THE FACTORY AUTOMATION COMPANY

FANUC

New multi-path BasePMC

User Manual

Series 30*i*/31*i*/32*i*/35*i*-MODEL B / B Plus
Series 0*i*-MODEL F / F Plus
Series Power Motion *i* MODEL A

Version 3.0.0

Author: Bloesch, Peter (FANUC Switzerland)

Revision: 1.0
Modified: Tuesday, 07 September 2021

Disclaimer

1 Table of Contents

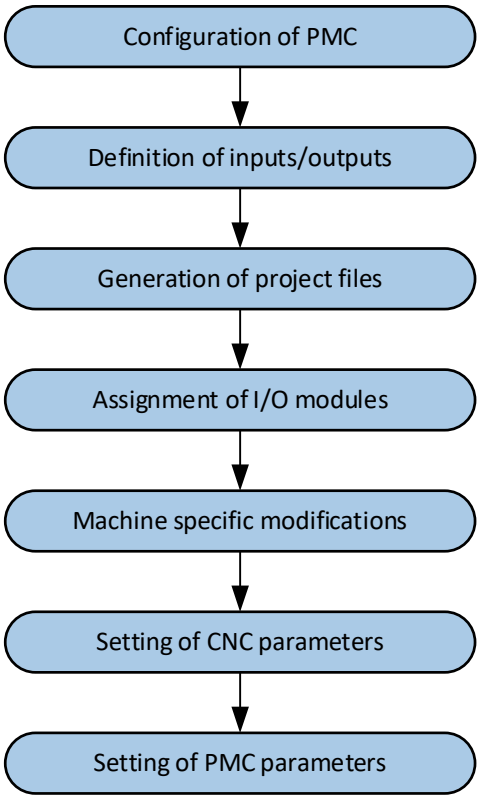
1 Table of Contents 1
2 Introduction 3
3 Configuration of the PMC software 4
4 Definitions of inputs / outputs 22
4.1 i_o.def 22
4.2 dcs_i_o.def 26
4.3 Standard Machine Operator's Panel with 55 keys 28
4.4 Virtual Operator's Panel 29
4.5 Small Machine Operator's Panel with 30 keys 31
4.6 *i*Pendant 32
4.7 Handy Machine Operator's Panel (A02B-0259-C221/C241) 33
5 Generation of project files 34
5.1 make_mem.cmd 34
5.2 make_io_mem.cmd 35
5.3 make_msg.cmd 36
5.4 make_dcs.cmd 37
6 Assignment of I/O modules 38
6.1 I/O Link *i* 38
6.1.1 Small machine operator's panel D (A02B-0338-C151#...) 38
6.1.2 Standard machine operator's panel with 55 keys) 39
6.1.3 Safety machine operator's panel 40
6.1.4 Handy Machine Operator's Panel (channel 1) 41
6.1.5 Selectable I/O Link *i* assignment 42
7 Machine specific modifications 43
7.1 Structure of MBasePMC.LAD 43
7.1.1 Modification of inputs and outputs 45
7.2 Structure of BaseDCS.LAD 49
7.2.1 Modification of inputs and outputs 50
8 Message management 54
8.1 Managing multiple languages 55
8.2 External message files 56
9 CNC parameter setting 57
9.1 Type of PMC memory 57
9.2 Communication method of I/O Link 57
9.3 M codes 58
9.4 Each axis workpiece coordinate system preset 58
9.5 Spindle (common to all spindles) 58
9.6 Serial spindle 59

9.7 Analogue spindle (spindle_support 2) 59
9.8 PMC axis used as a spindle 60
9.9 Spindle control with servo motor 61
9.10 Spindle orientation (serial spindle) 62
9.11 Reference position return 62
9.12 Message management 63
9.13 *i*Pendant 63
9.14 Dual Check Safety 64
9.15 Software operator's panel 64
9.16 Incremental feed and hand wheel 65
9.17 One Touch Macro Call 66
10 PMC parameter setting 67
10.1 Keep Relay 67
10.2 Timer 67
10.3 Data 68
10.3.1 Spindle data 68
10.3.2 Reference position return 69
11 Appendix 70
11.1 License of mcpp 70
11.2 GNU GENERAL PUBLIC LICENSE 70

2 Introduction

The library NewMP_BasePMC-x.x.x allows an easy creation of a basic multi-path PMC Ladder program. The functionality of the Software (operator's panel, spindle, number of axes etc.) will be defined in a configuration file (config.def). A part of the inputs/outputs can be adjusted in a second configuration file (i_o.def).
The command file make_mem.cmd creates a mnemonic file (MBasePMC.mne) and a symbol file (symbol.csv) with the help of a pre-processing tool and after that, the FANUC LADDER III command-line compiler generates the MBasePMC.LAD (project file for FANUC LADDER III) and MBASEPMC.MEM (memory card file that can be stored into the control).
Make_io_mem.cmd generates the IoLink_i.FIL (project file for FANUC LADDER III) and IOLINK_I.MEM (memory card file which can be stored into the control). The basic I/O configuration will be taken from the config.def file.
There is also support for Multi-language PMC Message included. make_msg.cmd creates the Multi-language PMC Message file msg_pmc1.mem from source files.
A Dual Check Safety PMC can also be created using the NewMP_BasePMC-x.x.x library. The functionality is taken from the config.def file. A part of the inputs/outputs can be adjusted in the i_o_dcs.def file.
The command file make_dcs.cmd creates a mnemonic file (MBaseDCS.mne) and a symbol file (symbol_dcs.csv) with the help of a preprocessing tool and after that the FANUC LADDER III command-line compiler generates the MBaseDCS.LAD (project file for FANUC LADDER III) and MBASEDCS.MEM (memory card file which can be stored into the control).

In FANUC LADDER III, the used I/O modules must be specified and the machine specific changes and parametrization of the control must be done.



3 Configuration of the PMC software

The file config.def in the directory NewMP_BasePMC-x.x.x will be used to configure the multi-path base PMC program. The highlighted values must be adapted to the given circumstances. The modified definition file must be stored under the same name in ASCII format.

For each definition, a valid and reasonable value must be entered. Otherwise, the project file MBasePMC.LAD cannot be created without any errors.

The following options are available:

```
// ----------------------------------------------------------------------------
// Control type
// ----------------------------------------------------------------------------
// Selection of control type
//
//    0  =  m_type
//    1  =  t_type

   #define  control_type              0


// Used for the axis key assignment on the small operator's panel with 30 keys
// A02B-0338-C151#M,#T
```

The `control_type` variable enables the selection of the accurate axis key layout when a small operator's panel with 30 keys is used.

```
// ----------------------------------------------------------------------------
// PMC type
// ----------------------------------------------------------------------------
//    1  Power Motion i-A PMC
//    2  35i-B PMC
//    3  0i-F  PMC
//    4  32i-B PMC
//    5  31i-B PMC
//    6  30i-B PMC

   #define  pmc_type                  5
```

This variable defines the PMC type.

```
// ------------------------------------------------------------------------
// Function Block
// ------------------------------------------------------------------------
//    0  no Function Block support
//    1  Function Block support

   #define  function_block             0
```

This variable adds Function Block support.

```
// ----------------------------------------------------
// PMC memory type
// ----------------------------------------------------
// 0 = Memory-B
// 1 = Memory-C
// 2 = Memory-D
// 3 = Memory-E            (30i-B nbr_of_path > 10)

   #define  pmc_mem_type               0
```

This variable defines the used memory type (see PMC PROGRAMMING MANUAL chapter PMC SPECIFICATIONS about the differences). Set parameter N11940 accordingly.

In case of 8 or more path (`nbr_of_path`), PMC memory C or D must be chosen (Relay addresses greater R7999 are used)
If the number of controlled path is 11 or more or the number of controlled axis is 81 or more, choose PMC Memory-E.

```
// ------------------------------------------------------------------------
// Number of CNC path
// ------------------------------------------------------------------------
// Selection of number of CNC path
// 1-15

   #define  nbr_of_path                3

// Note: When nbr_of_path > 8 the PMC Memory C or Memory D must be used
//       pmc_mem_type 1 or 2
//       When nbr_of_path > 10 the PMC Memory E must be selected
```

The variable `nbr_of_path` determines the number of included CNC path in the PMC.

```
// ------------------------------------------------------------------------
// Net comments
// ------------------------------------------------------------------------
// Selection if net comments should be displayed
//
//    0  =  No net comment
//    1  =  Display net comment

   #define  net_comment                1
```

The variable `net_comment` determines whether net comments are included in the PMC.

```
// ----------------------------------------------------------------------
// Demonstration control
// ----------------------------------------------------------------------
// For demonstration use, some signals will be by-passed
//
//    0  =  Normal use
//    1  =  Control is used for demonstration

   #define  demo_control              0
```

If the CNC is to be used for test purposes only, this variable allows certain signals and functions to be by-passed (e.g. Initialization, spindle monitoring, M codes, over travel, etc.).

```
// ----------------------------------------------------------------------
// Emergency stop
// ----------------------------------------------------------------------
// If the emergency stop is not wired it is possible to by-pass the signal.
//
//    0  =  No emergency stop by-pass
//    1  =  Emergency stop by-pass

   #define  emergency_by_pass         0


// Note: If the emergency_by_pass equal 1, you must define
//       start_address_op equal 0 or 1
```

If the emergency stop input (X008.4) is not yet connected, this variable allows the signal to be by-passed. When using the emergency stop by-pass, the machine operator's panel must use the address range starting at E9900/E9920 (`start_address_op` equal `0`) or X100/Y100 (`start_address_op` equal `1`)


```
// ----------------------------------------------------------------------
// Operator panel
// ----------------------------------------------------------------------
// Selection of the operator panel
//
//    0  =  Software operator's panel     (30i/31i/32i option -J960 required)
//                                        (35i-B, PMi-A Basic option 2 required)
//    1  =  Small machine operator's panel with 30 keys
//    2  =  Standard machine operator's panel with 55 keys
//    3  =  iPendant (for CNC)

   #define  operatorspanel            2
```

One of four operator panels can be selected with the variable `operatorspanel`. Functions including mode switching, axis selection, spindle operation and coolant are pre-programmed where applicable.

```
// ----------------------------------------------------------------------
// Virtual operator's panel
// ----------------------------------------------------------------------
//    0 = No virtual operator's panel with 55 keys
//    1 = Virtual operator's panel with 55 keys

   #define virtual_op_panel           0
```

In case of `operatorspanel` equal 2 and `start_address_op` equal 1 or 2 a virtual operator's panel with 55 keys for remote control can be added.

```
// ------------------------------------------
// Supplementary software operator's panel
// ------------------------------------------
//    0  =  No supplementary software operator's panel
//    1  =  Software operator's panel     (30i/31i/32i option -J960 required)
//                                        (35i-B, PMi-A Basic option 2 required)

   #define  soft_op_panel              0
```

In case of `operatorspanel` equal 1, `operatorspanel` equal 2 or `operatorspanel` equal 3, a software operator's panel can be added for remote maintenance.

```
// ----------------------------------------------------------------------
// Start addresses for I/O area of the operator panel
// ----------------------------------------------------------------------
// There are three areas to define the start adresses of the operator panel
//
//    0  =  E9900 / E9920
//    1  =  X0100 / Y0100
//    2  =  X0006 / Y0000

   #define  start_address_op           2


// Note: If the emergency_by_pass equal 1, you must define
//       start_address_op equal 0 or 1
//       In case of operatorspanel equal 0 or 3 you must define
//       start_address_op equal 0
```

For maximum flexibility, three I/O ranges are pre-defined for the machine operator's panel. The selected I/O range is automatically entered in the I/O module definition table of FANUC LADDER III or the IoLink_i.FIL file which contains the I/O definitions in case of I/O Link *i*.

```
// ----------------------------------------------------------------------
// Program protect key
// ----------------------------------------------------------------------
//    0  = No Program protect key
//    1  = Program protect key

   #define pgm_protect_key            1

// Note: This definition will be only taken into account when
// operatorspanel == 2 or start_address_op == 0
```

The safety machine operator's panel type B with 55 keys has no program protect key. With this definition, the memory protect signals can be set to logic 1 in case of `pgm_protect_key 0`, otherwise an input address will be checked.

```
// ----------------------------------------------------------------------
// Messages
// ----------------------------------------------------------------------
// Selection of language support
// (In case of 30i/31i/32i controls, one of the following options for
// display is required: External message -J911 or External data input -J913)
// (In case of 35i-B and PMi-A, basic option 2 is required)
//
//    0  =  One language supported
//    1  =  Five languages supported
//    2  =  External message file (PMC message multi-language -S977)

   #define  language_support           2


// ------------------------------------------------
// Select language for one language support
// ------------------------------------------------
//    0  =  english
//    1  =  german
//    2  =  french
//    3  =  italian
//    4  =  spanish

   #define  language_choose            0
```

Either one language, five languages or external message files are available for selection by the variable `language_support`. Each selection puts 64 CNC alarm messages (1xxx), 96 user alarm messages (20xx) and 240 operator messages (2100 - 2339) at disposal.
When five languages are selected, the "message shift function" allows the selection of each predefined language.

MESSAGE SHIFT VALUE	0	English
	400	German
	800	French
	1200	Italian
	1600	Spanish

In case of external message files, the messages are defined in the files msg_eng.txt, msg_ger.txt, msg_fre.txt, msg_ita.txt, msg_esp.txt and msg_title.txt. The batch file make_msg.cmd is used to create the msg_pmc1.mem file, which can be stored into the control.

```
// ----------------------------------------------------------------------
// M-code
// ----------------------------------------------------------------------
// Selection of M-code support
//
//    0  =  One M-code per block
//    1  =  Up to three M-codes per block
//    2  =  Up to five M-codes per block

   #define  m_code_support            1
```

Depending on the CNC parameter setting, up to five M-Functions can be defined for a single block. With the variable `m_code_support`, the corresponding M-Code decoding functionality can be included in the ladder. 400 M-Codes are at disposal (M000 – M399).

```
// ----------------------------------------------------------------------
// T-code
// ----------------------------------------------------------------------
// Selection of T-code support
//
//    0  =  No support
//    1  =  T-code

   #define  t_code_support_P01        1        // T-code support path 01
   #define  t_code_support_P02        0        // T-code support path 02
   #define  t_code_support_P03        0        // T-code support path 03
   #define  t_code_support_P04        0        // T-code support path 04
   #define  t_code_support_P05        0        // T-code support path 05
   #define  t_code_support_P06        0        // T-code support path 06
   #define  t_code_support_P07        0        // T-code support path 07
   #define  t_code_support_P08        0        // T-code support path 08
   #define  t_code_support_P09        0        // T-code support path 09
   #define  t_code_support_P10        0        // T-code support path 10
   #define  t_code_support_P11        0        // T-code support path 11
   #define  t_code_support_P12        0        // T-code support path 12
   #define  t_code_support_P13        0        // T-code support path 13
   #define  t_code_support_P14        0        // T-code support path 14
   #define  t_code_support_P15        0        // T-code support path 15
```

With these variables, T code functionality for each path can be included.

If you set **`t_code_support_P01 -1`**, the modules for the T code support will not be included.

```
// ----------------------------------------------------------------------
// B-code
// ----------------------------------------------------------------------
// Selection of B-code support
// (In case of 3xi-B controls option 2nd auxiliary function -J920 is required)
//
//    0  =  No support
//    1  =  B-code

   #define  b_code_support_P01          -1       // B-code support path 01
   #define  b_code_support_P02          0        // B-code support path 02
   #define  b_code_support_P03          0        // B-code support path 03
   #define  b_code_support_P04          0        // B-code support path 04
   #define  b_code_support_P05          0        // B-code support path 05
   #define  b_code_support_P06          0        // B-code support path 06
   #define  b_code_support_P07          0        // B-code support path 07
   #define  b_code_support_P08          0        // B-code support path 08
   #define  b_code_support_P09          0        // B-code support path 09
   #define  b_code_support_P10          0        // B-code support path 10
   #define  b_code_support_P11          0        // B-code support path 11
   #define  b_code_support_P12          0        // B-code support path 12
   #define  b_code_support_P13          0        // B-code support path 13
   #define  b_code_support_P14          0        // B-code support path 14
   #define  b_code_support_P15          0        // B-code support path 1
```

With these variables, functionality for the "2nd auxiliary function" for each path can be included.

If you set **b_code_support_P01 -1**, the code for the B code support of all paths will not be included.

```
// ----------------------------------------------------------------------
// Spindle
// ----------------------------------------------------------------------
// Selection of spindle support
//
//    0  =  No spindle
//    1  =  Serial spindle    (3xi-B and PMi-A controls option -J850 required)
//    2  =  Analog spindle    (3xi-B controls option -J860 required)
//    3  =  Analog spindle I/O module A (12-BIT ANALOG OUTPUT MODULE ADA02A)
//    4  =  PMC axis used as a spindle (30i/31i/32i option -J804 required)
//                                     (35i-B, PMi-A: basic option 1 required)
//    5  =  Spindle control with servo motor (option -J978 required)

   #define  spindle_support_P01         1        // spindle support path 01
   #define  spindle_support_P02         5        // spindle support path 02
   #define  spindle_support_P03         0        // spindle support path 03
   #define  spindle_support_P04         0        // spindle support path 04
   #define  spindle_support_P05         0        // spindle support path 05
   #define  spindle_support_P06         0        // spindle support path 06
   #define  spindle_support_P07         0        // spindle support path 07
   #define  spindle_support_P08         0        // spindle support path 08
   #define  spindle_support_P09         0        // spindle support path 09
   #define  spindle_support_P10         0        // spindle support path 10
   #define  spindle_support_P11         0        // spindle support path 11
   #define  spindle_support_P12         0        // spindle support path 12
```

```
   #define  spindle_support_P13        0        // spindle support path 13
   #define  spindle_support_P14        0        // spindle support path 14
   #define  spindle_support_P15        0        // spindle support path 15
```

With these directives, the desired spindle type per CNC path can be added to the PMC. If no spindle support is chosen (`spindle_support 0`) all spindle related modules will not be included (SpiCtrl_Pxx, SpiRun_Pxx, SpiOri_Pxx).

