero.....	4-17
Branch if >= Zero.....	4-17
Branch if Event Reset.....	4-17
Branch if Event Set	4-17
Branch if Not = to Zero.....	4-17
Branch if Ref.....	4-17
Branch with Stop.....	4-15
BRF.....	4-17
BSE.....	7-24
BSR.....	4-16
BST	4-15
BTE.....	7-24
BUFFERS	8-4
Bus Voltage.....	6-3
Bxx.001	3-2
Bxx.002.....	3-2, 6-8
Bxx.003.....	3-2
Bxx.004.....	3-2
Bxx.005.....	3-2
Bxx.006.....	3-2
Bxx.007.....	3-2
Bxx.012.....	3-2
Bxx.013.....	3-2
Bxx.014.....	7-16
Bxx.015.....	7-16
Bxx.016.....	7-16
Bxx.017.....	7-17
Bxx.018.....	7-17
Bxx.019.....	7-17
Bxx.020.....	7-17
Bxx.021.....	7-17
Bxx.024.....	7-17
Bxx.025.....	7-17
Bxx.029.....	3-2
Bxx.030.....	3-2
Bxx.031.....	3-2
Bxx.032.....	3-2, 6-15
Bxx.033.....	3-2, 6-15
Bxx.034.....	3-3
Bxx.035.....	3-3
Bxx.036.....	3-3
Bxx.037.....	3-3
Bxx.038.....	3-3
Bxx.039.....	3-3
Bxx.040.....	3-3
Bxx.041.....	3-3
Bxx.042.....	3-3
Bxx.043.....	7-18
Bxx.044.....	3-3, 7-18

Bxx.046	3-3
Bxx.047	3-4
Bxx.048	3-4
Bxx.049	3-4
Bxx.050	3-4
Bxx.051	3-4
Bxx.052	3-4
Bxx.053	3-4
Bxx.054	3-4
Bxx.055	3-4
Bxx.056	3-4
BYT_CHAR.....	5-18
BYT_GRAY.....	5-18
BYTE_INT.....	5-18
BYTE_USI.....	5-19

C

CAL	
Instruction	5-4
CALC	
Instruction	5-4
CALCN	
Instruction	5-4
Cam Shift, combined spindle.....	6-47
Cartesian-Polar Coordinate.....	3-3
C-axis mode.....	3-3
C-Axis Parameters	3-11
CEV.....	4-19
CHAR_BYT.....	5-19
Character string functions.....	5-33
Clear Error.....	6-14
CLOS_COM.....	5-47
CLOS_SOT	5-51
CLR_COM	5-50
CNC⇌SPS.....	6-1, 6-14
COM - Standard data type.....	5-45
Command, Negative	6-39
Command, Positive.....	6-39
Commanded Axis Position	6-50
Commanded Speed, Spindle.....	6-34
Comparators	
Operation.....	5-15
Comparators	5-15
Instruction	5-7
Computer Requirements	8-2
CONCAT_B.....	5-32
CONCAT_S.....	5-35
CONCAT_W.....	5-32
Conditional Stop.....	6-13
CONFIG.SYS.....	8-3
Constant Surface Speed.....	6-17
Constants, NC.....	4-19
Continuous Jog Mode.....	6-8
Control Reset.....	3-3, 6-14
Control Voltage	6-2
Coordinate System.....	3-7, 3-14
COS	4-20
Cosine.....	4-20
Counters	

Up and down	5-41
CTUD_INT	5-42
CTUD_UIN	5-42
CTUD_USI	5-41
Current	6-47
Current, Peak	6-46
Cxx.000	3-4, 3-9, 3-11
Cxx.001	3-5, 3-9, 3-12
Cxx.002	3-9
Cxx.003	3-5
Cxx.004	3-5, 3-9
Cxx.005	3-5
Cxx.006	3-5, 3-12
Cxx.007	3-5, 3-12
Cxx.008	3-5, 3-12
Cxx.009	3-5, 3-9
Cxx.010	3-5, 3-9
Cxx.011	3-5
Cxx.012	3-5
Cxx.013	3-6, 3-12
Cxx.014	3-6, 3-12
Cxx.015	3-6
Cxx.016	3-6, 3-12
Cxx.017	3-6, 3-12
Cxx.018	3-6, 3-12
Cxx.019	3-6, 3-12
Cxx.020	3-6, 3-9, 3-12, 6-32, 6-33
Cxx.021	3-6, 3-9, 3-13
Cxx.022	3-6, 3-9, 3-13, 6-8
Cxx.023	3-6, 3-13
Cxx.024	3-7, 3-9
Cxx.025	3-7, 3-13
Cxx.032	3-7, 3-13
Cxx.033	3-7
Cxx.034	3-7
Cxx.035	3-10
Cxx.036	3-10
Cxx.037	3-10, 3-13
Cxx.038	3-10, 3-13
Cxx.039	3-10
Cxx.040	3-7, 3-10
Cxx.041	3-7, 3-10
Cxx.042	3-7
Cxx.043	3-7
Cxx.044	3-7
Cxx.045	3-7
Cxx.047	3-10, 3-13
Cxx.048	3-10, 3-13
Cxx.049	3-10, 3-13
Cxx.050	3-10, 3-13
Cxx.051	3-10, 3-14
Cxx.052	3-10, 3-14
Cxx.053	3-7, 3-14
Cxx.054	3-12
Cxx.055	3-14
Cxx.056	3-11
Cxx.057	3-9, 3-14
Cxx.058	3-9, 3-14
Cxx.059	3-9, 3-14
Cxx.060	3-8, 3-11, 3-14
Cxx.061	3-11, 3-14

Cxx.062	3-11, 3-14
Cxx.063	3-11, 3-14
Cxx.064	3-11, 3-15
Cxx.065	3-11, 3-14
Cxx.066	3-11, 3-14
Cxx.067	3-11, 3-15
Cxx.068	3-11, 3-14
Cxx.069	3-11, 3-14
Cxx.070	3-11, 3-15
Cxx.071	3-11, 3-14
Cxx.072	3-11, 3-14
Cxx.073	3-11, 3-15
Cxx.074	3-11, 3-15
Cxx.075	3-8, 3-15
Cxx.076	3-8, 3-15
Cxx.077	3-8
Cxx.078	3-8
Cxx.079	3-8
Cxx.080	3-8
Cxx.081	3-8
Cxx.082	3-8
Cxx.083	3-8
Cxx.084	3-11, 3-15
Cxx.085	3-8
Cxx.086	3-9, 3-15