If you set **`spindle_support_P01 -1`**, the code for the spindle support of all paths will not be included.

```
// In case of spindle_support_Pxx equal 4 or 5 define the related servo axis in
// spindle_sv_axis_Pxx

   #define  spindle_sv_axis_P01        1        // axis nbr. for spindle path 01
   #define  spindle_sv_axis_P02        3        // axis nbr. for spindle path 02
   #define  spindle_sv_axis_P03        1        // axis nbr. for spindle path 03
   #define  spindle_sv_axis_P04        1        // axis nbr. for spindle path 04
   #define  spindle_sv_axis_P05        1        // axis nbr. for spindle path 05
   #define  spindle_sv_axis_P06        1        // axis nbr. for spindle path 06
   #define  spindle_sv_axis_P07        1        // axis nbr. for spindle path 07
   #define  spindle_sv_axis_P08        1        // axis nbr. for spindle path 08
   #define  spindle_sv_axis_P09        1        // axis nbr. for spindle path 09
   #define  spindle_sv_axis_P10        1        // axis nbr. for spindle path 10
   #define  spindle_sv_axis_P11        1        // axis nbr. for spindle path 11
   #define  spindle_sv_axis_P12        1        // axis nbr. for spindle path 12
   #define  spindle_sv_axis_P13        1        // axis nbr. for spindle path 13
   #define  spindle_sv_axis_P14        1        // axis nbr. for spindle path 14
   #define  spindle_sv_axis_P15        1        // axis nbr. for spindle path 15
```

If a servo motor is used as a spindle axis, the variable **`spindle_sv_axis_Pxx`** defines which axis of this path is used for spindle control.

This axis must be within the number of axis specified in variable **`axis_number_Pxx`** of the related CNC path.

```
// ------------------------------------------------
// Spindle override
// ------------------------------------------------
// Selection of the spindle override
// (For Software operator's panel specify 0)
//
//    0  =  Spindle override 100%
//    1  =  Spindle override with gray coded selector

   #define  spindle_override           1
```

This variable defines the type of spindle override.

- The value 0 clamps the spindle override to 100%. For the software operator's panel and iPendant, the spindle override is automatically set to 100%.

- The value 1 binds the spindle override functionality to the Gray coded, rotary switch of the following operator panels: Small Operator's Panel (A02B-0338-C151#), Sub Panel A A02B-0236-C232, Sub Panel D A02B-0236-C244 etc.

```
// ----------------------------------------------------
// Spindle orientation
// ----------------------------------------------------
// Selection of the spindle orientation
// (In case of 3xi-B and PMi-A controls option Spindle orientation -J853 is
// required)
//
//    0  =  No spindle orientation
//    1  =  Spindle orientation (One position)
//    2  =  Spindle orientation (External stop position)

   #define  spindle_orientation_P01    1        // spindle orientation path 01
   #define  spindle_orientation_P02    0        // spindle orientation path 02
   #define  spindle_orientation_P03    0        // spindle orientation path 03
   #define  spindle_orientation_P04    0        // spindle orientation path 04
   #define  spindle_orientation_P05    0        // spindle orientation path 05
   #define  spindle_orientation_P06    0        // spindle orientation path 06
   #define  spindle_orientation_P07    0        // spindle orientation path 07
   #define  spindle_orientation_P08    0        // spindle orientation path 08
   #define  spindle_orientation_P09    0        // spindle orientation path 09
   #define  spindle_orientation_P10    0        // spindle orientation path 10
   #define  spindle_orientation_P11    0        // spindle orientation path 11
   #define  spindle_orientation_P12    0        // spindle orientation path 12
   #define  spindle_orientation_P13    0        // spindle orientation path 13
   #define  spindle_orientation_P14    0        // spindle orientation path 14
   #define  spindle_orientation_P15    0        // spindle orientation path 15
```

The variable `spindle_orientation_Pxx` selects the type of spindle orientation for a serial spindle of each CNC path.

- The value 0 causes the spindle orientation module to be excluded from the ladder
- The value 1 indicates spindle orientation to a fixed position (e.g. for tool change)
- The value 2 indicates spindle orientation to a desired position (e.g. M19 S1000 = 100.0 degrees)

```
// ----------------------------------------------------
// Rigid tapping
// ----------------------------------------------------
// Selection of rigid tapping
// (In case of 3xi-B controls option Rigid tap -J828 is required)

//    0  =  No rigid tapping
//    1  =  Rigid tapping

   #define  rigid_tapping_P01          1        // rigid tapping path 01
   #define  rigid_tapping_P02          1        // rigid tapping path 02
   #define  rigid_tapping_P03          0        // rigid tapping path 03
   #define  rigid_tapping_P04          0        // rigid tapping path 04
   #define  rigid_tapping_P05          0        // rigid tapping path 05
   #define  rigid_tapping_P06          0        // rigid tapping path 06
   #define  rigid_tapping_P07          0        // rigid tapping path 07
   #define  rigid_tapping_P08          0        // rigid tapping path 08
   #define  rigid_tapping_P09          0        // rigid tapping path 09
   #define  rigid_tapping_P10          0        // rigid tapping path 10
   #define  rigid_tapping_P11          0        // rigid tapping path 11
   #define  rigid_tapping_P12          0        // rigid tapping path 12
   #define  rigid_tapping_P13          0        // rigid tapping path 13
   #define  rigid_tapping_P14          0        // rigid tapping path 14
```

```
   #define  rigid_tapping_P15          0        // rigid tapping path 15
```

With these variables, rigid tapping can be included for a “serial spindle” and “spindle control with servo motor” for each CNC path.

```
// ----------------------------------------------------
// Cs Contour Control
// ----------------------------------------------------
// Selection of Cs Contour Control
// (In case of 3xi-B controls option Cs contouring control -J852 is required)

//    0  =  No Cs Contour Control
//    1  =  Cs Contour Control (serial spindle)

   #define  cs_contour_ctrl_P01        1        // cs contour control path 01
   #define  cs_contour_ctrl_P02        0        // cs contour control path 02
   #define  cs_contour_ctrl_P03        0        // cs contour control path 03
   #define  cs_contour_ctrl_P04        0        // cs contour control path 04
   #define  cs_contour_ctrl_P05        0        // cs contour control path 05
   #define  cs_contour_ctrl_P06        0        // cs contour control path 06
   #define  cs_contour_ctrl_P07        0        // cs contour control path 07
   #define  cs_contour_ctrl_P08        0        // cs contour control path 08
   #define  cs_contour_ctrl_P09        0        // cs contour control path 09
   #define  cs_contour_ctrl_P10        0        // cs contour control path 10
   #define  cs_contour_ctrl_P11        0        // cs contour control path 11
   #define  cs_contour_ctrl_P12        0        // cs contour control path 12
   #define  cs_contour_ctrl_P13        0        // cs contour control path 13
   #define  cs_contour_ctrl_P14        0        // cs contour control path 14
   #define  cs_contour_ctrl_P15        0        // cs contour control path 15
```

With these variables, Cs contour control can be included for a serial spindle.

```
// ----------------------------------------------------------------------------
// Coolant
// ----------------------------------------------------------------------------
//
//    0  =  No coolant
//    1  =  Coolant

   #define  coolant_support_P01        1        // coolant support path 01
   #define  coolant_support_P02        0        // coolant support path 02
   #define  coolant_support_P03        0        // coolant support path 03
   #define  coolant_support_P04        0        // coolant support path 04
   #define  coolant_support_P05        0        // coolant support path 05
   #define  coolant_support_P06        0        // coolant support path 06
   #define  coolant_support_P07        0        // coolant support path 07
   #define  coolant_support_P08        0        // coolant support path 08
   #define  coolant_support_P09        0        // coolant support path 09
   #define  coolant_support_P10        0        // coolant support path 10
   #define  coolant_support_P11        0        // coolant support path 11
   #define  coolant_support_P12        0        // coolant support path 12
   #define  coolant_support_P13        0        // coolant support path 13
   #define  coolant_support_P14        0        // coolant support path 14
   #define  coolant_support_P15        0        // coolant support path 15
```

The variable coolant_support_Pxx includes coolant support for the first coolant for each CNC path.

If you set **coolant_support_P01 -1**, the code for the coolant support of all paths will not be included.

```
// ----------------------------------------------------------------------
// Lubrication
// ----------------------------------------------------------------------
//
//    0  =  No lubrication
//    1  =  Lubrication with level and pressure control

   #define  lubrication_support        1
```

This variable determines whether the module for centralized lubrication with level and pressure control is included in the PMC. Lubrication frequency, operation time and pump time-out can be set individually with timers.

```
// ----------------------------------------------------------------------
// One Touch Macro Call
// ----------------------------------------------------------------------
// Option One Touch Macro Call -S655 is required)
//
//    0  =  No One Touch Macro Call
//    1  =  One Touch Macro Call included

   #define  otmc_support_P01           0        // omtc support path 01
   #define  otmc_support_P02           0        // omtc support path 02
   #define  otmc_support_P03           1        // omtc support path 03
   #define  otmc_support_P04           0        // omtc support path 04
   #define  otmc_support_P05           0        // omtc support path 05
   #define  otmc_support_P06           0        // omtc support path 06
   #define  otmc_support_P07           0        // omtc support path 07
   #define  otmc_support_P08           0        // omtc support path 08
   #define  otmc_support_P09           0        // omtc support path 09
   #define  otmc_support_P10           0        // omtc support path 10
   #define  otmc_support_P11           0        // omtc support path 11
   #define  otmc_support_P12           0        // omtc support path 12
   #define  otmc_support_P13           0        // omtc support path 13
   #define  otmc_support_P14           0        // omtc support path 14
   #define  otmc_support_P15           0        // omtc support path 15

// Will be only included in case of standard machine operator's panel with
// 55keys (operatorspanel 2)
// Only valid for 30i/31i/32i/0i-F controls
```

This variable determines whether "One Touch Macro Call" support is included in the PMC.

If you set **otmc_support_P01 equal -1** the code for the one touch macro call support of all paths will not be included.

```
// ----------------------------------------------------------------------
// Dual Check Safety
// ----------------------------------------------------------------------
// (Option Dual Check Satety -S661 is required)
//
//    0  =  No Dual Check Safety
//    1  =  Dual Check Safety

   #define  dcs_support                1
```

This variable determines whether "Dual Check Safety" support is included in the PMC.

```
// ------------------------------------------------
// MCC test
// ------------------------------------------------
//
//    0  =  No MCC test
//    1  =  execute MCC test

   #define  mcc_test                   1
```

This variable determines whether the code for the "Dual Check Safety MCC test" will be included in the PMC.

```
// ------------------------------------------------
// brake test
// ------------------------------------------------
//
//    0  =  No brake test
//    1  =  execute brake test

   #define  brake_test                 1
```

This variable determines whether the code for the Dual Check Safety brake test will be included in the PMC.

```
// ----------------------------------------------------------------------
// Axes
// ----------------------------------------------------------------------
// Maximum number of supported axes per path (Minimum 1 axis)
//    4  Software operator's panel
//    4  Small operator's panel with 30 keys
//    6  iPendant (for CNC)
//    8  Standard machine operator's panel

   #define  axis_number_P01            5       // number of axis path 01
   #define  axis_number_P02            4       // number of axis path 02
   #define  axis_number_P03            3       // number of axis path 03
   #define  axis_number_P04            1       // number of axis path 04
   #define  axis_number_P05            1       // number of axis path 05
   #define  axis_number_P06            1       // number of axis path 06
   #define  axis_number_P07            1       // number of axis path 07
   #define  axis_number_P08            1       // number of axis path 08
   #define  axis_number_P09            1       // number of axis path 09
   #define  axis_number_P10            1       // number of axis path 10
   #define  axis_number_P11            1       // number of axis path 11
```

```
   #define  axis_number_P12            1        // number of axis path 12
   #define  axis_number_P13            1        // number of axis path 13
   #define  axis_number_P14            1        // number of axis path 14
   #define  axis_number_P15            1        // number of axis path 15
```

With these variables, the number of axes per CNC path is set. The minimum number of axes per path is 1. The maximum number of axes per path depends on the type of the selected machine operator's panel.

```
// ------------------------------------------------
// Handwheel
// ------------------------------------------------
// Handwheel selection
// (In case of 30i/31i/32i controls option Manual handle feed 1-unit -J835
// is required)
// (In case of 35i-B and PMi-A control, basic option 1 is required)

//    0  =  No handwheel
//    1  =  One handwheel
//    2  =  Handy machine operator's panel (A02B-0259-C221/C241)
//    3  =  Portable manual pulse generator (A13B-0206-Cxyz#....)

   #define  handwheel_support          3
```

The variable `handwheel_support` causes support for a manual pulse generator to be included in the PMC.

```
// ------------------------------------------------------
// Reference position return
// ------------------------------------------------------
// (NC = Contact normally closed)
// (NO = Contact normally open)

// Reference position return procedure
//    0  =  No reference position return     // not reference return
//    1  =  Plus direction    NC             // negative axis limit
//    2  =  Minus direction   NC             // positive axis limit
//    3  =  Plus direction    NO             // negative axis limit
//    4  =  Minus direction   NO             // positive axis limit
//    5  =  Without dog plus direction
//    6  =  Without dog minus direction


   #define  ref_dir_1st_axis_P01       -1     // ref. return 1st axis path 01
   #define  ref_dir_2nd_axis_P01       0     // ref. return 2nd axis path 01
   #define  ref_dir_3rd_axis_P01       0     // ref. return 3rd axis path 01
   #define  ref_dir_4th_axis_P01       0     // ref. return 4th axis path 01
   #define  ref_dir_5th_axis_P01       0     // ref. return 5th axis path 01
   #define  ref_dir_6th_axis_P01       0     // ref. return 6th axis path 01
   #define  ref_dir_7th_axis_P01       0     // ref. return 7th axis path 01
   #define  ref_dir_8th_axis_P01       0     // ref. return 8th axis path 01

   #define  ref_dir_1st_axis_P02       -1     // ref. return 1st axis path 02
   #define  ref_dir_2nd_axis_P02       0     // ref. return 2nd axis path 02
   #define  ref_dir_3rd_axis_P02       0     // ref. return 3rd axis path 02
   #define  ref_dir_4th_axis_P02       0     // ref. return 4th axis path 02
   #define  ref_dir_5th_axis_P02       0     // ref. return 5th axis path 02
   #define  ref_dir_6th_axis_P02       0     // ref. return 6th axis path 02
   #define  ref_dir_7th_axis_P02       0     // ref. return 7th axis path 02
   #define  ref_dir_8th_axis_P02       0     // ref. return 8th axis path 02
```

```
#define  ref_dir_1st_axis_P03       -1    // ref. return 1st axis path 03
#define  ref_dir_2nd_axis_P03       0     // ref. return 2nd axis path 03
#define  ref_dir_3rd_axis_P03       0     // ref. return 3rd axis path 03
#define  ref_dir_4th_axis_P03       0     // ref. return 4th axis path 03
#define  ref_dir_5th_axis_P03       0     // ref. return 5th axis path 03
#define  ref_dir_6th_axis_P03       0     // ref. return 6th axis path 03
#define  ref_dir_7th_axis_P03       0     // ref. return 7th axis path 03
#define  ref_dir_8th_axis_P03       0     // ref. return 8th axis path 03

#define  ref_dir_1st_axis_P04       0     // ref. return 1st axis path 04
#define  ref_dir_2nd_axis_P04       0     // ref. return 2nd axis path 04
#define  ref_dir_3rd_axis_P04       0     // ref. return 3rd axis path 04
#define  ref_dir_4th_axis_P04       0     // ref. return 4th axis path 04
#define  ref_dir_5th_axis_P04       0     // ref. return 5th axis path 04
#define  ref_dir_6th_axis_P04       0     // ref. return 6th axis path 04
#define  ref_dir_7th_axis_P04       0     // ref. return 7th axis path 04
#define  ref_dir_8th_axis_P04       0     // ref. return 8th axis path 04

#define  ref_dir_1st_axis_P05       0     // ref. return 1st axis path 05
#define  ref_dir_2nd_axis_P05       0     // ref. return 2nd axis path 05
#define  ref_dir_3rd_axis_P05       0     // ref. return 3rd axis path 05
#define  ref_dir_4th_axis_P05       0     // ref. return 4th axis path 05
#define  ref_dir_5th_axis_P05       0     // ref. return 5th axis path 05
#define  ref_dir_6th_axis_P05       0     // ref. return 6th axis path 05
#define  ref_dir_7th_axis_P05       0     // ref. return 7th axis path 05
#define  ref_dir_8th_axis_P05       0     // ref. return 8th axis path 05

#define  ref_dir_1st_axis_P06       0     // ref. return 1st axis path 06
#define  ref_dir_2nd_axis_P06       0     // ref. return 2nd axis path 06
#define  ref_dir_3rd_axis_P06       0     // ref. return 3rd axis path 06
#define  ref_dir_4th_axis_P06       0     // ref. return 4th axis path 06
#define  ref_dir_5th_axis_P06       0     // ref. return 5th axis path 06
#define  ref_dir_6th_axis_P06       0     // ref. return 6th axis path 06
#define  ref_dir_7th_axis_P06       0     // ref. return 7th axis path 06
#define  ref_dir_8th_axis_P06       0     // ref. return 8th axis path 06

#define  ref_dir_1st_axis_P07       0     // ref. return 1st axis path 07
#define  ref_dir_2nd_axis_P07       0     // ref. return 2nd axis path 07
#define  ref_dir_3rd_axis_P07       0     // ref. return 3rd axis path 07
#define  ref_dir_4th_axis_P07       0     // ref. return 4th axis path 07
#define  ref_dir_5th_axis_P07       0     // ref. return 5th axis path 07
#define  ref_dir_6th_axis_P07       0     // ref. return 6th axis path 07
#define  ref_dir_7th_axis_P07       0     // ref. return 7th axis path 07
#define  ref_dir_8th_axis_P07       0     // ref. return 8th axis path 07

#define  ref_dir_1st_axis_P08       0     // ref. return 1st axis path 08
#define  ref_dir_2nd_axis_P08       0     // ref. return 2nd axis path 08
#define  ref_dir_3rd_axis_P08       0     // ref. return 3rd axis path 08
#define  ref_dir_4th_axis_P08       0     // ref. return 4th axis path 08
#define  ref_dir_5th_axis_P08       0     // ref. return 5th axis path 08
#define  ref_dir_6th_axis_P08       0     // ref. return 6th axis path 08
#define  ref_dir_7th_axis_P08       0     // ref. return 7th axis path 08
#define  ref_dir_8th_axis_P08       0     // ref. return 8th axis path 08

#define  ref_dir_1st_axis_P09       0     // ref. return 1st axis path 09
#define  ref_dir_2nd_axis_P09       0     // ref. return 2nd axis path 09
#define  ref_dir_3rd_axis_P09       0     // ref. return 3rd axis path 09
#define  ref_dir_4th_axis_P09       0     // ref. return 4th axis path 09
#define  ref_dir_5th_axis_P09       0     // ref. return 5th axis path 09
#define  ref_dir_6th_axis_P09       0     // ref. return 6th axis path 09
#define  ref_dir_7th_axis_P09       0     // ref. return 7th axis path 09
#define  ref_dir_8th_axis_P09       0     // ref. return 8th axis path 09

#define  ref_dir_1st_axis_P10       0     // ref. return 1st axis path 10
#define  ref_dir_2nd_axis_P10       0     // ref. return 2nd axis path 10
```

```
#define  ref_dir_3rd_axis_P10       0     // ref. return 3rd axis path 10
#define  ref_dir_4th_axis_P10       0     // ref. return 4th axis path 10
#define  ref_dir_5th_axis_P10       0     // ref. return 5th axis path 10
#define  ref_dir_6th_axis_P10       0     // ref. return 6th axis path 10
#define  ref_dir_7th_axis_P10       0     // ref. return 7th axis path 10
#define  ref_dir_8th_axis_P10       0     // ref. return 8th axis path 10

#define  ref_dir_1st_axis_P11       0     // ref. return 1st axis path 11
#define  ref_dir_2nd_axis_P11       0     // ref. return 2nd axis path 11
#define  ref_dir_3rd_axis_P11       0     // ref. return 3rd axis path 11
#define  ref_dir_4th_axis_P11       0     // ref. return 4th axis path 11
#define  ref_dir_5th_axis_P11       0     // ref. return 5th axis path 11
#define  ref_dir_6th_axis_P11       0     // ref. return 6th axis path 11
#define  ref_dir_7th_axis_P11       0     // ref. return 7th axis path 11
#define  ref_dir_8th_axis_P11       0     // ref. return 8th axis path 11

#define  ref_dir_1st_axis_P12       0     // ref. return 1st axis path 12
#define  ref_dir_2nd_axis_P12       0     // ref. return 2nd axis path 12
#define  ref_dir_3rd_axis_P12       0     // ref. return 3rd axis path 12
#define  ref_dir_4th_axis_P12       0     // ref. return 4th axis path 12
#define  ref_dir_5th_axis_P12       0     // ref. return 5th axis path 12
#define  ref_dir_6th_axis_P12       0     // ref. return 6th axis path 12
#define  ref_dir_7th_axis_P12       0     // ref. return 7th axis path 12
#define  ref_dir_8th_axis_P12       0     // ref. return 8th axis path 12

#define  ref_dir_1st_axis_P13       0     // ref. return 1st axis path 13
#define  ref_dir_2nd_axis_P13       0     // ref. return 2nd axis path 13
#define  ref_dir_3rd_axis_P13       0     // ref. return 3rd axis path 13
#define  ref_dir_4th_axis_P13       0     // ref. return 4th axis path 13
#define  ref_dir_5th_axis_P13       0     // ref. return 5th axis path 13
#define  ref_dir_6th_axis_P13       0     // ref. return 6th axis path 13
#define  ref_dir_7th_axis_P13       0     // ref. return 7th axis path 13
#define  ref_dir_8th_axis_P13       0     // ref. return 8th axis path 13

#define  ref_dir_1st_axis_P14       0     // ref. return 1st axis path 14
#define  ref_dir_2nd_axis_P14       0     // ref. return 2nd axis path 14
#define  ref_dir_3rd_axis_P14       0     // ref. return 3rd axis path 14
#define  ref_dir_4th_axis_P14       0     // ref. return 4th axis path 14
#define  ref_dir_5th_axis_P14       0     // ref. return 5th axis path 14
#define  ref_dir_6th_axis_P14       0     // ref. return 6th axis path 14
#define  ref_dir_7th_axis_P14       0     // ref. return 7th axis path 14
#define  ref_dir_8th_axis_P14       0     // ref. return 8th axis path 14

#define  ref_dir_1st_axis_P15       0     // ref. return 1st axis path 15
#define  ref_dir_2nd_axis_P15       0     // ref. return 2nd axis path 15
#define  ref_dir_3rd_axis_P15       0     // ref. return 3rd axis path 15
#define  ref_dir_4th_axis_P15       0     // ref. return 4th axis path 15
#define  ref_dir_5th_axis_P15       0     // ref. return 5th axis path 15
#define  ref_dir_6th_axis_P15       0     // ref. return 6th axis path 15
#define  ref_dir_7th_axis_P15       0     // ref. return 7th axis path 15
#define  ref_dir_8th_axis_P15       0     // ref. return 8th axis path 15
```

If you set **`ref_dir_1st_axis_Pxx -1`** (only 1st axis), the code for the reference return of all axes of this path will not be included.

The reference position return method is defined for each axis independently:

- 0 = Reference position return is not executed (e.g. for absolute encoder)
- 1 = The reference switch is normally closed type (NC) and the dog is positioned at the negative end of the axis stroke.

- 2 = The reference switch is normally closed type (NC) and the dog is positioned at the positive end of the axis stroke.
- 3 = The reference switch is normally open type (NO) and the dog is positioned at the negative end of the axis stroke.
- 4 = The reference switch is normally open type (NO) and the dog is positioned at the positive end of the axis stroke.
- 5 = Reference position return without dog in the positive axis direction
- 6 = Reference position return without dog in the negative axis direction

The reference position return for a single axis can be switched on / off with the corresponding keep relay Ky.x:
y=0 → path01, y=1 → path02 etc.
x=0 → 1st axis, x=1 → 2nd axis etc.).

```
// ----------------------------------------------------
// Axes brake control
// ----------------------------------------------------
// Selection of axis holding brake

// 0 = No axis brake
// 1 = axis brake

   #define  brake_1st_axis_P01          0     // axis brake 1st axis path 01
   #define  brake_2nd_axis_P01          0     // axis brake 2nd axis path 01
   #define  brake_3rd_axis_P01          1     // axis brake 3rd axis path 01
   #define  brake_4th_axis_P01          0     // axis brake 4th axis path 01
   #define  brake_5th_axis_P01          0     // axis brake 5th axis path 01
   #define  brake_6th_axis_P01          0     // axis brake 6th axis path 01
   #define  brake_7th_axis_P01          0     // axis brake 7th axis path 01
   #define  brake_8th_axis_P01          0     // axis brake 8th axis path 01

   #define  brake_1st_axis_P02          0     // axis brake 1st axis path 02
   #define  brake_2nd_axis_P02          1     // axis brake 2nd axis path 02
   #define  brake_3rd_axis_P02          0     // axis brake 3rd axis path 02
   #define  brake_4th_axis_P02          0     // axis brake 4th axis path 02
   #define  brake_5th_axis_P02          0     // axis brake 5th axis path 02
   #define  brake_6th_axis_P02          0     // axis brake 6th axis path 02
   #define  brake_7th_axis_P02          0     // axis brake 7th axis path 02
   #define  brake_8th_axis_P02          0     // axis brake 8th axis path 02

   #define  brake_1st_axis_P03          0     // axis brake 1st axis path 03
   #define  brake_2nd_axis_P03          0     // axis brake 2nd axis path 03
   #define  brake_3rd_axis_P03          0     // axis brake 3rd axis path 03
   #define  brake_4th_axis_P03          0     // axis brake 4th axis path 03
   #define  brake_5th_axis_P03          0     // axis brake 5th axis path 03
   #define  brake_6th_axis_P03          0     // axis brake 6th axis path 03
   #define  brake_7th_axis_P03          0     // axis brake 7th axis path 03
   #define  brake_8th_axis_P03          0     // axis brake 8th axis path 03

   #define  brake_1st_axis_P04          0     // axis brake 1st axis path 04
   #define  brake_2nd_axis_P04          0     // axis brake 2nd axis path 04
   #define  brake_3rd_axis_P04          0     // axis brake 3rd axis path 04
   #define  brake_4th_axis_P04          0     // axis brake 4th axis path 04
   #define  brake_5th_axis_P04          0     // axis brake 5th axis path 04
   #define  brake_6th_axis_P04          0     // axis brake 6th axis path 04
   #define  brake_7th_axis_P04          0     // axis brake 7th axis path 04
   #define  brake_8th_axis_P04          0     // axis brake 8th axis path 04
```

```
#define  brake_1st_axis_P05        0     // axis brake 1st axis path 05
#define  brake_2nd_axis_P05        0     // axis brake 2nd axis path 05
#define  brake_3rd_axis_P05        0     // axis brake 3rd axis path 05
#define  brake_4th_axis_P05        0     // axis brake 4th axis path 05
#define  brake_5th_axis_P05        0     // axis brake 5th axis path 05
#define  brake_6th_axis_P05        0     // axis brake 6th axis path 05
#define  brake_7th_axis_P05        0     // axis brake 7th axis path 05
#define  brake_8th_axis_P05        0     // axis brake 8th axis path 05

#define  brake_1st_axis_P06        0     // axis brake 1st axis path 06
#define  brake_2nd_axis_P06        0     // axis brake 2nd axis path 06
#define  brake_3rd_axis_P06        0     // axis brake 3rd axis path 06
#define  brake_4th_axis_P06        0     // axis brake 4th axis path 06
#define  brake_5th_axis_P06        0     // axis brake 5th axis path 06
#define  brake_6th_axis_P06        0     // axis brake 6th axis path 06
#define  brake_7th_axis_P06        0     // axis brake 7th axis path 06
#define  brake_8th_axis_P06        0     // axis brake 8th axis path 06

#define  brake_1st_axis_P07        0     // axis brake 1st axis path 07
#define  brake_2nd_axis_P07        0     // axis brake 2nd axis path 07
#define  brake_3rd_axis_P07        0     // axis brake 3rd axis path 07
#define  brake_4th_axis_P07        0     // axis brake 4th axis path 07
#define  brake_5th_axis_P07        0     // axis brake 5th axis path 07
#define  brake_6th_axis_P07        0     // axis brake 6th axis path 07
#define  brake_7th_axis_P07        0     // axis brake 7th axis path 07
#define  brake_8th_axis_P07        0     // axis brake 8th axis path 07

#define  brake_1st_axis_P08        0     // axis brake 1st axis path 08
#define  brake_2nd_axis_P08        0     // axis brake 2nd axis path 08
#define  brake_3rd_axis_P08        0     // axis brake 3rd axis path 08
#define  brake_4th_axis_P08        0     // axis brake 4th axis path 08
#define  brake_5th_axis_P08        0     // axis brake 5th axis path 08
#define  brake_6th_axis_P08        0     // axis brake 6th axis path 08
#define  brake_7th_axis_P08        0     // axis brake 7th axis path 08
#define  brake_8th_axis_P08        0     // axis brake 8th axis path 08

#define  brake_1st_axis_P09        0     // axis brake 1st axis path 09
#define  brake_2nd_axis_P09        0     // axis brake 2nd axis path 09
#define  brake_3rd_axis_P09        0     // axis brake 3rd axis path 09
#define  brake_4th_axis_P09        0     // axis brake 4th axis path 09
#define  brake_5th_axis_P09        0     // axis brake 5th axis path 09
#define  brake_6th_axis_P09        0     // axis brake 6th axis path 09
#define  brake_7th_axis_P09        0     // axis brake 7th axis path 09
#define  brake_8th_axis_P09        0     // axis brake 8th axis path 09

#define  brake_1st_axis_P10        0     // axis brake 1st axis path 10
#define  brake_2nd_axis_P10        0     // axis brake 2nd axis path 10
#define  brake_3rd_axis_P10        0     // axis brake 3rd axis path 10
#define  brake_4th_axis_P10        0     // axis brake 4th axis path 10
#define  brake_5th_axis_P10        0     // axis brake 5th axis path 10
#define  brake_6th_axis_P10        0     // axis brake 6th axis path 10
#define  brake_7th_axis_P10        0     // axis brake 7th axis path 10
#define  brake_8th_axis_P10        0     // axis brake 8th axis path 10

#define  brake_1st_axis_P11        0     // axis brake 1st axis path 11
#define  brake_2nd_axis_P11        0     // axis brake 2nd axis path 11
#define  brake_3rd_axis_P11        0     // axis brake 3rd axis path 11
#define  brake_4th_axis_P11        0     // axis brake 4th axis path 11
#define  brake_5th_axis_P11        0     // axis brake 5th axis path 11
#define  brake_6th_axis_P11        0     // axis brake 6th axis path 11
#define  brake_7th_axis_P11        0     // axis brake 7th axis path 11
#define  brake_8th_axis_P11        0     // axis brake 8th axis path 11
```

```
#define  brake_1st_axis_P12          0     // axis brake 1st axis path 12
#define  brake_2nd_axis_P12          0     // axis brake 2nd axis path 12
#define  brake_3rd_axis_P12          0     // axis brake 3rd axis path 12
#define  brake_4th_axis_P12          0     // axis brake 4th axis path 12
#define  brake_5th_axis_P12          0     // axis brake 5th axis path 12
#define  brake_6th_axis_P12          0     // axis brake 6th axis path 12
#define  brake_7th_axis_P12          0     // axis brake 7th axis path 12
#define  brake_8th_axis_P12          0     // axis brake 8th axis path 12

#define  brake_1st_axis_P13          0     // axis brake 1st axis path 13
#define  brake_2nd_axis_P13          0     // axis brake 2nd axis path 13
#define  brake_3rd_axis_P13          0     // axis brake 3rd axis path 13
#define  brake_4th_axis_P13          0     // axis brake 4th axis path 13
#define  brake_5th_axis_P13          0     // axis brake 5th axis path 13
#define  brake_6th_axis_P13          0     // axis brake 6th axis path 13
#define  brake_7th_axis_P13          0     // axis brake 7th axis path 13
#define  brake_8th_axis_P13          0     // axis brake 8th axis path 13

#define  brake_1st_axis_P14          0     // axis brake 1st axis path 14
#define  brake_2nd_axis_P14          0     // axis brake 2nd axis path 14
#define  brake_3rd_axis_P14          0     // axis brake 3rd axis path 14
#define  brake_4th_axis_P14          0     // axis brake 4th axis path 14
#define  brake_5th_axis_P14          0     // axis brake 5th axis path 14
#define  brake_6th_axis_P14          0     // axis brake 6th axis path 14
#define  brake_7th_axis_P14          0     // axis brake 7th axis path 14
#define  brake_8th_axis_P14          0     // axis brake 8th axis path 14

#define  brake_1st_axis_P15          0     // axis brake 1st axis path 15
#define  brake_2nd_axis_P15          0     // axis brake 2nd axis path 15
#define  brake_3rd_axis_P15          0     // axis brake 3rd axis path 15
#define  brake_4th_axis_P15          0     // axis brake 4th axis path 15
#define  brake_5th_axis_P15          0     // axis brake 5th axis path 15
#define  brake_6th_axis_P15          0     // axis brake 6th axis path 15
#define  brake_7th_axis_P15          0     // axis brake 7th axis path 15
#define  brake_8th_axis_P15          0     // axis brake 8th axis path 15
```

The variables include holding brake support for the servo axes.

4 Definitions of inputs / outputs

For the definition of the I/O addresses, the following files are available

i_o.def	for all I/O addresses other than the machine operator's panel
dcs_i_o.def	for all I/O addresses related to the Dual Check Safety PMC.
op_55keys.def	for the standard operator's panel with 55 keys (need not to be modified)
op_30keys.def	for the small operator's panel with 30 keys (need not to be modified)
i_pendant.def	for the iPendant (for CNC) (need not to be modified)
hmop.def	for the handy machine operator's panel (need not to be modified)

4.1 i_o.def

The file i_o.def in the directory NewMP_BasePMC-x.x.x includes definitions for all I/O addresses for the 1st PMC other than those for the machine operator panels.

The highlighted values must be adapted for the given machine. The modified definition file must be stored under the same name in ASCII format.