D

D Correction, Radius	3-2
Data Exchange	4-22
Data types	
Arrays	5-1
Data types	5-1
Derived	5-1
Structures (STRUCT)	5-1
Date and time	5-44
DATE_RD	5-44
DCD_RD	7-44
DCD_WR	7-44
D-Correction	3-2
DDS Spindle Mode	6-41
decimal places displayed	3-2
Decimal Places, disp	3-2
Decimal Places, prog	3-2
Declaring Gateways	6-1
default	3-1, 3-2, 3-3, 3-9, 3-11
Default, linear interpolation	3-3
default, mm/inch (671/670)	3-2
Define Mechanism	6-26
Define Process	6-19
DELETE	5-37
Derived data types	5-1
DEV	4-19
Deviating M-functions	3-3
Deviating Q-functions	3-4
DEVICE command	8-4
DEVICEHIGH	8-4
Diagnostic Messages Editing, GUI (Indramat default)	2-27
Diagnostics	8-1
Diameter Prog., para	3-2

Digital Drive Parameter (Servo, Spindle) - Archive.....	1-5
Digital Drive Parameter Archive, GUI (Indramat default) ..	2-7
Digital Drive Parameter Editing, GUI (Indramat default) ..	2-30
Digital Drive Parameter Print Out, GUI (Indramat default)	2-34
Digital Drive Ready	6-36
Digital Drives.....	3-8
DINT_DW	5-19
DINT_INT	5-20
DINT_TIM.....	5-21
DINT_UDI.....	5-20
Direction for Polar Coord. Transformation.....	3-8, 3-15
Disable Mechanism.....	6-26
Distance Between Droop Compensation Points.....	3-8
DIV	
Instruction.....	5-6
Operation.....	5-14
Documentation, GUI (Indramat default)	2-31
DOS	8-2
Doskey	8-4
DP	4-14
Droop Compensation	6-49
DW_DINT	5-21

E

E^.....	4-21
Edge evaluation.....	5-40
Editing, GUI (Indramat default).....	2-25
EEV.....	4-19
Effective ACCELERATION.....	4-28
Encoder Res., Spdl.....	3-9
Encoder Resolution.....	3-5
EQ	
Operation.....	5-16
EQ	
Instruction.....	5-8
Equal to	
Instruction.....	5-8
Error handling.....	5-54
Error Handling in User Files, SPS	5-56
Error handling of the function blocks for serial interfaces	5-52
Error handling of the time-to-integer conversion.....	5-28
Error handling sequence, SPS.....	5-55
Error handling, SPS	5-54
Errors and their causes, SPS	5-56
Errors in functions, SPS.....	5-57
Errors in operations and IL instructions, SPS.....	5-62
E-Stop (Machine).....	6-2
E-Stop condition (1...3).....	6-2
E-Stop PB (Station)	6-2
Events	
Branch on NC-event to Sub.....	4-19
Cancel NC-event Supervision.....	4-19
Disable NC-event Supervision.....	4-19
Enable NC-event Supervision.....	4-19
Jump on Event	4-19
Reset NC-Event	4-18
Set NC-Event.....	4-18
Wait until NC-event is Set.....	4-18
Wait until NC-event Reset.....	4-18

Exiting Menus, GUI (Indramat default).....	2-4
Exponential.....	4-21
External 24V.....	6-2
external mechanisms.....	3-1

F

F_TRIG.....	5-41
Fault in Mechanism	8-1
Fault List.....	8-2
FAX.....	4-25
FB.....	5-39
Feed Hold	6-12
Feed Override	6-18
Feed to Positive Stop.....	6-43
Feedback Location.....	3-8
Feedback System	6-49
Feedrate	3-6
Feedrate (rotary)	3-12
Feedrate, Max chg	3-6
Feedrate, Max chg (rotary)	3-12
Filter, Command Position.....	6-43
FIND.....	5-38
Firmware function blocks	5-53
Firmware functions.....	5-38
Flip-Flops	5-39
Free Axis	4-25
Full Depth (Process).....	6-23
Funct. for Digital Drives.....	4-22
Function block calls.....	5-4
Function blocks	5-39
Functions	5-17
Functions for time-to-integer conversion.....	5-27
Further Information to Diagnostics, GUI (Indramat default)	2-16

G

G0.....	4-4
G0 Active.....	6-17
G00.....	6-17
G08.....	6-17
G1.....	4-4
G15.....	3-2, 4-5
G16.....	3-2, 4-5
G17.....	3-2, 4-5
G18.....	3-2, 4-5
G19.....	3-2, 4-5
G2.....	4-4, 4-9
G20.....	4-5
G3.....	4-4, 4-9
G30.....	4-6
G31.....	4-6
G32.....	4-6
G33.....	4-6, 6-17
G36.....	3-4, 4-6
G37.....	3-4, 4-6
G38.....	3-4, 4-6
G4.....	4-4
G40.....	4-7

G41	4-7	Feedrate	4-10, 4-11
G42	4-7	Feedrate	4-10
G43	4-7	Floating Tapping	4-8
G44	4-7	Free Plane Selection	4-5
G47	4-7	Homming Cycle	4-9
G48	4-7	Inch Prog.	4-9
G49	4-7	Incremental Values	4-10
G50	4-7	Linear Inter.	4-4
G51	4-7	Look Ahead Function, OFF	4-27
G52	4-8	Look Ahead Function, ON	4-27
G53	4-8	Machine Zero	4-8
G54	3-2, 4-8	Metric Prog.	4-9
G55	4-8	Modal Function, OFF	4-27
G56	4-8	Modal Function, ON	4-27
G57	4-8	NC-Block Restart/Re-Positioning	4-9
G58	4-8	Negative Direction, with Modulo Calculation	4-6
G59	3-2, 4-8	Offset	4-7
G6	4-4	Pos w/o Lag	4-5
G61	4-8	Pos. w/Lag	4-4
G62	4-8	Positive Direction, with Modulo Calculation	4-6
G63	4-8, 6-17	Positive Stop	4-9
G64	4-8, 6-17	Positive Stop	4-9
G65	4-8, 6-17	Prog. Workpiece Zero	4-8
G66	4-8	Radius Prog.	4-5
G7	4-5	Rigid Tapping	4-8
G70	4-9	Scaling	4-10
G71	4-9	Shortest Distance, with Modulo Calculation	4-6
G74	4-9, 6-15, 6-17	Spindle Speed	4-11
G75	3-7, 4-9	Spindle Speed Limit	4-10
G76	4-9	Thread Cutting	4-6
G77	4-9	Tool Length Cor.	4-7
G8	4-5	Tool Path Comp.	4-7
G9	4-5	XY Plane	4-5
G90	4-10	YZ-Plane	4-5
G91	4-10	Zero Point	4-8
G92	4-10	ZX Plane	4-5
G93	4-10	GE	
G94	4-10	Operation	5-15
G95	4-11	GE	
G96	4-11, 6-17	Instruction	5-8
G97	4-11	Gear Change	6-42
G98	4-11	Gearbox Input Revolutions, Main Spdl. Mode	3-13
G99	4-11	Gearbox Input, (rotary)	3-12
Gateway Signals	6-1	Gearbox Input, Axis	3-5
GAX	4-25	Gearbox Output Revolutions, Main Spdl. Mode	3-13
G-Codes	4-3	Gearbox Output, (rotary)	3-12
Absolute Dimensions	4-10	Gearbox Output, Axis	3-5
Block Trans. w/o Lag	4-8	Get Axis	4-25
Block Trans. with Lag	4-8	Global Documentation, GUI (Indramat default)	2-31
Circular Inter.	4-4, 4-9	Global Variable	4-27
Constant Feed	4-11	GRAY_BYT	5-22
Constant Surface Speed	4-11	Greater than	
Constant Wheel Surface Speed	4-8	Instruction	5-7
Contour Trans.	4-7	Greater than or equal to	5-8
Contour Trans.	4-7	GT	
Contouring	4-5	Operation	5-15
Coordinate Transformation, Cancel	4-6	GT	
Coordinate Transformation, On	4-6	Instruction	5-7
Cylinder Surface Interpolation	4-6		
Diameter Prog.	4-5		
Dwell Time	4-4		