Only valid I/O addresses must be set in the definition file in order to create an error-free symbol file. Be careful not to enter duplicate addresses into the table.

```
// ---------------------------------------------------------------------------
// lubrication
// ---------------------------------------------------------------------------
#if lubrication_support > 0

   #define  __S_LEVEL_LUBRIC         X4.0
   #define  __S_PRESSURE_LUBRIC      X4.1
   #define  __OUTPUT_LUBRIC_ON       Y10.1
#endif
```

The inputs and output for the lubrication need only to be defined if lubrication support has been included in the PMC.

```
// ---------------------------------------------------------------------------
// Path switching
// in case of operatorspanel 0 (Software operator's panel) or
// operatorspanel 1 (Small machine operator's panel with 30 keys),
// define the inputs for path synchronization and path switching
// ---------------------------------------------------------------------------
#if operatorspanel < 2 && nbr_of_path > 1

   #define  __PATH_SYNC             X15.4
   #define  __PATH_SYNC_LAMP        Y11.4

   #define  __PATH_PLUS             X15.5
   #define  __PATH_PLUS_LAMP        Y11.5

   #if nbr_of_path > 2
      #define  __PATH_MINUS         X15.6
      #define  __PATH_MINUS_LAMP    Y11.6
   #endif
#endif
```

In case of `operatorspanel 0` (Software operator's panel) or `operatorspanel 1` (Small machine operator's panel with 30 keys), the I/O for the path switching keys have to be defined.

```
// ---------------------------------------------------------------------------
// portable manual pulse generator (A13B-0206-Cxyz#....)
// ---------------------------------------------------------------------------
#if handwheel_support == 3
   #define  __PMPG_SELECTOR_X1      X4.4
   #define  __PMPG_SELECTOR_X10     X4.5
   #define  __PMPG_SELECTOR_X100    X4.6
   #define  __PMPG_SELECTOR_X1000   X4.7

   #define  __PMPG_SELECTOR_AXIS_1  X5.0
   #define  __PMPG_SELECTOR_AXIS_2  X5.1
   #define  __PMPG_SELECTOR_AXIS_3  X5.2
   #define  __PMPG_SELECTOR_AXIS_4  X5.3
   #define  __PMPG_SELECTOR_AXIS_5  X5.4
   #define  __PMPG_SELECTOR_AXIS_6  X5.5

   #define  __PMPG_LED              Y10.2
#endif
```

The inputs and output for the portable manual pulse generator (A13B-0206-Cxyz#....) need only to be defined when the variable `handwheel_support` has the value 3 in the config.def file.

```
// ----------------------------------------------------------------------
// Dual Check Safety
// ----------------------------------------------------------------------
#if dcs_support == 1
// --------------------------------------
// main contactor (MCC)
// --------------------------------------
   #define  __MCC_STATE              X8.0
   #define  __OUTPUT_MCC             Y8.0
```

The input of the MCC state and the output for the MCC contactor have to be defined if Dual Check Safety support has been included in the PMC.

```
// --------------------------------------
// door status
// --------------------------------------
   #define  __GUARD_STATE            X8.1
   #define  __UNLOCK_DOOR            Y8.1
```

The input for the guard lock switch contact state and the output to control the guard lock have to be defined if Dual Check Safety support has been included in the PMC.

```
// --------------------------------------
// hold to run button
// --------------------------------------
   #define  __HOLD_TO_RUN_BUTTON     X8.2
```

The input for the Hold-To-Run-Button (Dead man's switch) has to be defined if Dual Check Safety support has been included in the PMC.

```
// --------------------------------------
// switch to activate setup mode
// --------------------------------------
   #define  __SETUP_SWITCH           X8.3
```

The input for the setup switch (axes and spindle move enable at open door) has to be defined if Dual Check Safety support has been included in the PMC.

```
// --------------------------------------
// In case standard machine operator's
// panel with 55 keys isn't used, define
// the key to open the protection door
// --------------------------------------
   #if operatorspanel != 2

      #define  __DOOR_KEY            X8.5
      #define  __DOOR_LAMP           Y8.7
   #endif
#endif
```

In case of `operatorspanel` equal `0`, `operatorspanel` equal `1` or `operatorspanel` equal `3`, the input for the DOOR-Button and the output for the lamp which indicates DOOR OPEN status must be defined if Dual Check Safety support has been included in the PMC.

```
// ----------------------------------------------------------------------
// overtravel inputs
// ----------------------------------------------------------------------
```

The code for the overtravel inputs need to be modified at the beginning of each axis specific sub-program **Axis0y_Pxx** (see chapter 7.1.1).

```
// ----------------------------------------------------------------------
// reference inputs
// ----------------------------------------------------------------------
```

The code for the reference input need to be modified at the beginning of each axis specific sub-program **Axis0y_Pxx** (see chapter 7.1.1).

```
// ----------------------------------------------------------------------
// serial spindle safe torque off STO (LEVEL-UP ALPHA iSP-B, BETA iSVSP-B)
// this functionality will be included in case of 3xi-B, 0i-F, PMi-A controls
// and Dual Check Safety active
// ----------------------------------------------------------------------
```

The code for the serial spindle safe off outputs need to be modified at the beginning of each spindle specific sub-program **SpiRun_Pxx** (see chapter 7.1.1).

```
// ----------------------------------------------------------------------
// analog spindle I/O module A (ADA02A)
// ----------------------------------------------------------------------
```

The code for the outputs for the analogue output module need to be modified at the end of each path specific sub-program **SpiRun_Pxx** (see chapter 7.1.1).

```
// ----------------------------------------------------------------------
// coolant
// ----------------------------------------------------------------------
```

The code for the coolant output need to be modified at the end of each path specific coolant sub-program **Coolant_Pxx** (see chapter 7.1.1).

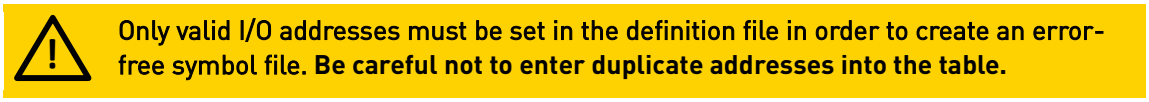
```
// ----------------------------------------------------------------------
// axis brake
// ----------------------------------------------------------------------
```

The code for the brake output need to be modified at the end of each axis specific sub-program **Axis0y_Pxx,** which includes brake support (see chapter 7.1.1).

4.2 dcs_i_o.def

The file dcs_i_o.def in the directory NewMP_BasePMC-x.x.x includes definitions for all I/O addresses related to the Dual Check Safety PMC.

The highlighted values must be adapted for the given machine. The modified definition file must be stored under the same name in ASCII format.




```
// ------------------------------------------------------------------------
// main contactor (MCC)
// ------------------------------------------------------------------------
   #define  __MCC_STATE             X8.0
   #define  __OUTPUT_MCC            Y8.0
```

The input for the MCC contact state and the output to control the MCC have to be defined if Dual Check Safety support has been included in the PMC.

```
// ------------------------------------------------------------------------
// door status
// ------------------------------------------------------------------------
   #define  __GUARD_STATE           X8.1
```

The input for the guard lock switch contact state has to be defined if Dual Check Safety support has been included in the PMC.

```
// ------------------------------------------------------------------------
// hold to run button
// ------------------------------------------------------------------------
   #define  __HOLD_TO_RUN_BUTTON    X8.2
```

The input for the Hold-To-Run-Button (dead man's switch) has to be defined if Dual Check Safety support has been included in the PMC.

```
// ------------------------------------------------------------------------
// switch to activate setup mode
// ------------------------------------------------------------------------
   #define  __SETUP_SWITCH          X8.3
```

The input for the set-up switch (axes and spindle move enable at open door) has to be defined if Dual Check Safety support has been included in the PMC.

```
// ----------------------------------------------------------------------------
// serial spindle safe torque off STO (LEVEL-UP ALPHA iSP-B, BETA iSVSP-B)
// this functionality will be included in case of 3xi-B, 0i-F and PMi-A controls
// ----------------------------------------------------------------------------
```

The code for the inputs for the serial spindle safe torque off signals need to be modified at the beginning of the sub-program **Machine_Inputs** (see chapter 7.2.1). The code for the serial spindle safe off outputs need to be modified at the end of the sub-program **Machine_Outputs** (see chapter 7.2.1).

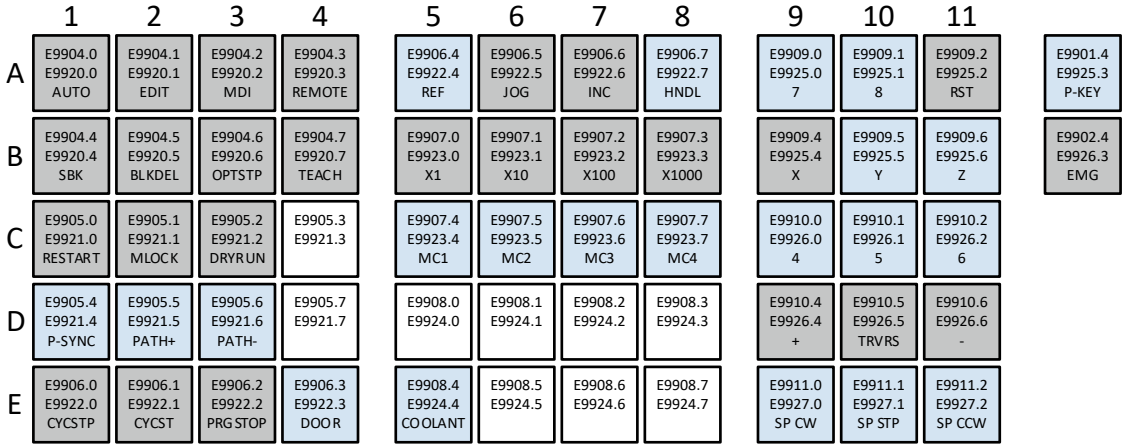
```
// ----------------------------------------------------------------------------
// axis brake
// ----------------------------------------------------------------------------
```

The code for the brake outputs need to be modified at the end of the sub-program **Machine_Outputs** (see chapter 7.2.1).

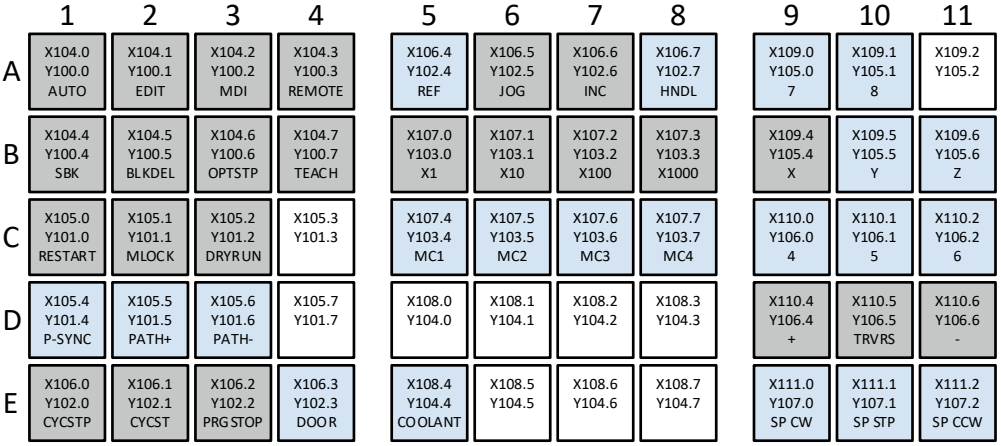
4.3 Standard Machine Operator's Panel with 55 keys

The inputs and outputs for the Standard Machine Operator's Panel with 55 keys are defined in the file op_55keys.def in the directory NewMP_BasePMC-x.x.x\def. The allocation of the inputs/outputs to the individual keys can be seen in the figures below. All greyed keys are programmed as part of the basic PMC. Depending on the configuration, the blued keys may be also part of the basic PMC.

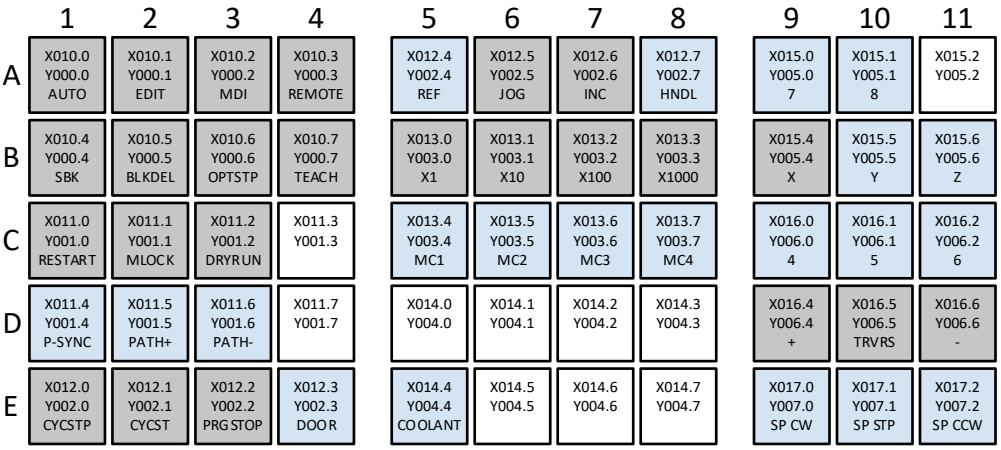
start_address_op 0: (I/O E9900/E9920)



start_address_op 1: (I/O X0100/Y0100)



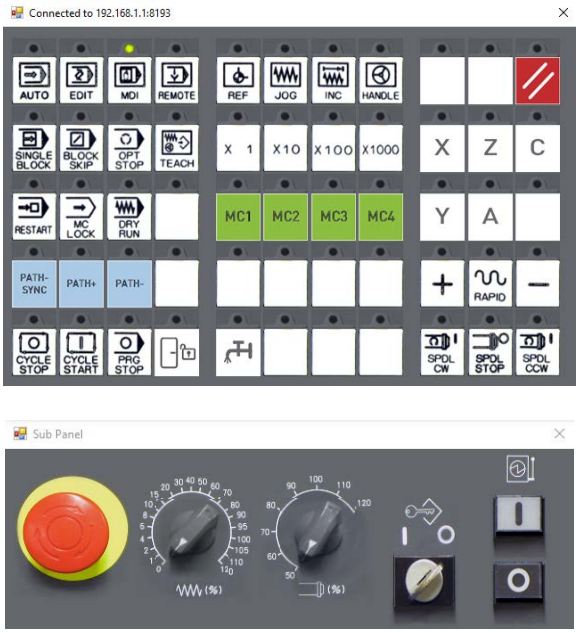
start_address_op 2: (I/O X0006/Y0000)



4.4 Virtual Operator's Panel

The Virtual Operator's Panel is a tool for remote control of the machine for maintenance or start-up purposes. It can be used when the following conditions in the config.def file are fulfilled:

- Standard machine operator's panel with 55 keys selected (`operatorspanel 2`)
- Virtual operator's panel with 55 keys selected (`virtual_op_panel 1`)
- `start_address_op 1` or `2`
- `demo_control 0`



The Virtual Operator's Panel uses the same E-address range than the Standard Machine Operator's Panel with 55 keys (`start_address_op 0`, I/O E9900/E9920)

Activation of the Virtual Operator's Panel:

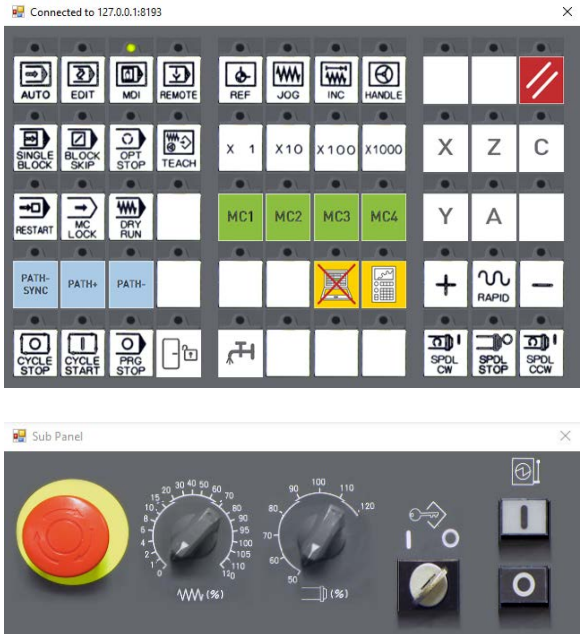
Keys	Description
	Press the Emergency Stop Button on the operator's panel of the machine
+ + -	Press the + and - -keys on the operator's panel of the machine for 3 seconds (LED's of this two buttons are blinking until the panel is switched).While the Virtual Operator's Panel is active the LED of the REMOTE key is blinking.
	Release the Emergency Stop Button

Deactivation of Virtual Operator's Panel:

Key	Description
	Press the Emergency Stop Button on the operator's panel of the machine. The LED of the REMOTE key stops blinking.

When the control is used for demonstration purposes, the switching of the Operator's Panel differs from a real machine. The following conditions in the config.def file must be fulfilled:

- Standard machine operator's panel with 55 keys selected (`operatorspanel 2`)
- Virtual operator's panel with 55 keys selected (`virtual_op_panel 1`)
- `start_address_op 1` or `2`
- `demo_control 1`



Key combination	Function
	This key on the operator's panel of the control disables the remote access of the Virtual Operator's Panel. When the LED of this key is active, remote access is not possible.
	This key is used to switch between the real and the virtual operator's panel.

4.5 Small Machine Operator's Panel with 30 keys

The inputs and outputs for the Small Machine Operator's Panel with 30 keys are defined in the file op_30keys.def in the directory NewMP_BasePMC-x.x.x\def. The allocation of the inputs/outputs to the individual keys can be seen in the figures below. All greyed keys are programmed as part of the basic PMC. Depending on the configuration, the blued keys may be also part of the basic PMC. The functionality of control specific keys is determined by the setting of the variable `control_type` in the config.def file.

start_address_op 0: (I/O E9900/E9920)

	1	2	3	4	5	6	7
A	E9904.0 E9920.0 AUTO	E9904.1 E9920.1 EDIT	E9904.2 E9920.2 MDI	E9904.3 E9920.3 REMOTE	E9904.4 E9920.4 REF	E9904.5 E9920.5 JOG	E9904.6 E9920.6 P-KEY
B	E9905.0 E9921.0 INC	E9905.1 E9921.1 HNDL	E9905.2 E9921.2 TEACH	E9905.3 E9921.3 +Z/+3	E9905.4 E9921.4 +Y/-X	E9905.5 E9921.5 -4/-4	E9905.6 E9921.6 RST
C	E9906.0 E9922.0 SBK	E9906.1 E9922.1 BLKDEL	E9906.2 E9922.2 OPTSTP	E9906.3 E9922.3 -X/+X	E9906.4 E9922.4 TRVRS	E9906.5 E9922.5 -X/+Z	E9906.6 E9922.6 EMG
D	E9907.0 E9923.0 RESTART	E9907.1 E9923.1 MLOCK	E9907.2 E9923.2 DRYRUN	E9907.3 E9923.3 +4/+4	E9907.4 E9923.4 -Y/+X	E9907.5 E9923.5 -Z/-3	
E	E9908.0 E9924.0 CYCSTP	E9908.1 E9924.1 CYCST	E9908.2 E9924.2 PRGSTOP	E9908.3 E9924.3 SP CW	E9908.4 E9924.4 SP STP	E9908.5 E9924.5 SP CCW	

start_address_op 1: (I/O X0100/Y0100)

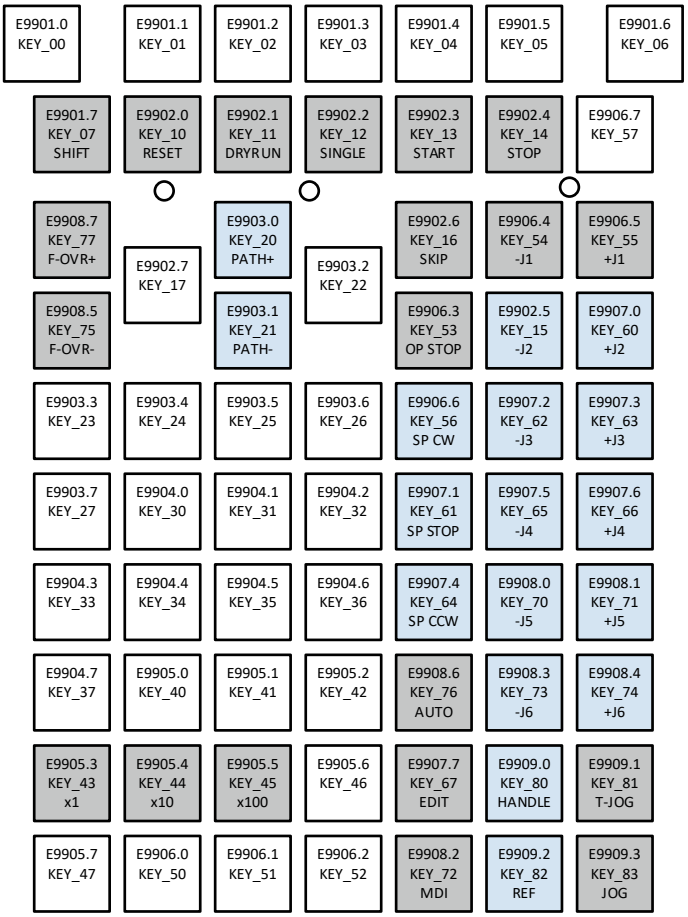
	1	2	3	4	5	6
A	X104.0 Y100.0 AUTO	X104.1 Y100.1 EDIT	X104.2 Y100.2 MDI	X104.3 Y100.3 REMOTE	X104.4 Y100.4 REF	X104.5 Y100.5 JOG
B	X105.0 Y101.0 INC	X105.1 Y101.1 HNDL	X105.2 Y101.2 TEACH	Y105.3 +Z/+3	X105.4 +Y/-X	X105.5 -4/-4
C	X106.0 Y102.0 SBK	X106.1 Y102.1 BLKDEL	X106.2 Y102.2 OPTSTP	X106.3 -X/+X	X106.4 TRVRS	X106.5 -X/+Z
D	X107.0 Y103.0 RESTART	X107.1 Y103.1 MLOCK	X107.2 Y103.2 DRYRUN	X107.3 +4/+4	X107.4 -Y/+X	X107.5 -Z/-3
E	X108.0 Y104.0 CYCSTP	X108.1 Y104.1 CYCST	X108.2 Y104.2 PRGSTOP	X108.3 Y104.3 SP CW	X108.4 Y104.4 SP STP	X108.5 Y104.5 SP CCW

start_address_op 2: (I/O X0006/Y0000)

	1	2	3	4	5	6
A	X010.0 Y000.0 AUTO	X010.1 Y000.1 EDIT	X010.2 Y000.2 MDI	X010.3 Y000.3 REMOTE	X010.4 Y000.4 REF	X010.5 Y000.5 JOG
B	X011.0 Y001.0 INC	X011.1 Y001.1 HNDL	X011.2 Y001.2 TEACH	Y011.3 +Z/+3	X011.4 +Y/-X	X011.5 -4/-4
C	X012.0 Y002.0 SBK	X012.1 Y002.1 BLKDEL	X012.2 Y002.2 OPTSTP	X012.3 -X/+X	X012.4 TRVRS	X012.5 -X/+Z
D	X013.0 Y003.0 RESTART	X013.1 Y003.1 MLOCK	X013.2 Y003.2 DRYRUN	X013.3 +4/+4	X013.4 -Y/+X	X013.5 -Z/-3
E	X014.0 Y004.0 CYCSTP	X014.1 Y004.1 CYCST	X014.2 Y004.2 PRGSTOP	X014.3 Y004.3 SP CW	X014.4 Y004.4 SP STP	X014.5 Y004.5 SP CCW

4.6 *i*Pendant

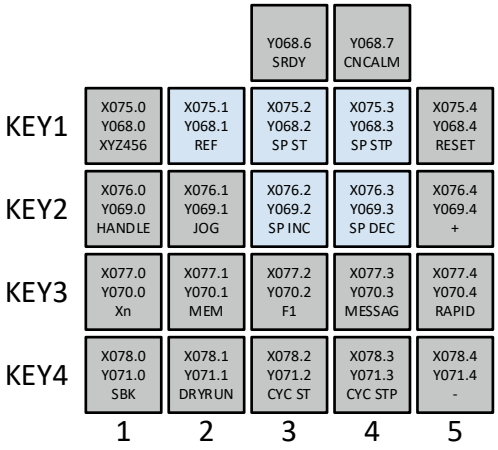
The inputs for the *i*Pendant (A02B-0333-C260/C261/C262/C263) are defined in the file i_pendant.def in the directory NewMP_BasePMC-x.x.x\def. The allocation of the inputs to the individual keys can be seen in the figure below. All greyed keys are programmed as part of the basic PMC. Depending on the configuration, the blued keys may be also part of the basic PMC.



Key combination	Function
E9901.7 KEY_07 SHIFT + E9903.0 KEY_20 PATH+ E9901.7 KEY_07 SHIFT + E9903.1 KEY_21 PATH-	Path synchronization
E9901.7 KEY_07 SHIFT + E9909.3 KEY_83 JOG	Step feed mode (INC)
E9901.7 KEY_07 SHIFT + Exxxx.y KEY_xx +/-Jx	JOG feed at rapid traverse rate
E9901.7 KEY_07 SHIFT + E9908.5 KEY_75 F-OVR-	Feed override 0%
E9901.7 KEY_07 SHIFT + E9908.7 KEY_77 F-OVR+	Feed override 100%

4.7 Handy Machine Operator's Panel (A02B-0259-C221/C241)

The inputs and outputs for the Handy Machine Operator's Panel (A02B-0259-C221/C241) are defined in the file hmop.def in the directory NewMP_BasePMC-x.x.x\def. The allocation of the inputs/outputs to the individual keys can be seen in the figure below. All greyed keys are programmed as part of the basic PMC. Depending on the configuration, the blued keys may be also part of the basic PMC.



Key combination	Function
X077.2 Y070.2 F1 + X076.0 Y069.0 HANDLE	Step feed mode (INC)
X077.2 Y070.2 F1 + X075.0 Y068.0 XYZ456	Decrement selected axis number
X077.2 Y070.2 F1 + X077.0 Y070.0 Xn	Decrement handle/inc feed magnification
X077.2 Y070.2 F1 + X075.2 Y068.2 SP ST	Manual spindle start CCW
X077.2 Y070.2 F1 + X076.3 Y069.3 SP DEC	Spindle override 50%
X077.2 Y070.2 F1 + X076.2 Y069.2 SP INC	Spindle override 100%

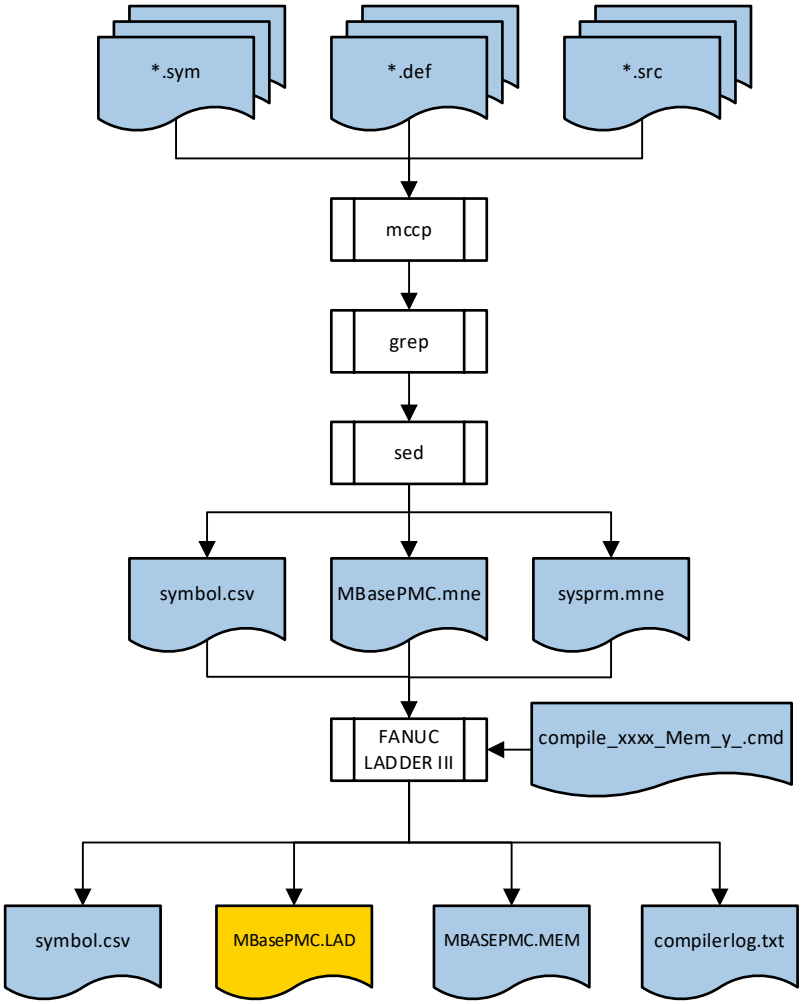
5 Generation of project files

The following batch files in the directory NewMP_BasePMC-x.x.x are at your disposal for the generation of the different FANUC LADDER III project files:

make_mem.cmd	creates the FANUC LADDER III project file MBasePMC.LAD
make_io_mem.cmd	creates the I/O Link *i* project file IoLink_i.FIL
make_msg.cmd	creates the external message file msg_pmc1.mem
make_dcs.cmd	creates the FANUC LADDER III project file MBaseDCS.LAD

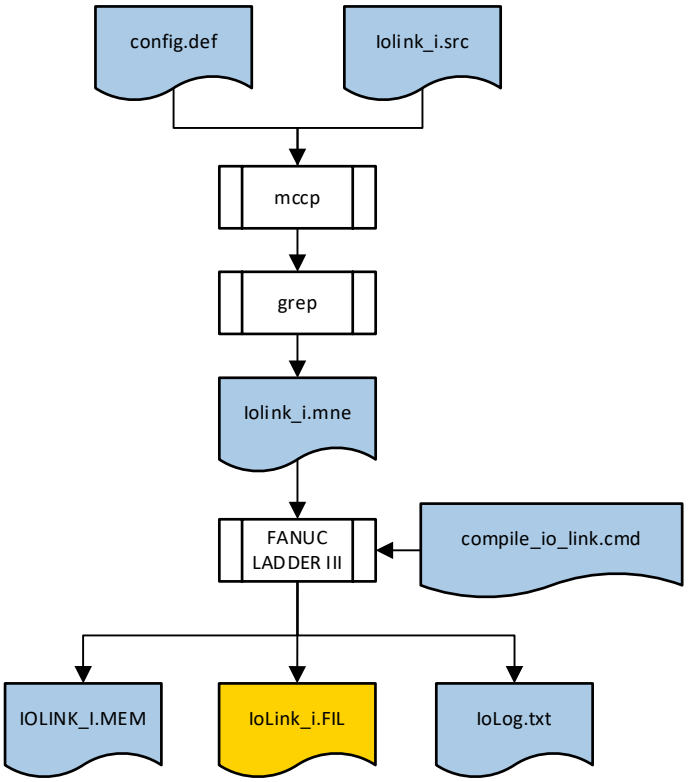
The mcpp preprocessor from Kiyoshi Matsui is used for the generation of the mnemonic files and the symbol file. Some aesthetic formatting is accomplished with utilities from the GNU Win project (grep, sed, date, sleep).

5.1 make_mem.cmd



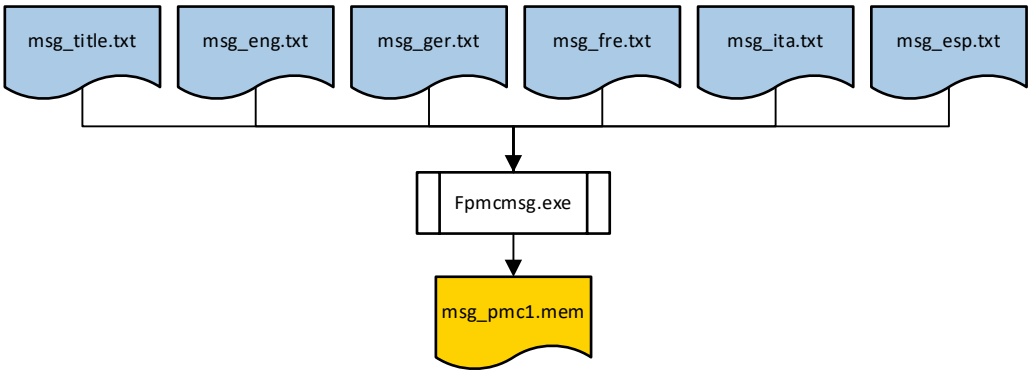
The command file **make_mem.cmd** generates the **MBasePMC.LAD** project file for FANUC LADDER III. **MBasePMC.LAD** contains the ladder code for the first PMC with the functionality, which was defined in **config.def**. The file compilerlog.txt contains the log information of the FANUC LADDER III command line compiler.

5.2 make_io_mem.cmd



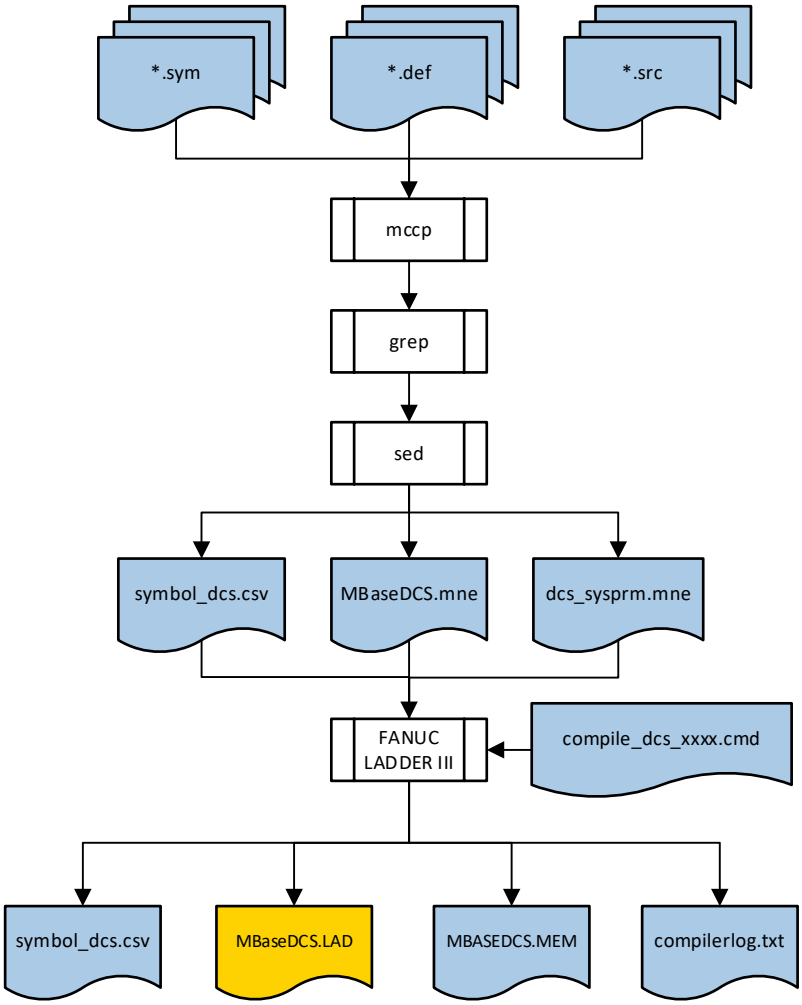
The command file **make_io_mem.cmd** generates the I/O Link *i* project file **IoLink_i.FIL** for FANUC LADDER III. **IoLink_i.FIL** contains the I/O groups, which were defined in **config.def**. The file IoLog.txt contains the log information of the FANUC LADDER III command line compiler.

5.3 make_msg.cmd



The command file **make_msg.cmd** calls the multi-language PMC message creation tool, which creates the file **msg_pmc1.mem**. This file contains all the messages for the first PMC which were defined in the language definition files msg_xxx.txt. The memory card file can be directly stored into the control.

5.4 make_dcs.cmd



The command file **make_dcs.cmd** generates the **MBaseDCS.LAD** project file for FANUC LADDER III. **MBaseDCS.LAD** contains the ladder code with the functionality, which was defined in **config.def**. The file compilerlog.txt contains the log information of the FANUC LADDER III command line compiler.

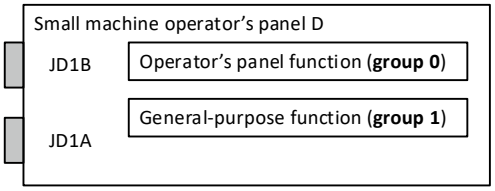
6 Assignment of I/O modules

Because of the possible different I/O modules which can be used, it is necessary to define them in FANUC LADDER III. The I/O modules for the operator's panel and handy machine operator's panel will be declared automatically. Depending on the defined address range (start_address_op) in the config.def file (X0006/Y0000 or X0100/Y0100) the I/O modules will be assigned as follows:

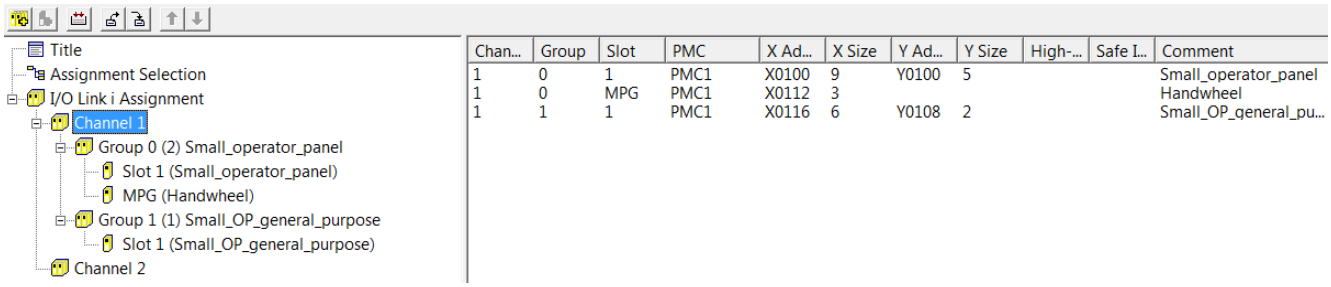
6.1 I/O Link *i*

6.1.1 Small machine operator's panel D (A02B-0338-C151#...)

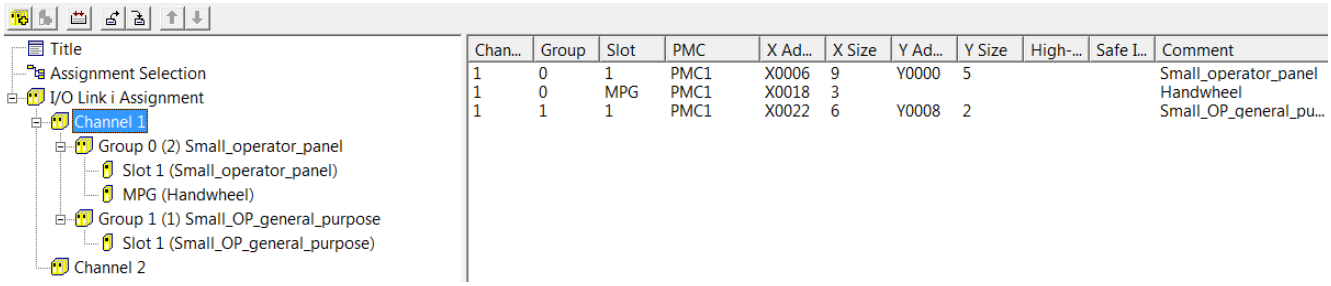
In case of the small operator's panel, two I/O groups will be added.