H

H_BYTE	5-33
H_WORD	5-33
Halt, Programmed	4-15
Hardware Limits (SPS⇒CNC)	6-27
HELP SCREEN DISPLAY	8-5
HLT	4-15
Hold Motion	6-34
HOME label	3-2
Home Limit Switch	3-7, 6-29
Home Single Axis	6-30
HOME, label	4-1, 4-16
Homed Signal	6-36
homing	3-6, 3-12
Homing Dir., Analog	3-6
Homing of Feed Axes	6-17
Homing Spd, (rotary)	3-12
Homing Spd, Analog	3-6

I

iAXIS	6-52
iDEA	6-53
iMECH	6-54
Import rules	5-39, 5-53
In Pos. Window, Spdl	3-10
Incremental Jog Mode	6-8
INDRAMAT DDE Server	
Deployment Conditions	
Cooperation with other Software Components	8-13
Required Software	8-13
Deployment Conditions	8-13
INDRAMAT DDE Server	
Automatic Start	8-16
First Execution of Password Administration	8-15
Password Administration	8-15
Program Call	8-14
Register New User	8-16
User Interface	8-14
INPOS	6-38
INSERT	5-36
INT	4-20
INT_BCD	5-22
INT_BYTE	5-23
INT_DINT	5-23
INT_STR	5-23
INT_USI	5-24
INT_WORD	5-24
Integer	4-20
INTEGER-to-TIME conversion	5-29
Internal processes	3-1
Interpolation Plane, para.	3-2
Interrupting NC-Events	4-18
iPROC	6-52

J

Jerk Limitation	6-44
JEV	4-19
JMP	4-15, 5-10
Instruction	5-4
JMPC	5-10
Instruction	5-4
JMPCN	5-10
Instruction	5-4
Jog Axis in Neg.	6-31
Jog Axis in Pos.	6-30
Jog Mode Selection	6-8
Jogging	3-9, 3-13, 3-14
Jogging Rapid, Spdl	3-9, 3-13, 3-14
Jogging Spd Rapid	3-6
Jogging Speed, Axis	3-6
Jogging Speed, Spdl	3-9, 3-12, 3-14
JSR	4-16
Jump	5-10
Jump to NC-prog.	4-15
Jump to NC-prog. Sub.	4-16
Jumps	
Instruction	5-4

K

KEYBIN	8-4
Kv Factor, Analog	3-5
Kv Factor, Spdl	3-9

L

L_BYTE	5-33
L_WORD	5-33
LA_OFF	4-27
LA_ON	4-27
Labels, NC	4-2
Ladder diagram	5-9
LAG	6-47
LD	4-22, 5-2, 5-10
LDN	5-2, 5-10
LE	
Instruction	5-9
Operation	5-16
LEFT	5-34
LEN	5-33
Less than	
Instruction	5-9
Less than or equal	
Instruction	5-9
LG	4-21
Linear	3-1
Linear Encoder	3-5
linear scale	3-5
List of Droop Compensation Values	3-8
LN	4-21
Load SOT Messages, GUI (Indramat default)	2-5

Lock Process	6-20
Logarithm.....	4-21, 4-22
Logarithm.....	4-21
LOGLIST.EXE	8-2
Look Ahead.....	4-27
LP	4-14
LT	
Instruction.....	5-9
LT	
Operation.....	5-17

M

M00,M001,M002,M030	4-13
M13,M113,M213,M313	4-13
M14,M114,M214,M314	4-13
M19.....	6-33
M19,M119,M219,M319	4-13
M3.....	6-32
M3,M103,M203,M303	4-13
M4.....	6-33
M4,M104,M204,M304	4-13
M5.....	6-33
M5,M105,M205,M305	4-13
MA.....	3-7
Machine Parameter Archive, GUI (Indramat default).....	2-12
Machine Parameter Print Out, GUI (Indramat default).....	2-33
Machine Parameter, GUI (Indramat default).....	2-25
MAG_ACT	7-37
MAG_Q	7-37
Main Contactor.....	6-3
Main Spindle.....	3-1
MAKETIME.....	5-29
Management process.....	3-1, 6-19
Management Programs	4-2
Manual Data Input	6-16
MANUAL MODE	6-8
Mathematical Expressions	4-19
Max. Operating Speed of Incr. Encoder.....	3-10
Max. Spindle Speed, Gear Range (1..4).....	3-11, 3-14
Maximum Spindle Acceleration of Gear Range (1..4)	3-11, 3-15
MDI	6-16
Measuring Function	6-46
Mechanism Number.....	6-27
Mechanism Program Number	6-27
MEN	7-23
Message for MUI, NC.....	4-2
Message for SOT	4-3
Messages.....	8-1
METB	4-27
MFP	7-22
M-function of functional group 10	3-4
M-function of functional group 14	3-4
M-function of functional group 15	3-4
M-function of functional group 5	3-4
M-function of functional group 6	3-4
M-function of functional group 7	3-4
M-function of functional group 8	3-4
M-function of functional group 9	3-4

MHP	7-20
MID	5-35
Min. Spindle Speed, Gear Range (1..4).....	3-11, 3-14
<u>Minimum Execution Time for a Block</u>	4-27
MMP	7-21
MMV	7-28
MMV_Q.....	7-28
MOD	
Instruction	5-7
MOD	
Operation	5-14
MODBUS.....	5-51
MODF_OFF.....	4-27
MODF_ON.....	4-27
Monitoring Window	3-7, 6-48
Monitoring Window, Spdl.....	3-9
MOP	7-22
motion processes	3-1
Motor Direction, Spindle.....	3-13
Motor Overtemperature	6-28
Motor Speed, Spdl.....	3-9
Motor. Spd., Analog.....	3-5
MOUSE.....	8-4
MRF	7-20, 7-27
MRF_Q	7-27
MRY.....	7-22
MS-DOS.....	8-2
MTP	7-21
MT-RUN.BAT	8-2, 8-5
MUL	
Instruction	5-6
Multiple Errors, SPS	5-56
MxxC.POK.....	6-26
MxxC.QP	6-26
MxxS.AP	6-26
MxxS.DP	6-26
MxxS.LP	6-26
MxxS.PRGNR.....	6-27
MxxS.PROC.....	6-27
MxxS.RP	6-26

N

N<MIN	6-38
N=CMD.....	6-34, 6-37
N=MAX	6-38
NC	5-10
NC Memory.....	4-1
NC Program Structure	4-1
NC-Block Elements.....	4-2
NC-Events	4-18
NC-Functions	4-20
NC-memory A	4-1
NC-memory B	4-1
NC-program (w. Tool Setup List) Archive, GUI (Indramat default)	2-13
NC-Program Number	6-11
NC-program Print Out, GUI (Indramat default).....	2-32
NC-Program Selection.....	6-10
NC-Variables.....	4-22

NC-variables, GUI (Indramat default)..... 2-19
 NE
 Instruction..... 5-8
 Operation..... 5-16
 NMP..... 4-22
 NO..... 5-10
 Not equal to
 Instruction..... 5-8
 NT 3.51..... 8-2
 Number of associated axis (droop comp.)..... 3-8
 Number of Droop Compensation Points..... 3-8
 Number of Spindle Gear Ranges..... 3-11, 3-14
 Numeric functions..... 5-27

O

Off-line Mode, GUI (Indramat default)..... 2-2
 Offset Table Data, GUI (Indramat default)..... 2-17
 Offset table Selection..... 4-24
 Offset Tables..... 4-23
 Offset Tables, Max #..... 3-2
 Online Mode, GUI (Indramat default)..... 2-2
 Open Loop, Spindle..... 3-9
 OPEN_COM..... 5-47
 OPEN_SOT..... 5-50
 Operating Modes, CNC..... 6-5
 Operations..... 5-11
 Operators, NC..... 4-20
 OR
 Instruction..... 5-3
 Operation..... 5-11
 OR(
 Instruction..... 5-3
 Orient Spd, Spdl..... 3-14
 ORN
 Instruction..... 5-3
 ORN(
 Instruction..... 5-3
 Oscilloscope, GUI (Indramat default)..... 2-20
 OTD..... 4-24
 Override, Axis..... 6-31
 Overtemp. Power Supply..... 6-3
 Overtravel Switch..... 6-27

P

P factor for Spindle Synchronization..... 6-45
 P ≥ x..... 6-38
 P-7-3514..... 6-50
 P-7-3560..... 6-39
 P-7-3561..... 6-39
 P-7-3562..... 6-39
 P-7-3563..... 6-40
 P-7-3564..... 6-40
 P-7-3565..... 6-47
 P-7-3566..... 6-40
 P-7-3567..... 6-40
 P-7-3568..... 6-41
 P-7-3569..... 6-41

P-7-3570..... 6-43
 P-7-3571..... 6-48
 P-7-3572..... 6-42
 P-7-3573..... 6-42
 P-7-3574..... 6-42
 P-7-3575..... 6-42
 P-7-3576..... 6-43
 P-7-3577..... 6-43
 P-7-3578..... 6-43
 P-7-3579..... 6-40
 P-7-358..... 6-43
 P-7-3580..... 6-40
 P-7-3581..... 6-40
 P-7-3583..... 6-44
 P-7-3584..... 6-44
 P-7-3585..... 6-48
 P-7-3586..... 6-44
 P-7-3587..... 6-44
 P-7-3588..... 6-45
 P-7-3590..... 6-45
 P-7-3591..... 6-48
 P-7-3592..... 6-47
 P-7-3593..... 6-48
 P-7-3594..... 6-49
 P-7-3595..... 6-46
 P-7-3596..... 6-46
 P-7-3598..... 6-46
 P-7-3599..... 6-41
 P-7-3600..... 6-47
 P-7-3601..... 6-47
 P-7-3602..... 6-45
 P-7-3603..... 6-47
 P-7-3604..... 6-47
 P-7-3605..... 6-49
 P-7-3606..... 6-49
 P-7-3607..... 6-49
 P-7-3608..... 6-49
 P-7-3609..... 6-49
 P-7-3610..... 6-44, 6-49
 P-7-3611..... 6-50
 P-7-3612..... 6-50
 P-7-3613..... 6-50
 P-7-3615..... 6-50
 P-7-3616..... 6-50
 P-7-3617..... 6-50
 P-7-3618..... 6-50
 P-7-3621..... 6-51
 P-7-3623..... 6-51
 P-7-3625..... 6-51
 P-7-3626..... 6-51
 Parameter Set in Drive Used for Rotary Mode..... 3-14
 Parameter test..... 6-9
 Parameter Test Mode..... 6-9
 Parametric Jog Mode..... 6-8
 Parametric Jogging, (rotary)..... 3-13
 Part Machined..... 6-26
 Part Program..... 4-2
 Password Setup, GUI (Indramat default)..... 2-3
 Path Acceleration..... 4-25
 PBL.EXE..... 6-1
 PLC (SPS) Logic Analysis, GUI ^{(1),(2),(3)} (Indramat default)..... 2-23