`start_address_op` equal `1` and `handwheel_support` equal `1`



`start_address_op` equal `2` and `handwheel_support` equal `1`

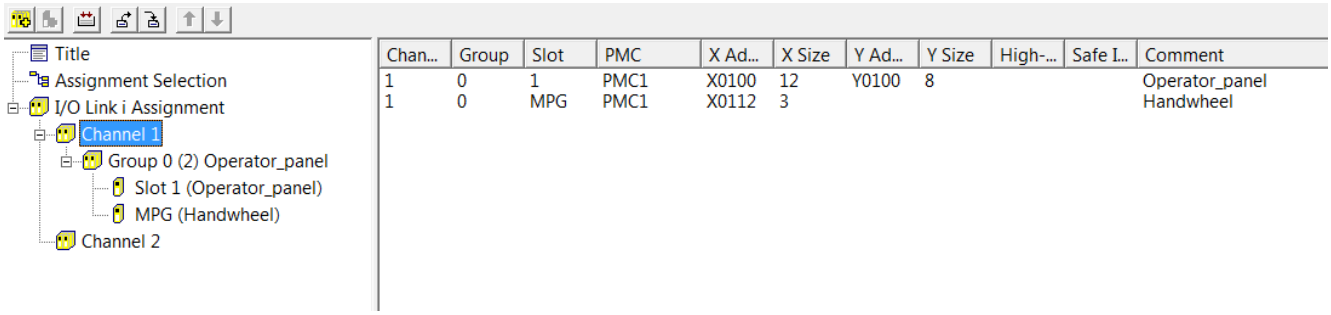


If you add an I/O group, check the basic group count of selectable I/O Link *i* assignment in the Assignment Selection.

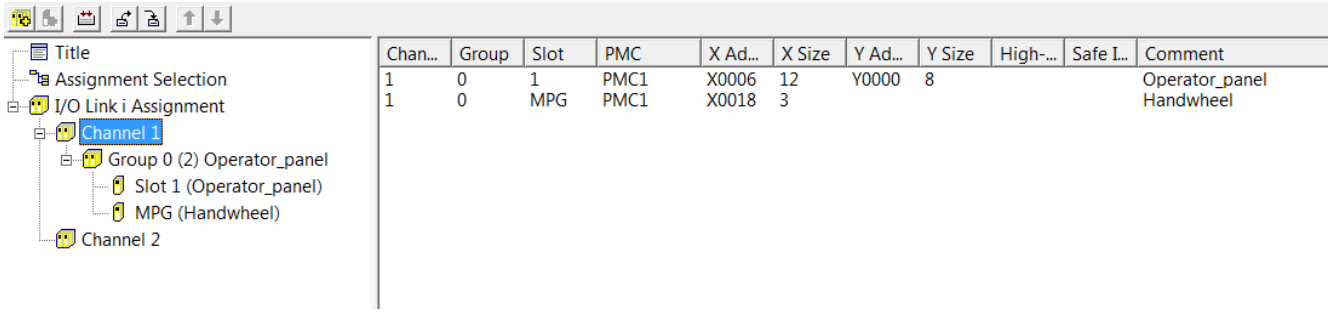
The I/O groups related to Dual Check Safety (safe I/O) are not included in IoLink_i.FIL.

6.1.2 Standard machine operator's panel with 55 keys)

`start_address_op` equal `1`, `handwheel_support` equal `1` and `dcs_support` equal `0`



`start_address_op` equal `2`, `handwheel_support` equal `1` and `dcs_support` equal `0`

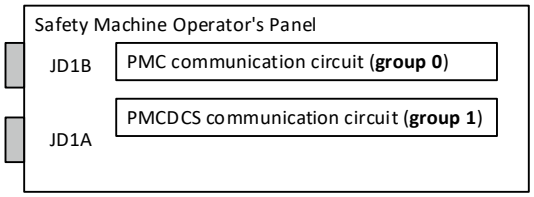


If you add an I/O group, check the basic group count of selectable I/O Link *i* assignment in the Assignment Selection.

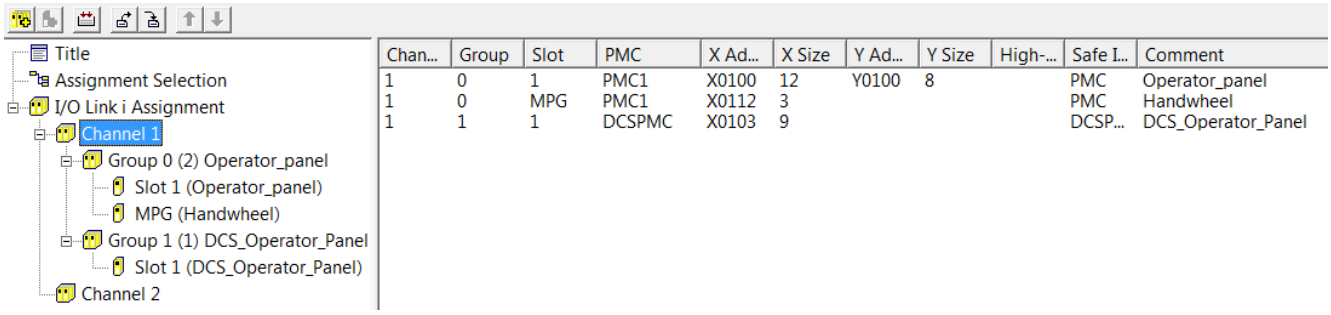
The I/O groups related to Dual Check Safety (safe I/O) are not included in IoLink_i.FIL.

6.1.3 Safety machine operator's panel

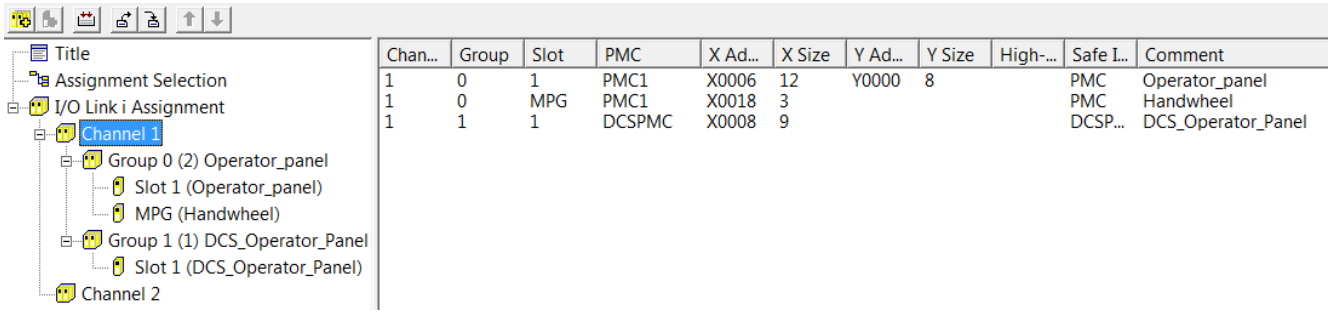
If dual check safety is selected, the 2nd input group for the DCSPMC will also be added to the I/O Link *i* configuration if `operatorspanel` equal 2.



`start_address_op` equal 1, `handwheel_support` equal 1, `dcs_support` equal 1



`start_address_op` equal 2, `handwheel_support` equal 1, `dcs_support` equal 1

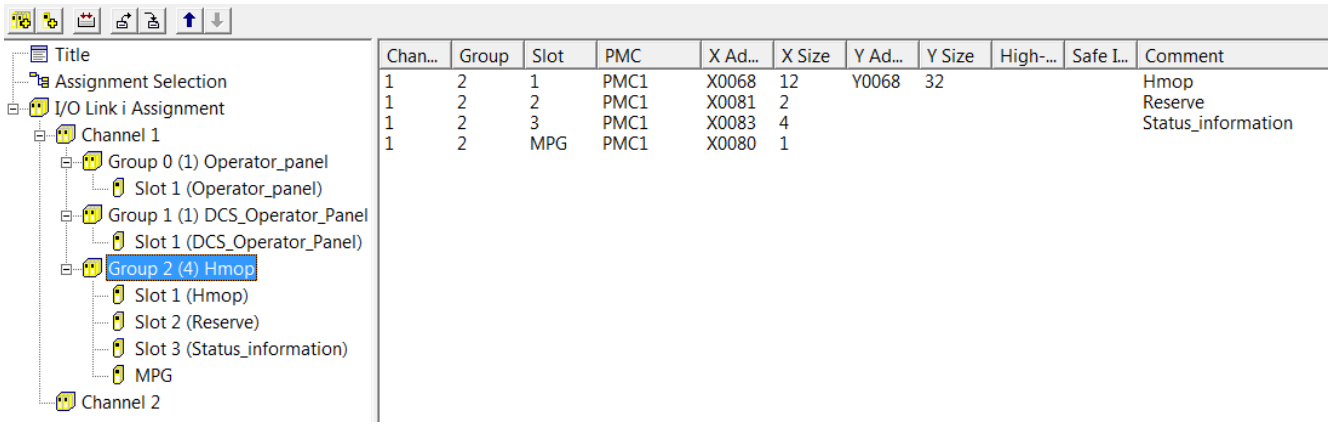


If you add an I/O group, check the basic group count of selectable I/O Link *i* assignment in the Assignment Selection.

The additional I/O groups related to Dual Check Safety (safe I/O) are not included in IoLink_i.FIL.

6.1.4 Handy Machine Operator's Panel (channel 1)

`handwheel_support 2`



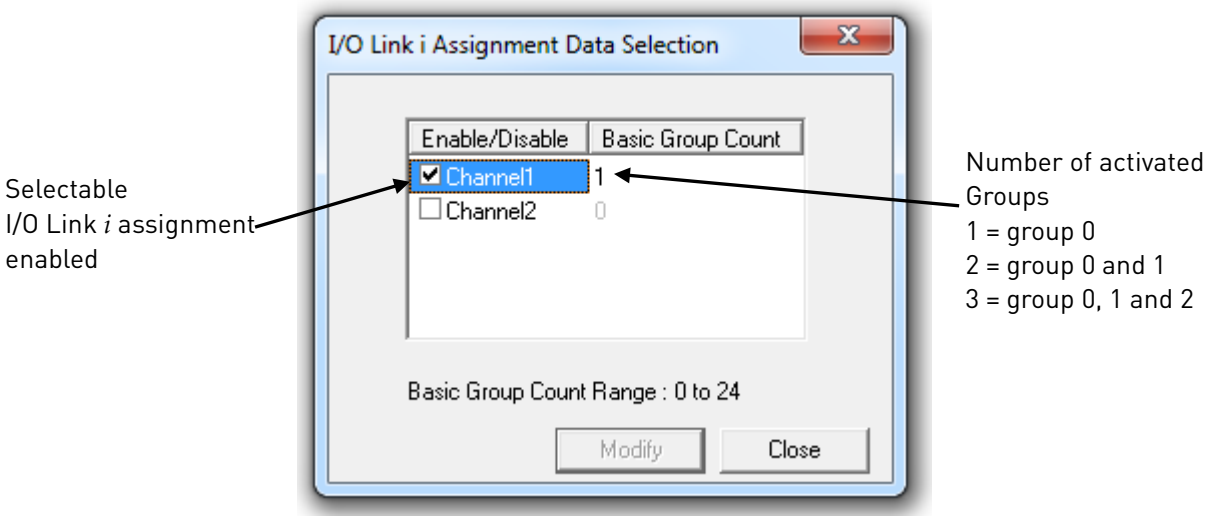
If you add an I/O group, check the basic group count of selectable I/O Link *i* assignment in the Assignment Selection.

The I/O groups related to Dual Check Safety (safe I/O) are not included in IoLink_i.FIL.

6.1.5 Selectable I/O Link *i* assignment

This function enables the common use of a sequence program for several machines, which have different I/O device configurations. If you have chosen `start_address_op` equal `1` (X0100 / Y0100) or `2` (X0006 / Y0000) or hand-`wheel_support` equal `2`, the selectable I/O Link *i* assignment function will be automatically activated. Additional I/O groups must be added in FANUC LADDER III or in the system parameters of the control (see FANUC LADDER III Operator's Manual B-66234EN or PMC Programming manual B-64393EN, B-63983EN or B-64513EN).

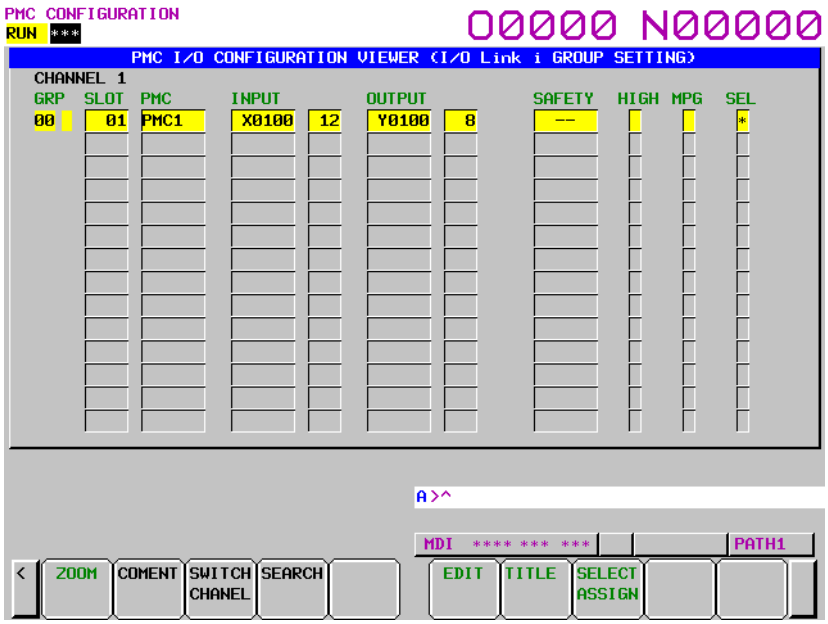
Assignment selection in FANUC LADDER III



You have only to specify the number of groups (Basic Group Count) which are common to all machines starting from group 0. It is impossible to skip groups which were activated by this function.

Setting of I/O groups in the I/O CONFIGRATION VIEWER (I/O Link *i* SELECTABLE I/O) screen of the control

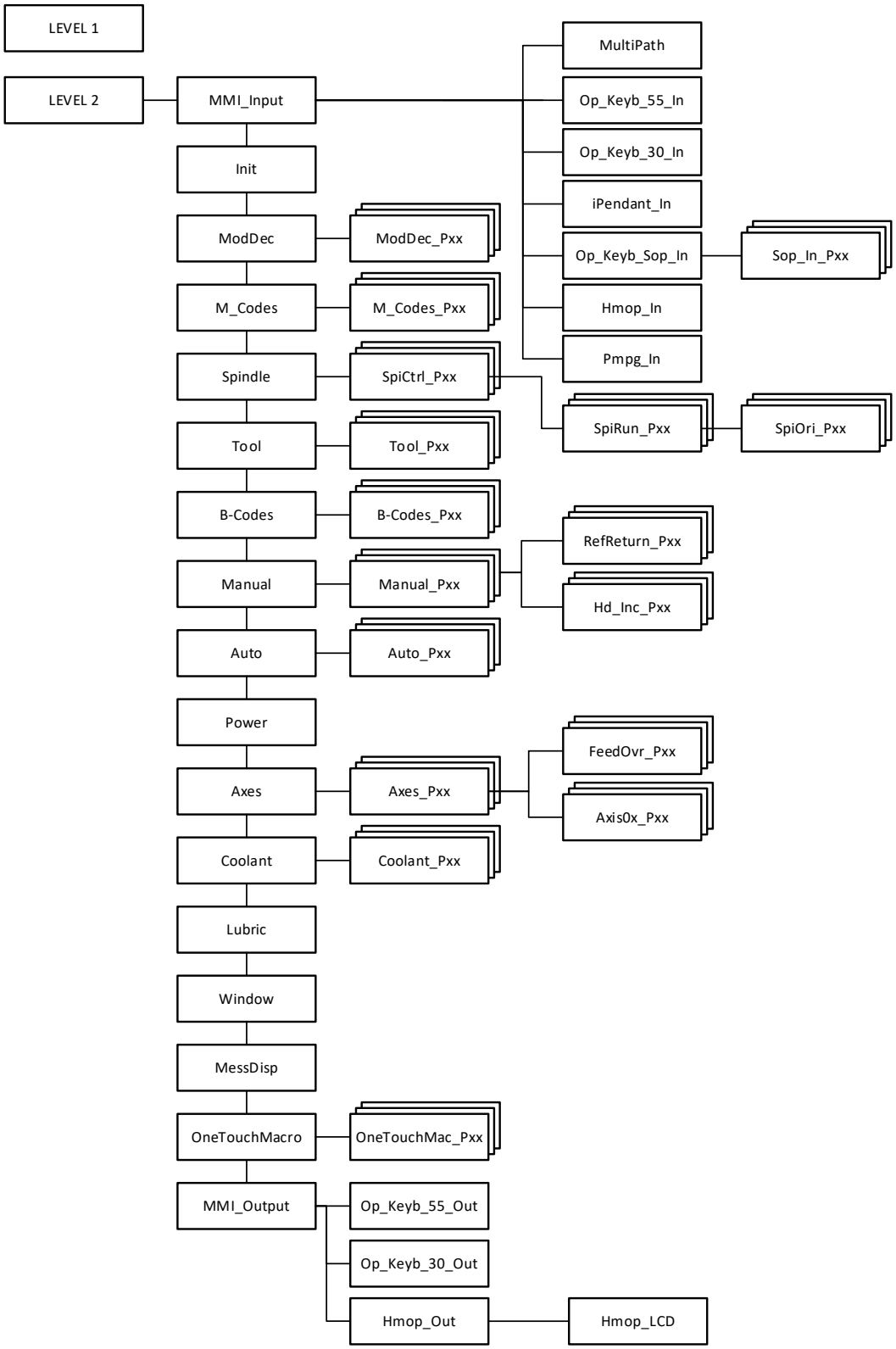
For further information, please read the related section in the PMC programming manual.