PLC (SPS) Logic Analysis, GUI ^{(4),(5)} (Indramat default) ..	2-24	PxxC.ESTAT	6-2, 6-4
PLC (SPS) Logic Display Setup, GUI (Indramat default) .	2-22	PxxC.ESTP1	6-2, 6-4
PMP	4-22	PxxC.ESTP2	6-2
POK	4-15	PxxC.ESTP3	6-2
Polygon Course Interpolation	6-48	PxxC.EXT24	6-2, 6-4
Position Feedback Dir.....	3-7	PxxC.FOVRD	6-18
Position Feedback Dir., Spdl	3-10	PxxC.JOGM(y).....	6-8
Position Feedback Location	3-5	PxxC.LINE	6-2, 6-4
Position Feedback Location (digital drives)	3-11, 3-14	PxxC.M001	6-13
position gain, Spdl	3-9	PxxC.MGAP	7-45
Position Polarity, (rotary)	3-12	PxxC.MGBP	7-45
Position Polarity, Analog	3-6	PxxC.MGENA	7-44
Position Strobe	4-22	PxxC.MGHOM	7-45
Position Window	3-6, 6-38	PxxC.MGMAN	7-44
Position Window, (rotary)	3-13, 3-14	PxxC.MGNEG	7-45
Positioning, Spinle.....	3-9	PxxC.MGNTC.....	7-46
Positive Stop	3-7, 6-43	PxxC.MGPOS	7-45
Power (Process) ON Req.	6-4	PxxC.MODE(y).....	6-5
Power Enable	6-4	PxxC.MODE0	6-30
Power Interrupt	6-16	PxxC.MODE1	6-30
Power is ON.....	6-5	PxxC.N_CMD	6-24, 6-34
Power ON (CNC).....	6-4	PxxC.PARAM	6-9
Power On Control	6-4	PxxC.POK	6-23
Process / Part Prog. Synch.	6-19	PxxC.POWON	6-4
Process Acknowledgment	6-22	PxxC.PRGNR	6-10, 6-27
Process Control Commands	4-14	PxxC.PWRDY	6-3, 6-5
Advance Process	4-14	PxxC.QP	6-23
Define Process	4-14	PxxC.RAPID	6-13
Full Depth.....	4-15	PxxC.REPOS	6-13, 6-17
Lock Process.....	4-14	PxxC.RESTA	6-13, 6-17
Reverse Process	4-14	PxxC.REV	6-6, 6-7, 6-12
Select Process	4-14	PxxC.ROVRD	6-19
Wait for Process	4-14	PxxC.RP	6-22
Process Control Signals	6-11	PxxC.SCONn	6-24
Process Enable,	6-11	PxxC.SINGL	6-11, 6-12
Process Error.....	6-14	PxxC.SOVRD	6-18
Process Number	6-17	PxxC.SP	6-11
process parameters	3-2	PxxC.SPHLT	6-23, 6-35
Process Program	4-2	PxxC.SPSTP	6-23, 6-35
Process Signals, Gateways	6-2	PxxC.SSnMT.....	6-24
Process Status Signals.....	6-14	PxxC.SSnON.....	6-25
Process Stop.....	6-12	PxxC.STOP	6-11, 6-12
Program Active	6-15	PxxC.TPSUP	6-3, 6-4
Program Selection (NC).....	6-10	PxxC.UDRDY	6-3, 6-5
Program Stopped	6-16	PxxS.ACTIV	6-6, 6-7, 6-15
Programming Units / Table Revolution	3-5, 3-12	PxxS.AP	6-21
<i>PxC.ROVRD</i>	4-4	PxxS.CREST	6-17
Pxx(y).AP	6-21	PxxS.DP	6-19, 6-26
Pxx(y).DP	6-19	PxxS.ERROR	6-4, 6-11, 6-12, 6-14, 6-30
Pxx(y).POK	6-23	PxxS.G00	6-17
Pxx(y).QP	6-22	PxxS.G08	6-17
Pxx(y).RP.....	6-22	PxxS.G74	6-17
PxxC.ADV	6-6, 6-7, 6-11	PxxS.G96	6-17
PxxC.AP	6-21	PxxS.LP.....	6-20, 6-26
PxxC.BBMSP	6-3, 6-5	PxxS.MDIAC	6-16
PxxC.BBSUP	6-3, 6-4	PxxS.MGBP	7-46
PxxC.BLSKP	6-13	PxxS.MGCP	7-47
PxxC.CLEAR	6-14	PxxS.MGENA	7-46
PxxC.DP	6-19	PxxS.MGERR	7-48
PxxC.EMACH	6-2, 6-4	PxxS.MGMAN	7-46
PxxC.ENABL	6-11, 6-30	PxxS.MGMOV	7-46

PxxS.MGREQ	7-46
PxxS.MGTWO	7-6, 7-47
PxxS.MGWRN	7-47
PxxS.POK	6-23
PxxS.POWEN	6-4
PxxS.POWER	6-5, 6-11, 6-12, 6-30
PxxS.POWIN	6-16
PxxS.POWON	6-4, 6-5
PxxS.PRGNR	6-10, 6-27
PxxS.PROC	6-17
PxxS.QP	6-23
PxxS.READY	6-6, 6-7, 6-11, 6-15
PxxS.REPOS	6-17
PxxS.REV	6-6, 6-7, 6-11, 6-16
PxxS.RP	6-22
PxxS.RUN	6-15, 6-30
PxxS.SCONn	6-25
PxxS.SSnER	6-25
PxxS.SSnOK	6-25
PxxS.STOP	6-7, 6-16, 6-30
PxxS.TRANS	6-16
PxxS.TREAD	6-17

Q

qAXIS	6-55
qDEA	6-56
Q-Function	4-14
qPROC	6-55

R

R_TRIG	5-40
Rapid Jog Spindle	6-34
Rapid Override	6-19
Rapid Traverse	6-13, 6-17
RD_BYTE	5-48
RD_STR	5-49
RE	4-18
Ready for Start	6-15
Real Time Status Bit #2	6-37
Ref. Position, (rotary)	3-12
Ref. Position, Analog	3-6
Reference Offset, Main Spdl	3-13
Reference Required	6-15
relay	5-10
REPLACE	5-38
REPOS	6-13
Repositioning	6-13, 6-17
Repositioning / Restart Active	6-17
RES	5-2, 5-10
RESC	5-3, 5-10
RESCN	5-3, 5-10
reset	5-10
Restart	6-17
Restarting	6-13
Restore, GUI (Indramat default)	2-15
Results as	
RET	4-15, 5-10

Instruction	5-4
RETC	5-10
Instruction	5-4
RETCN	5-10
Instruction	5-4
Return to-prog. Begin	4-15
REV	4-16
REV_SYNC	7-39
Reverse Process	6-22
Reverse Prog. Active	6-16
Reverse Prog. Exec. Req.	3-2
Reverse Program	4-1, 6-7, 6-12, 6-15
Reverse Program Start	6-6, 6-26
Reverse Required	3-2
Reverse Vector	4-16
Reverse vectors	4-1
RIGHT	5-34
ROL_BYTE, ROL_WORD, ROL_DW	5-31
ROR_BYTE, ROR_WORD, ROR_DW	5-32
Rotary	3-1
Rotary Axes Mode Default	3-4
rotary encoder	3-5
RP	4-14
RS - flip-flop	5-40
RTS	4-16

S

S-0-0047	6-38
S-0-0057	6-38
S-0-0124	6-38
S-0-0158	6-38
S-0-0330	6-37
SBCD_INT	5-24
SCARA Robots	6-39
SE	4-18
Semi-Auto. & Man Mode, PxxC.ADV	6-11
SEMI-AUTOMATIC MODE	6-7
SERCOS Real Time Control Bit #2	6-31
Serial interfaces	5-45
SET	5-2, 5-10
SET TEMP	8-4
SETC	5-2, 5-10
SETCN	5-2, 5-10
Setup	6-8
SHL_BYTE, SHL_WORD, SHL_DW	5-29
SHR_BYTE, SHR_WORD, SHR_DW	5-30
SIGN_INT	5-27
Simulation Mode	6-48
SIN	4-20
Sine	4-20
Single Block Mode	6-12
Skip NC-block	6-13
SMARTDRV	8-4
Smoother Time Constant	6-44
Software Travel Limits Immediately Active	3-8
SP	4-14
SPDL_ACT	7-38
SPDL_Q	7-38
SPF	4-23

Complete Tool Chg.....	7-29	Perform Complete Tool Chg.....	7-23
Complete Tool Chg. Ack.....	7-29	Position Tool Storage.....	7-28
Correction Type 1.....	7-5	Program Specific.....	7-14
Correction Type 2.....	7-5	Radius Offset.....	7-17
Correction Type 3.....	7-5	Radius Tool Wear Factor.....	7-3
Correction Type 4.....	7-5	Radius Wear.....	7-17
Correction Type 5.....	7-5	Read NC-Command TSM.....	7-31
D-Corrections.....	7-44	Read Spindle Mode.....	7-38
Defined Plan.....	7-18	Read Turret Mode.....	7-37
Des. of User Edge Data.....	7-16	Ref. Tool Storage.....	7-27
Des. of User Tool Data.....	7-15	Setup List Organization.....	7-3
Disable Tool Trans. Error.....	7-34	Spindle / Turret Axis.....	7-37
Enable Tool Storage.....	7-44	Spindle Selection.....	7-20
Endless Rotating Tool Storage.....	7-16	Spindle to Magazine.....	7-24
Gateway Control Signals.....	7-44	SPS Control Functions.....	7-26
Gateway Signals.....	7-44	Station Specific.....	7-14
Gateway Status Signals.....	7-46	Status.....	7-8
Geometric Data.....	7-7	Symbol for User Tool bit.....	7-16
Geometric Registers.....	7-7	Symbol for User Tool Edge.....	7-16
Half Pocket Overlap.....	7-5	Time Unit.....	7-2
Home Tool Strg.....	7-45	T-Number.....	7-8
Index.....	7-8	T-Number (T).....	7-1
Length Offset.....	7-17	Tool Chg.....	7-20
Length Unit.....	7-2, 7-6	Tool Chg. Between Mag and Spd.....	7-30
Length Wear.....	7-17	Tool Chg. Functions.....	7-29
Location.....	7-8	Tool Correction.....	7-18
Mag. to Spd. Neg. Ack.....	7-33	Tool Correction Type.....	7-5
Mag. to Spd. Pos. Ack.....	7-32	Tool Data.....	7-42
Mag. to Spd. Transfer.....	7-32	Tool Data Active List.....	7-8, 7-9
Mag. to Spd. Transfer Ack.....	7-33	Tool Data Read / Write.....	7-25
Mag. to Spd. Transfer Cancel.....	7-34	Tool Edge Data.....	7-2, 7-41
Magazine axis.....	7-17	Tool Edge Data Active List.....	7-10
Magazine to Spindle.....	7-23, 7-30	Tool Edge Orientation.....	7-2, 7-6
Man. Tool Strg. Mode Status.....	7-46	Tool Edge Sel.....	7-19
Manual Tool Strg. Mode.....	7-44	Tool Edge Status.....	7-13
Max. Amount of Edges.....	7-14	Tool Edges.....	7-1
Maximum and Minimum Tool Length.....	7-2	Tool Geometry Limits.....	7-15
Maximum and Minimum Tool Radius.....	7-2	Tool Index.....	7-4
Maximum Tool Life.....	7-2	Tool Length Correction.....	7-7
Move Free Pocket into Pos.].....	7-22	Tool Life.....	7-2, 7-6
Move Loc. into Pos.....	7-21	Tool Life Data.....	7-15
Move Old Pocket into Pos.....	7-22	Tool List.....	7-4
Move to Home Pos.....	7-20	Tool List Data.....	7-4
Move to Reference Pos.....	7-20	Tool Lists and Setup Lists.....	7-1
Move Tool into Pos.....	7-21	Tool Location.....	7-4
Move Tool Storage.....	7-28	Tool Management.....	7-16
Move Tool Strg. Neg.....	7-45	Tool Name.....	7-4
Move Tool Strg. Pos.....	7-45	Tool Name (ID).....	7-1
Name.....	7-8	Tool Pocket Empty.....	7-24
NC-Commands, Tool Change.....	7-23	Tool Radius Correction.....	7-7
NC-Commands, Tool Storage.....	7-20	Tool Selection.....	7-19
NC-Control Functions.....	7-18	Tool Setup List.....	7-1
number of grippers.....	7-17	Tool Spindle Empty.....	7-24
number of spindles with tool.....	7-17	Tool Status.....	7-11
number of Tool Locations in Storage.....	7-17	Tool Storage.....	7-4
Offset Register.....	7-14	Tool Storage / Tool Error.....	7-48
Offset Registers.....	7-7	Tool Storage Dry.....	7-22
offset to reference.....	7-17	Tool Storage Enable, Man Mode.....	7-23
Old Location.....	7-5	Tool Storage Ref.....	7-27
Organization of Setup List.....	7-14	Tool Storage, Type.....	7-16
Paramters.....	7-14	Tool Strg. Enabled.....	7-46

Tool Strg. in Base Pos.	7-46
Tool Strg. Moving	7-46
Tool Strg. to Base Pos.	7-45
Tool Transfer Functions	7-32, 7-35
Tool Warning Limit.....	7-47
Tool Worn Out	7-47
User Edge Data.....	7-7
User Tool Data (1..9).....	7-6
Warning Limit	7-6
Wear Factor (L1, L2, L3)	7-3
Wear Factor Register	7-14
Wear Factors.....	7-15
Wear Registers.....	7-7
Tool <u>R</u> adius <u>C</u> orrection for Look Ahead.....	4-28
Tool Technology	7-16
Torque.....	6-47
Torque of the Digital Drive.....	6-43
Torque, measured.....	6-46
Torsion.....	6-24
Total MT-CNC User Programs & Data backup, GUI (Indramat default).....	2-6
Total MT-CNC User Programs & Data Restore, GUI (Indramat default).....	2-15
TP	5-43
TPE	7-24
Translation Active.....	6-16
Travel Limits.....	3-5
TRC	4-28
Troubleshooting and Diagnostics, GUI (Indramat default)	2-16
TSE	7-24
TSM	7-24, 7-31
TSM_Q	7-31
TSRPG25I.....	8-5
Turret	3-1
Turret Axis.....	3-11

U

UD	6-3
UDI_DINT.....	5-25
Unit of Droop Compensation Values.....	3-8
Units, default system.....	3-2
User function blocks	5-53
User functions.....	5-38
User Programs & Data - Backup.....	1-1