7 Machine specific modifications

Because of different hardware solutions for the machines, it is necessary to modify the PMC program.

7.1 Structure of MBasePMC.LAD



LEVEL 1	The module includes the management of the emergency stop
LEVEL 2	Sub-program calls
MMI_Input	Selection of the active operator's panel and path specific address mapping
MultiPath	Path selection
Op_Keyb_55_In	Mapping of the inputs of the standard machine operator's panel with 55 keys to internal relays
Op_Keyb_30_In	Mapping of the inputs of the small machine operator's panel with 30 keys to internal relays
iPendant_In	Mapping of the inputs of the iPendant to internal relays
Op_Keyb_Sop_In	Mapping of the inputs of the software operator's panel to internal relays
Sop_In_Pxx	Mapping of the path specific inputs of the software operator's panel to internal relays
Hmop_In	Mapping of the inputs of the handy machine operator's panel to internal relays
Pmpg_In	Mapping of the inputs of the Portable manual pulse generator to internal relays
Init	Initialization of parameters and data at the start-up of the control
ModDec	Decoding of the machine operation modes
ModeDec_Pxx	Path specific decoding of machine operation modes
M_Codes	Decoding and completion of M codes
M_Codes_Pxx	Path specific decoding and completion of M codes
Spindle	Management module for the spindle (S code, spindle start / stop)
SpiCtrl_Pxx	Path specific management module for the spindle (S code, spindle start/stop)
SpiRun_Pxx	Path specific functions for driving and monitoring the spindle
SpiOri_Pxx	Path specific spindle orientation
Tool	Tool management
Tool_Pxx	Path specific tool management
B_Codes	Management of 2nd auxiliary function (B code)
B-Codes_Pxx	Path specific management of 2nd auxiliary function (B code)
Manual	Functions related to manual operation
Manual_Pxx	Path specific functions related to manual operation
RefReturn_Pxx	Path specific management of reference position return sequence
Hd_Inc_Pxx	Path specific management of the hand-wheel and incremental feed
Auto	Functions related to automatic operation
Auto_Pxx	Path specific functions related to automatic operation
Power	Management of power supply to axes and spindle amplifiers, set-up mode. The functions in this module must be adapted to the machine specific circumstances
Axes	Management of axes common signals, call of axis specific modules
Axes_Pxx	Path specific management of axes common signals, call of axis specific modules
FeedOvr_Pxx	Path specific management of the feed override
AxisOy_Pxx	Axis specific functions like over-travel, reference return, feed axis direction selection, servo off etc. related to the corresponding path.
Coolant	Coolant control
Coolant_Pxx	Path specific coolant control
Lubric	Management of the central lubrication
Window	Management of the "low speed" WINDOW READ/WRITE functions
MessDisp	Management and display of the machine specific messages
OneTouchMacro	Management of one touch macro calls
OneTouchMac_Pxx	Path specific one touch macro call
MMI_Output	Call of the operator's panel related output modules
Op_Keyb_55_Out	Mapping of the internal relays to the corresponding outputs of the standard machine operator's panel with 55 keys
Op_Keyb_30_Out	Mapping of the internal relays to the corresponding outputs of the small machine operator's panel with 30 keys
Hmop_Out	Mapping of the internal relays to the corresponding outputs of the handy machine operator's panel.
Hmop_LCD	Management of the LCD display of the handy machine operator's panel.

The used symbols can be found in the file **symbol.csv**.

The file **memory_management.xlsx** in the directory DOC includes the memory management of MBasePMC.LAD.

7.1.1 Modification of inputs and outputs

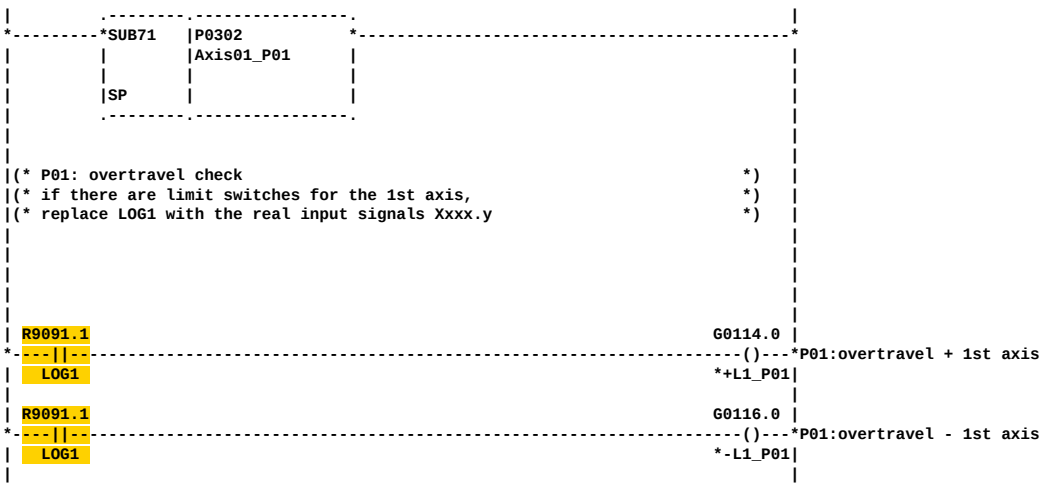
The following sub-programs may contain code, which has to be replaced by real input/output signals:

- **Axis0y_Pxx** (overtravel inputs, reference input, brake output)
- **SpiRun_Pxx** (safe torque off in case of serial spindle, outputs in case of 12-BIT ANALOG OUTPUT MODULE)
- **Coolant_Pxx** (output for path specific coolant system)

If required, the highlighted values must be replaced with the real input or output signals.

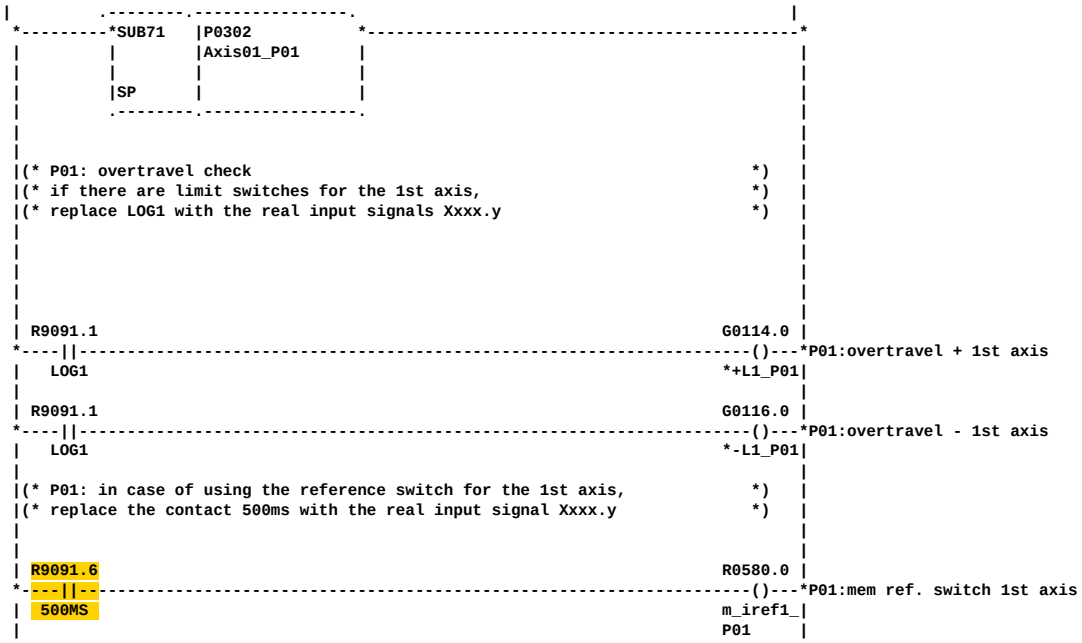
Axis overtravel inputs (sub-program Axis0y_Pxx)

Example overtravel check 1st axis, path01:



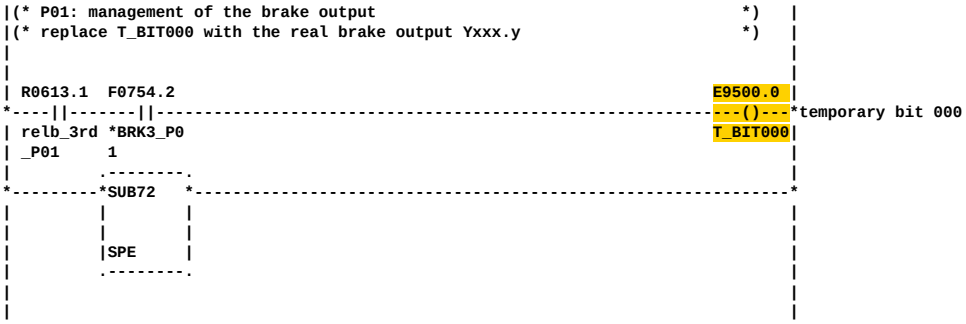
Axis reference input (sub-program Axis0y_Pxx)

Example reference input 1st axis, path01:



Axis brake output (sub-program Axis0y_Pxx)

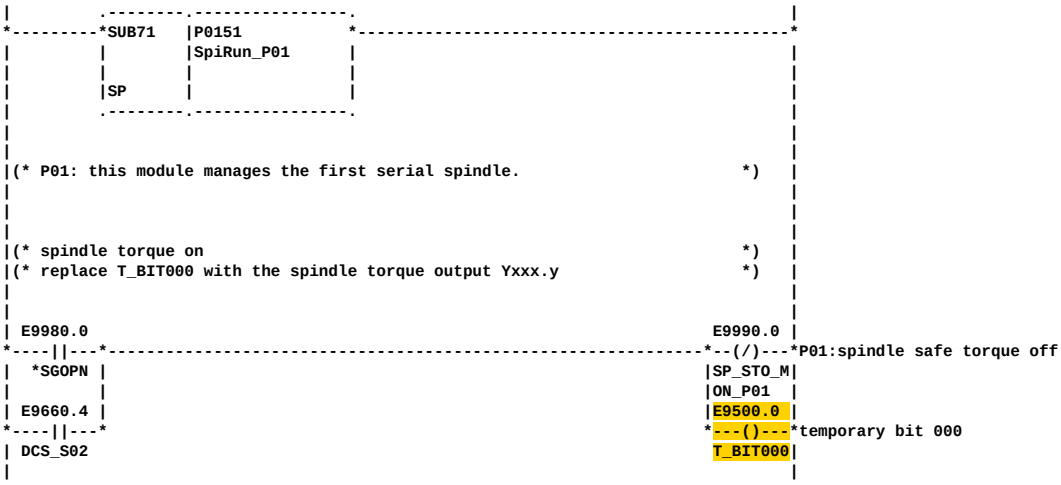
Example brake output 3rd axis, path01:



Spindle safe torque off in case of serial spindle and DCS (sub-program SpiRun_Pxx)

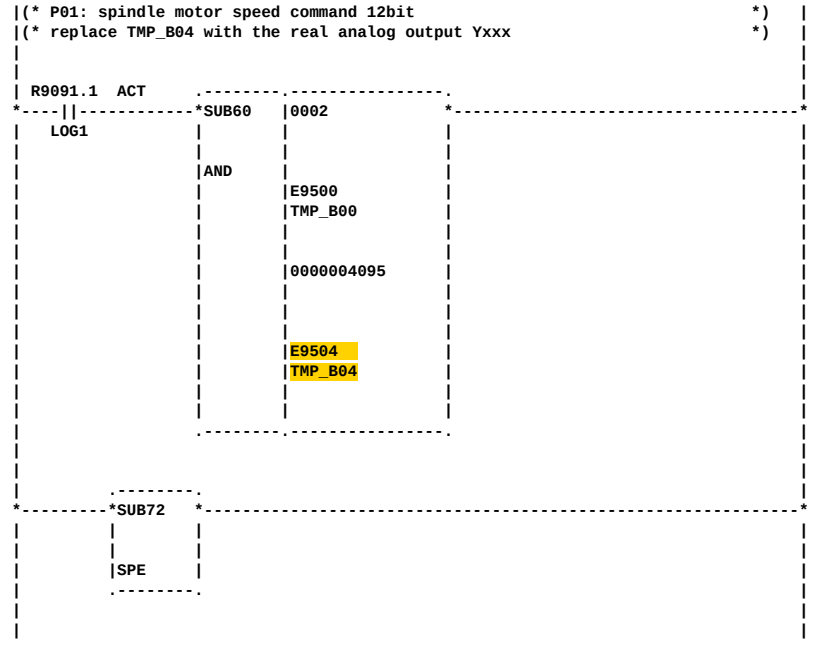
This functionality will be included in case of 3x*i*-B, 0*i*-F and PM*i*-A controls when spindle_support_Pxx = 1 and dcs_support = 1

Example safe torque off serial spindle path01:



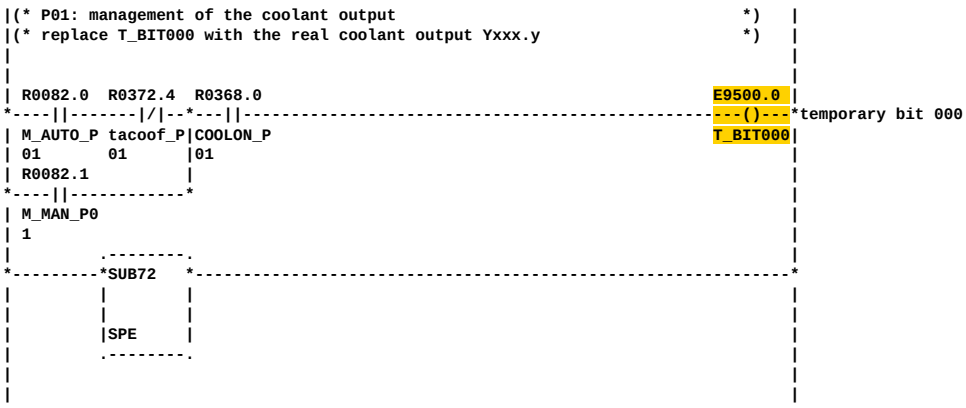
Spindle analogue outputs (sub-program SpiRun_Pxx)

Example outputs analogue spindle path01:

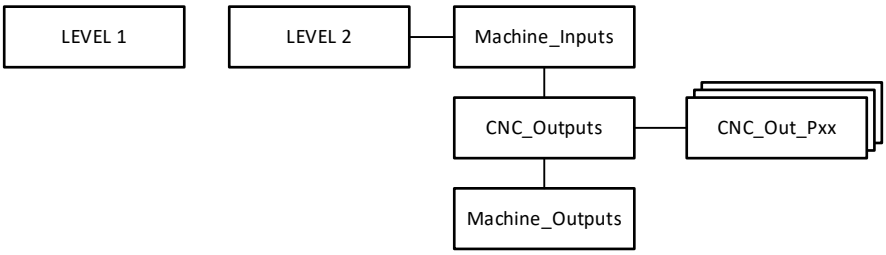


Coolant output (sub-program Coolant_Pxx)

Example coolant output path01:



7.2 Structure of BaseDCS.LAD



LEVEL 1	The module includes the management of the emergency stop
LEVEL 2	Sub-program calls
Machine_Inputs	Management of physical inputs and safe inputs
CNC_Outputs	Management of output signals to the CNC
Machine_Outputs	Management of physical outputs and safe outputs

The used symbols can be found in the file **symbol_dcs.csv**.

The file **dcs_memory_management.xlsx** in the directory DOC includes the memory management of MBaseDCS.LAD.

7.2.1 Modification of inputs and outputs

The following sub-programs may contain code, which has to be replaced by real input/output signals:

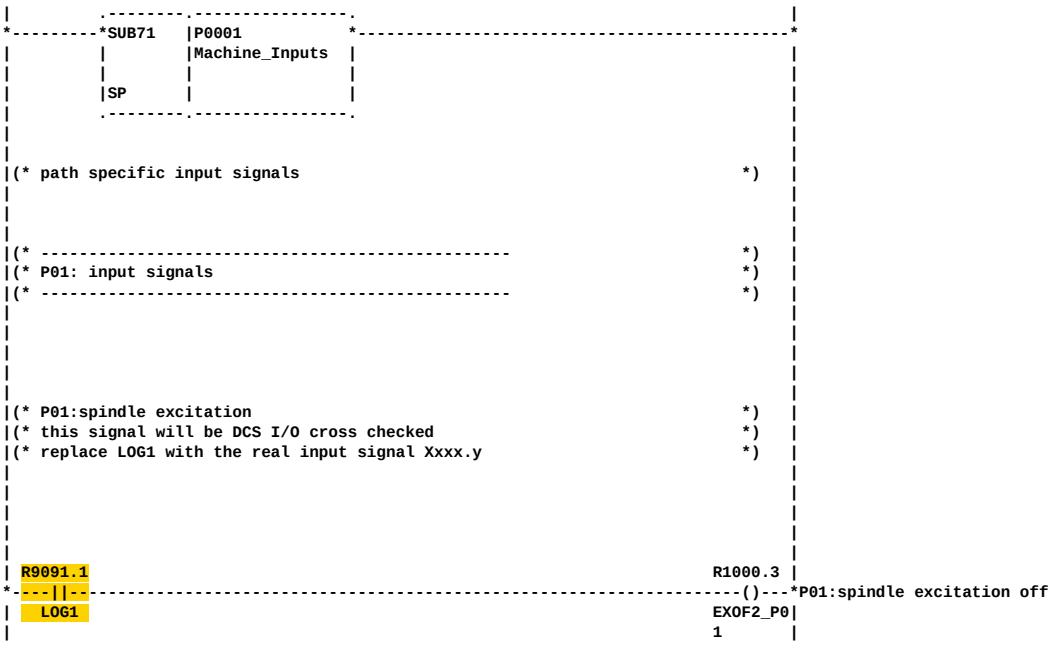
- **Machine_Inputs** (spindle excitation off or spindle safe torque off in case of serial spindles)
- **Machine_Outputs** (spindle safe torque off in case of serial spindles and brake outputs in case of axes)

If required, the highlighted values must be replaced with the real input or output signals.

Spindle excitation off inputs (sub-program Machine_Inputs)

This functionality will be included in case of 0*i*-D and 3x*i*-A controls when spindle_support_Pxx = 1 and dcs_support = 1

Example spindle excitation off input serial spindle path01:



Depending on the spindle configuration, the sub-program may contain multiple spindle excitation off inputs.