User Programs & Data - Restore	1-3
USI_BYTE	5-25
USI_INT.....	5-25
USINT.....	6-18

V

<u>V</u> elocity <u>F</u> actor for <u>B</u> lock <u>T</u> ransition.....	4-28
Velocity Optimized Block Transition.....	6-17
Velocity, Path.....	3-2
VFBT.....	4-28

W

Way Point Signals	6-37
Waypoints.....	3-7, 3-13
WER	4-18
WES	4-18
Window for Re-positioning and NC-block reset	3-9, 3-15
Windows NT 3.51	8-2
WORD_INT	5-26
WP.....	4-14
WR_BYTE	5-48
WR_STR	5-48

X

XFER_CHK	7-34
XMS	7-32
XMS_CA.....	7-34
XMS_NA	7-33
XMS_PA	7-32
XMS_Q	7-33
XOR	
Instruction	5-3
Operation	5-12
XOR(
Instruction	5-3
XORN	
Instruction	5-3
XORN(
Instruction	5-3

Customer Service

Americas (Canada, Latin America, and United States)

<p>Argentina</p> <p>Mannesmann Rexroth S.A.I.C. Division INDRAMAT Acassusso 48 41/7 1605 Munro (Buenos Aires) Argentina</p> <p>Phone: 01/756 01 40 01/756 02 40 Telex: 262 66 rexro ar Fax: 01/756 01 36</p>	<p>Argentina</p> <p>Nakase Asesoramiento Tecnico Diaz Velez 2929 1636 Olivos (Provincia de Buenos Aires) Argentina</p> <p>Phone: 01/790 52 30</p>	<p>Brazil</p> <p>Mannesmann Rexroth Automação Ltda. Divisão INDRAMAT Rua Georg Rexroth, 609 Vila Padre Anchieta BR-09.951-250 Diadema-SP Caixa Postal 377 BR-09.901-970 Diadema-SP</p> <p>Phone: 011/745 90 65 011/745 90 70 Fax: 011/745 90 50</p>	<p>Canada</p> <p>Basic Technologies Corporation Burlington Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8</p> <p>Phone: 905/335-55 11 Fax: 905/335-41 84</p>
<p>Mexico</p> <p>Motorización y Diseño de Controles, S.A. de C.V. Av. Dr. Gustavo Baz No. 288 Col. Parque Industrial la loma Apartado Postal No. 318 54060 Tlalnepantla Estado de Mexico Mexico</p> <p>Phone: 5/397 86 44 Fax: 5/398 98 88</p>	<p>USA</p> <p>Rexroth Corporation INDRAMAT Division 5150 Prairie Stone Parkway Hoffman Estates, Illinois 60192</p> <p>Phone: 847/645-36 00 Fax: 857/645-62 01</p>	<p>USA</p> <p>Rexroth Corporation INDRAMAT Division 2110 Austin Avenue Rochester Hills, Michigan 48309</p> <p>Phone: 810/853-82 90 Fax: 810/853-82 90</p>	<p>INDRAMAT Service Hotlines</p> <p>USA: 1-800-860-1055</p> <p>Canada: 1-847-645-3600 (request Service Department)</p> <p>Latin America: (US Phone No.) 847-645-3600 (request Service Department)</p>

Customer Service in the Americas

Asia, Australia, and Far East

<p>Australia</p> <p>Australian Industrial Machinery Services Pty. Ltd. Unit 3/45 Horne ST Campbellfield VIC 2061 Australia</p> <p>Phone: 03/93 59 0228 Fax: 03/93 59 02886</p>	<p>China</p> <p>Rexroth (China) Ltd. Shanghai Office Room 206 Shanghai Intern. Trade Centre 2200 Yanan Xi Lu Shanghai 200335 P.R. China</p> <p>Phone: 021/627 55 333 Fax: 021/627 55 666</p>	<p>China</p> <p>Rexroth (China) Ltd. Shanghai Parts & Service Centre 199 Wu Cao Road, Hua Cao Minhang District Shanghai 201 103 P.R. China</p> <p>Phone: 021/622 00 058 Fax: 021/622 00 068</p>	<p>China</p> <p>Rexroth (China) Ltd. 1430 China World Trade Centre 1, Jianguomenwai Avenue Beijing 100004 P.R. China</p> <p>Phone: 010/50 50 380 Fax: 010/50 50 379</p>
<p>China</p> <p>Rexroth (China) Ltd. A-5F., 123 Lian Shan Street Sha He Kou District Dalian 116 023 P.R. China</p> <p>Phone: 0411/46 78 930 Fax: 0411/46 78 932</p>	<p>Hong Kong</p> <p>Rexroth (China) Ltd. 19 Cheung Shun Street 1st Floor, Cheung Sha Wan, Kowloon, Hong Kong</p> <p>Phone: 741 13 51/-54 und 741 14 30 Telex: 3346 17 GL REX HX Fax: 786 40 19 786 07 33</p>	<p>India</p> <p>Mannesmann Rexroth (India) Ltd. INDRAMAT Division Plot. 96, Phase III Peenya Industrial Area Bangalore - 560058 India</p> <p>Phone: 80/839 21 01 80/839 73 74 Telex: 845 5028 REXB Fax: 80/839 43 45</p>	<p>Japan</p> <p>Rexroth Co., Ltd. INDRAMAT Division I.R. Building Nakamachidai 4-26-44 Tsuzuki-ku, Yokohama 226 Japan</p> <p>Phone: 045/942-72 10 Fax: 045/942-03 41</p>
<p>Korea</p> <p>Rexroth-Seki Co Ltd. 1500-12 Da-Dae-Dong Saha-Gu, Pusan, 604-050 Korea</p> <p>Phone: 051/264 90 01 Fax: 051/264 90 10</p>	<p>Korea</p> <p>Seo Chang Corporation Ltd. Room 903, Jeail Building 44-35 Yoide-Dong Youngdeungpo-Ku Seoul, Korea</p> <p>Phone: 02/780-82 07 ~9 Fax: 02/784-54 08</p>		

Customer Service in Asia, Australia, and the Far East

<p>Austria</p> <p>G.L.Rexroth Ges.m.b.H. Geschäftsbereich INDRAMAT Hägelingasse 3 A-1140 Wien</p> <p>Phone: 1/9852540-400 Fax: 1/9852540-93</p>	<p>Austria</p> <p>G.L.Rexroth Ges.m.b.H. Geschäftsbereich INDRAMAT Randlstraße 14 A-4061 Pasching</p> <p>Phone: 07229/4401-36 Fax: 07229/4401-80</p>	<p>Belgium</p> <p>Mannesmann Rexroth N.V.-S.A. INDRAMAT Division Industrielaan 8 B-1740 Ternat</p> <p>Phone: 02/5823180 Fax: 02/5824310</p>	<p>Denmark</p> <p>BEC Elektronik AS Zinkvej 6 DK-8900 Randers</p> <p>Phone: 086/447866 Fax: 086/447160</p>
<p>England</p> <p>Mannesmann Rexroth Ltd. INDRAMAT Division 4 Esland Place, Love Lane Cirencester, Glos GL7 1YG</p> <p>Phone: 01285/658671 Fax: 01285/654991</p>	<p>Finland</p> <p>Rexroth Mecman OY Riihimiehentie 3 SF-01720 Vantaa</p> <p>Phone: 0/848511 Fax: 0/846387</p>	<p>France</p> <p>Rexroth - Sigma S.A. Division INDRAMAT Parc des Barbanniers 4, Place du Village F-92632 Gennevilliers Cedex</p> <p>Phone: 1/41475430 Fax: 1/47946941</p>	<p>France</p> <p>Rexroth - Sigma S.A. Division INDRAMAT 91, Bd 1 Joliot Curie F-69634 Venissieux - Cx</p> <p>Phone: 78785256 Fax: 78785231</p>
<p>France</p> <p>Rexroth - Sigma S.A. Division INDRAMAT 270, Avenue de l'ardenne F-31100 Toulouse</p> <p>Phone: 61499519 Fax: 61310041</p>	<p>Italy</p> <p>Rexroth S.p.A. Divisione INDRAMAT Via G. Di Vittoria, 1 I-20063 Cernusco S/N.MI</p> <p>Phone: 02/92365-270 Fax: 02/92108069</p>	<p>Italy</p> <p>Rexroth S.p.A. Divisione INDRAMAT Via Borgomanero, 11 I-10145 Torino</p> <p>Phone: 011/7712230 Fax: 011/7710190</p>	<p>Netherlands</p> <p>Hydraudyne Hydrauliek B.V. Kruisbroeksestraat 1a P.O. Box 32 NL-5280 AA Boxtel</p> <p>Phone: 04116/51951 Fax: 04116/51483</p>
<p>Spain</p> <p>Rexroth S.A. Centro Industrial Santiago Obradors s/n E-08130 Santa Perpetua de Mogoda (Barcelona)</p> <p>Phone: 03/718 68 51 Telex: 591 81 Fax: 03/718 98 62</p>	<p>Spain</p> <p>Goimendi S.A. División Indramat Jolastokieta (Herrera) Apartado 11 37 E-San Sebastian, 20017</p> <p>Phone: 043/40 01 63 Telex: 361 72 Fax: 043/39 93 95</p>	<p>Sweden</p> <p>AB Rexroth Mecman INDRAMAT Division Varuvägen 7 S-125 81 Stockholm</p> <p>Phone: 08/727 92 00 Fax: 08/64 73 277</p>	<p>Switzerland</p> <p>Rexroth SA Département INDRAMAT Chemin de l'Ecole 6 CH-1036 Sullens</p> <p>Phone: 021/731 43 77 Fax: 021/731 46 78</p>
<p>Switzerland</p> <p>Rexroth AG Geschäftsbereich INDRAMAT Gewerbstraße 3 CH-8500 Frauenfeld</p> <p>Phone: 052/720 21 00 Fax: 052/720 21 11</p>	<p>Russia</p> <p>Tschudnenko E.B. Arsenia 22 153000 Ivanovo Russia</p> <p>Phone: 093/22 39 633</p>		

Customer Service in Europe

<p>Central Sales Region</p> <p>INDRAMAT GmbH Bgm.-Dr.-Nebel-Str. 2 D-97816 Lohr am Main</p> <p>Phone: 09352/40-0 Fax: 09352/40-4885</p>	<p>Eastern Sales Region</p> <p>INDRAMAT GmbH Beckerstraße 31 D-09120 Chemnitz</p> <p>Phone: 0371/3555-0 Fax: 0371/3555-230</p>	<p>Western Sales Region</p> <p>INDRAMAT GmbH Hansastraße 25 D-40849 Ratingen</p> <p>Phone: 02102/4318-0 Fax: 02102/41315</p>	<p>Northern Sales Region</p> <p>INDRAMAT GmbH Fährhausstraße 11 D-22085 Hamburg</p> <p>Phone: 040/227126-16 Fax: 040/227126-15</p>
<p>Southern Sales Region</p> <p>INDRAMAT GmbH Ridlerstraße 75 D-80339 München</p> <p>Phone: 089/540138-30 Fax: 089/540138-10</p>	<p>Southwestern Sales Region</p> <p>INDRAMAT GmbH Böblinger Straße 25 D-71229 Leonberg</p> <p>Phone: 07152/972-6 Fax: 07152/972-727</p>		<p>INDRAMAT Service Hotline</p> <p>INDRAMAT GmbH Phone: 0172/660 040 6</p> <p>-or-</p> <p>Phone: 0171/333 882 6</p>

Customer Service in Germany

READER COMMENT CARD AND REGISTRATION FORM

We welcome your evaluation of this manual. Please complete this form to help us to improve our publications and also to receive notification of revisions to this manual.

Publication: MT-CNC QUICK REFERENCE GUIDE

Publication No. IAE 74768

Part No. 610 944

Revision C, 10/97

Please indicate the degree to which each statement applies to you by circling whether you

① strongly agree, ② agree, ③ are uncertain, ④ disagree, or ⑤ strongly disagree with the statement.

- | | | | | | |
|--|---|---|---|---|---|
| 1. The manual is well organized. | ① | ② | ③ | ④ | ⑤ |
| 2. I can find the information I want. | ① | ② | ③ | ④ | ⑤ |
| 3. The index is thorough. | ① | ② | ③ | ④ | ⑤ |
| 4. The information is easy to understand. | ① | ② | ③ | ④ | ⑤ |
| 5. The manual is clearly written. | ① | ② | ③ | ④ | ⑤ |
| 6. Concepts and vocabulary are easy to understand. | ① | ② | ③ | ④ | ⑤ |
| 7. Examples are clear and helpful. | ① | ② | ③ | ④ | ⑤ |
| 8. The manual contains enough illustrations. | ① | ② | ③ | ④ | ⑤ |
| 9. Layout and format enhance the manual's usefulness. | ① | ② | ③ | ④ | ⑤ |
| 10. The quality of this manual would influence any repeat purchase decision. | ① | ② | ③ | ④ | ⑤ |
| 11. The scope of the information in the manual meets my needs. | ① | ② | ③ | ④ | ⑤ |

Name _____

Company _____

Address _____ Dept _____

City _____ State _____ Zip _____

Telephone (____)

Comments: _____

----- Tape Closed -----

From: _____

Technical Documentation Department
Rexroth Corporation/Indramat Division
5150 Prairie Stone Parkway
Hoffman Estates, Illinois 60192
USA

----- Tape Closed -----