Spindle safe torque off input (sub-program Machine_Inputs)

This functionality will be included in case of 3x*i*-B, 0*i*-F and PM*i*-A controls when spindle_support_Pxx = 1 and dcs_support = 1

Example safe torque off input serial spindle path01:

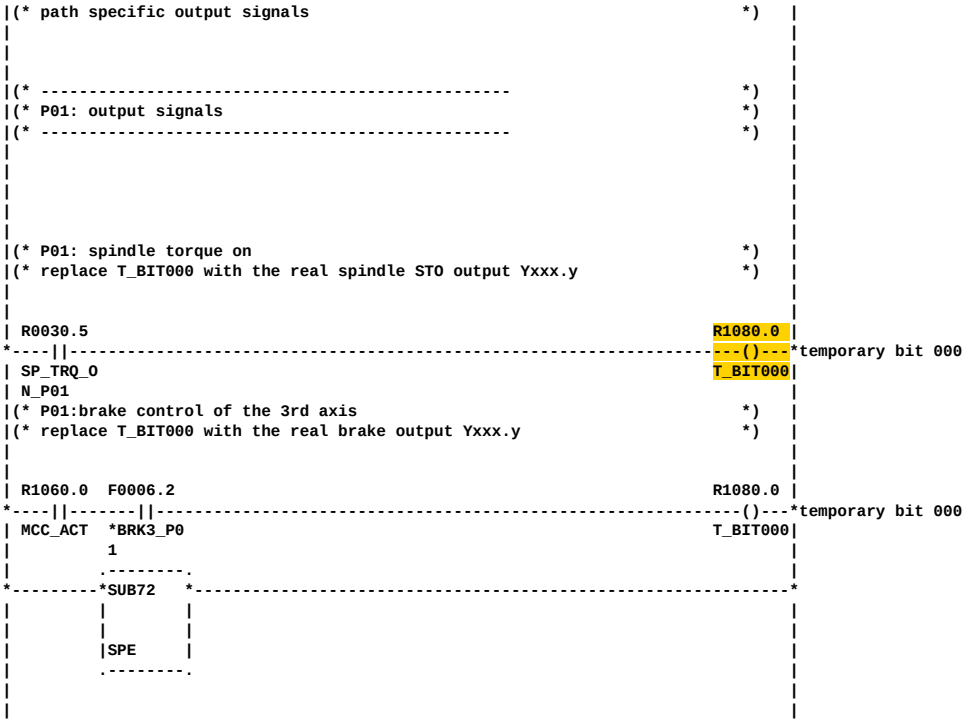
```
|         .--------.----------------.                                          |
*---------*SUB71   |P0001           *------------------------------------------*
|         |        |Machine_Inputs  |                                          |
|         |        |                |                                          |
|         |SP      |                |                                          |
|         .--------.----------------.                                          |
|                                                                              |
|                                                                              |
|(* path specific input signals                                        *)      |
|                                                                              |
|                                                                              |
|                                                                              |
|(* ------------------------------------------------                   *)      |
|(* P01: input signals                                                 *)      |
|(* ------------------------------------------------                   *)      |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|(* P01:input spindle safe torque off                                  *)      |
|(* replace LOG1 with the real input signal Xxxx.y                     *)      |
|                                                                              |
|                                                                              |
| R9091.1                                                          R0020.0 |
*---||--------------------------------------------------------------()---*P01:spindle STO state
|   LOG1                                                           SP_STO_P|
|                                                                  01      |
```

Depending on the spindle configuration, the sub-program may contain multiple spindle safe torque off inputs.

Spindle safe torque off output (sub-program Machine_Outputs)

This functionality will be included in case of 3x*i*-B, 0*i*-F and PM*i*-A controls when spindle_support_Pxx = 1 and dcs_support = 1

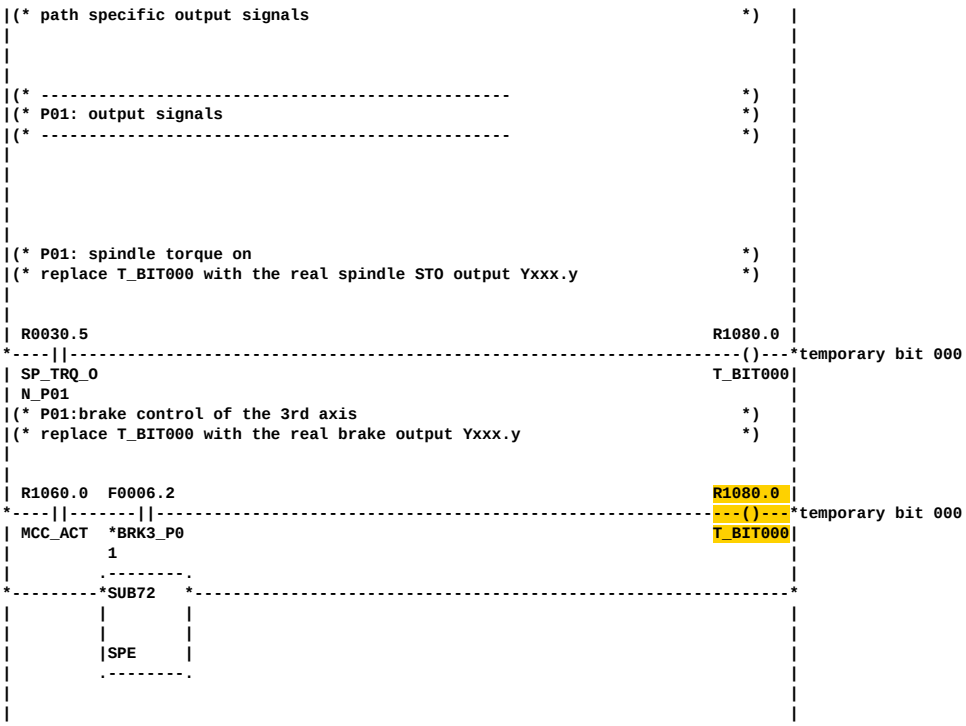
Example safe torque off output serial spindle path01:



Depending on the spindle configuration, the sub-program may contain multiple spindle safe torque off outputs.

Brake outputs (sub-program Machine_Outputs)

Example brake output 3rd axis path01:



Depending on the axes configuration, the sub-program may contain multiple brake outputs.

8 Message management

Either one language, five languages or external message files are available for message display (language_support). In case of 30*i* / 31*i* and 32*i* controls, one of the following options is required:

-J911 External message
-J913 External data input

In case of 35*i*-B and PM*i*-A, basic option 2 is required.

Useful additional options for 3x*i*-B, 0*i*-F and PM*i*-A controls:

-R856#512K PMC symbol, comment, and message capacity expansion (512KB)
-R856#1M PMC symbol, comment, and message capacity expansion (1MB)
-R856#2M PMC symbol, comment, and message capacity expansion (2MB)

Each selection puts 64 CNC alarm messages (EX1xxx), 96 user alarm messages (20xx) and 240 operator messages (2100 – 2339) at disposal.

For the messages, the following format is used:

CNC alarm message AL1+000= to AL1+063

Example: **AL1+000=EMERGENCY STOP ACTIVE**

User alarm message OP1+000= to OP1+95

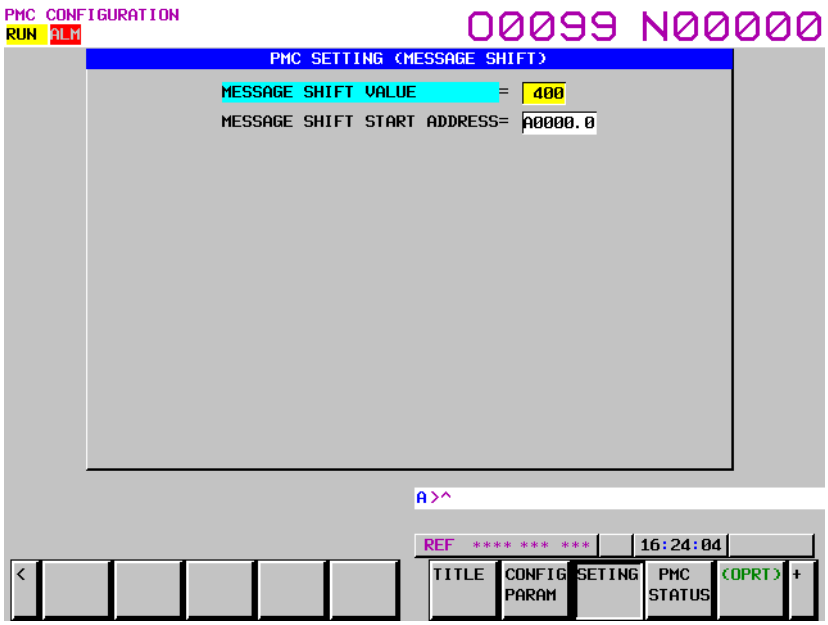
Example: **OP1+001=ZERO RETURN REQUIRED**

Operator message OP1+100= to OP1+339

Example: **OP1+100=CHECK CENTRAL LUBRICATION**

8.1 Managing multiple languages

With the five language selection (`language_support` equal `1`), the five languages can be selected using the "message shift function".



MESSAGE SHIFT VALUE	0	English
	400	German
	800	French
	1200	Italian
	1600	Spanish

The shift value can be input in the configuration screen or by Keep Relay K918-K919.

8.2 External message files

The root directory of the library includes the definition files for the external PMC messages.

msg_title.txt	includes the title of the external message file
msg_eng.txt	English messages
msg_ger.txt	German messages
msg_fre.txt	French messages
msg_ita.txt	Italian messages
msg_esp.txt	Spanish messages

The batch file **make_msg.cmd** generates the memory card format file **msg_pmc1.mem** which can be stored into the control.

Example:

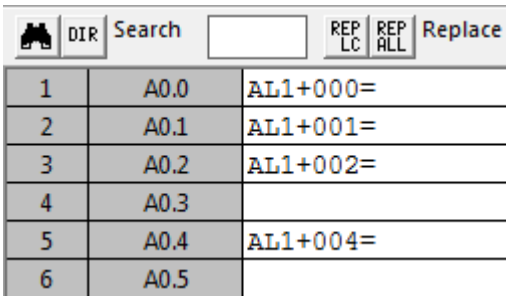
```
%@4-D

"error messages AL1+000= to AL1+063="
"--------------------------------------------------------------------------------"
$0    %A000.0     AL1+000=EMERGENCY STOP ACTIVE
$0    %A000.1     AL1+001=SPINDLE ALARM
$0    %A000.2     AL1+002=ERROR CENTRAL LUBRICATION
$0    %A000.4     AL1+004=TIMEOUT IPENDANT

"alarm messages OP1+000= to OP1+0095="
"--------------------------------------------------------------------------------"
$0    %A008.1     OP1+001=ZERO RETURN REQUIRED
$0    %A008.2     OP1+002=PROG. SPINDLE SPEED TOO HIGH
$0    %A008.3     OP1+003=M-CODE M[I230,D76] OUT OF RANGE

"operator messages OP1+100= to OP1+339="
"--------------------------------------------------------------------------------"
$0    %A020.0     OP1+100=CHECK CENTRAL LUBRICATION
$0    %A020.1     OP1+101=COOLANT SWITCHED OFF
$0    %A020.2     OP1+102=BATTERIE ALARM
$0    %A020.3     OP1+103=PROGRAM START DISABLED
$0    %A020.4     OP1+104=BRAKE TEST REQUIRED
%
```

In case of using external message files, the message number must be specified in FANUC LADDER III otherwise the messages will not be displayed.



9 CNC parameter setting

Depending on the type of parameter, it has to be modified at multiple places. The **KEY** column shows, if the parameter exists once or several times in the parameter file of the control.

KEY	QUANTITY	DESCRIPTION
C	1	Common parameter
P	1 -> n	Path related parameter
A	1 -> n	Axis related parameter
S	1 -> n	Spindle related parameter

9.1 Type of PMC memory

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
C	N11940				PMC memory type (1st PMC) 0 or 2: Memory B 3: Memory C 4: Memory D 5: Memory E (Multi Axes Control System)

9.2 Communication method of I/O Link

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
C	N11933#0	C1T	1		Specifies the communication method of channel 1. 0: I/O Link is used. 1: I/O Link *i* is used.
C	N11933#1	C2T	1		Specifies the communication method of channel 2. 0: I/O Link is used. 1: I/O Link *i* is used.

9.3 M codes

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
P	N03001#7	HMI	1		High-speed M, S, T, B interface
P	N03010		4/8/16	[ms]	Time lag in strobe signals MF, SF, TF, and BF
P	N03011		4/8/16	[ms]	Acceptable width of M, S, T, and B function completion signal (FIN)
P	N03404#7	M3B			Number of M codes which can be specified per block 1: 3 M codes
P	N11630#5	M5B			Number of M codes which can be specified per block 1: 5 M codes

Values of N03010 and N03011 depend on the PMC execution time set in parameter N11930.

9.4 Each axis workpiece coordinate system preset

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
C	N03006#6	WPS	1		Each axis workpiece coordinate system preset enabled
P	N03104#3	PPD	1		Relative position display is preset
A	N10410#1	EAX	1		Even while the axis is moved by PMC axis control, each axis workpiece preset signal function is effective
P	N11275		100		First M code to turn each axis workpiece coordinate system preset
P	N11276		9		Number of M codes used for each axis workpiece coordinate system preset
P	N11277#0	WPA	1		No alarm in case of auxiliary function lock

9.5 Spindle (common to all spindles)

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
P	N03705#4	EVS	1		Output of S code and SF signal (T type)
P	N03705#6	SFA	1		SF signal is always output (M type)
P	N03708#0	SAR	1		The spindle speed arrival signal (SAR) will be checked
P	N03740			[ms]	Delay before checking the SAR signal (delay in PMC 200ms)

9.6 Serial spindle

In case of 3x*i*-A/B and PM*i*-A controls, the following option is required:

-J850 Spindle serial output

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
P	N03706#6	CWM			Voltage polarity when the spindle speed voltage is output
P	N03706#7	TCW			Voltage polarity when the spindle speed voltage is output
S	N03716#0	A/S	1		Spindle motor type is serial spindle
S	N03717		1		Spindle amplifier number
S	N03718		1		Subscript for display
P	N03736				Maximum clamp speed of spindle motor (M-type)
S	N03741			[min^{-1}]	Maximum speed for gear 1
S	N03772			[min^{-1}]	Maximum spindle speed
C	N03786#4	ISS	0		The resolution enabled for the spindle speed command is: 0: Maximum spindle speed/4095[min-1] 1: Maximum spindle speed/16383[min-1]
S	N03791#0	SSEs	0		The resolution enabled for the spindle speed command: 0: depend to the bit 4(SSI) of parameter No.3798 and bit 4 (ISS) of parameter No.3786. 1: is a maximum spindle speed/1048575 [min-1].
C	N03798#4	SSI	0		The resolution enabled for the spindle speed command: 0: depend to the bit 4 (ISS) of parameter No.3786. 1: Maximum spindle speed/16383 [min-1].
S	N04030			[ms]	Soft start/stop time
S	N04399#2	SOSALW	1		Acc./dec. by soft start/stop is executed when the motor rotation is reset
C	N08133#5	SSN	0		Serial spindle (only 0i-D/F)

9.7 Analogue spindle (spindle_support 2)

In case of 30*i* / 31*i* and 32*i* controls, the following option is required:

-J860 Spindle Analog Output Function

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
P	N03706#6	CWM			Voltage polarity when the spindle speed voltage is output
P	N03706#7	TCW			Voltage polarity when the spindle speed voltage is output
S	N03716#0	A/S	1		Spindle motor type is serial spindle
S	N03717		1		Spindle amplifier number
S	N03718		1		Subscript for display
S	N03730		1000		Data used for adjusting the gain of the analogue output of spindle speed
S	N03731		0		Compensation value for the offset voltage of spindle speed analogue output
P	N03736				Maximum clamp speed of spindle motor (M-type)
S	N03741			[min^{-1}]	Maximum speed for gear 1
C	N03786#4	ISS	0		The resolution enabled for the spindle speed command is: 0: Maximum spindle speed/4095[min-1] 1: Maximum spindle speed/16383[min-1]
C	N03798#4	SSI	0		The resolution enabled for the spindle speed command: 0: depend to the bit 4 (ISS) of parameter No.3786. 1: Maximum spindle speed/16383 [min-1].
S	N03799#1	NDPs	1		Position coder disconnection alarm is not checked
C	N08133#5	SSN	1		Serial spindle is not used (only 0*i*-D/F)

9.8 PMC axis used as a spindle

In case of 30*i* / 31*i* and 32*i* controls, the following option is required:

-J804 Axis control by PMC

In case of 35*i*-B and PM*i*-A, basic option 1 is required.

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
A	N01006#0	ROTx	1		Rotary axis (A type)
A	N01006#1	ROSx	0		
A	N01008#0	ROA	1		Rotary axis roll over is valid
A	N01008#2	RLLx	1		Relative coordinates are rounded
A	N01260		360.0	[°]	Shift amount per rotation
P	N03105#1	PCF	1		Movement of PMC controlled axis is not added to the actual feedrate display
C	N03137#7	EAC	1		PMC axis status display screen is displayed (not available for 0*i*-D)
P	N03411		5		M code preventing buffering
P	N08001#2	OVE	1		Use of PMC axis specific signals like override...
A	N08003#3	FEX	0		Max. feedrate extended (**in case of Is-C and rpm > 546 1/min set this bit to 1**)
P	N08004#5	DSL	1		No display of alarm 139
P	N08005#4	EVP	1		„Position loop control" in case of „Speed Command" 10h
P	N08005#5	IFV	1		Override for each group in PMC axis control
P	N08007#2	VCP	1		Speed command in PMC axis control is FS16 type
A	N08010				Selection of DI/DO group for each axis controlled by PMC: path01=1, path02=5, path03=9, path04=13, path05=17, path06=21, path07=25, path08=29, path09=33, path10=37, path11=22, path12=26, path13=30, path14=34, path15=38
A	N08028			[ms]	Linear acc./dec. for speed commands for PMC axis control
A	N08040			Input increment	The amount of travel per one revolution of the motor at least command (must be an integer value, otherwise reference position is lost)

9.9 Spindle control with servo motor

The following options are required:

-J978 Spindle Control with Servo Motor (Live Tool Control)
-J850 Spindle serial output

Minimal parameter setting for spindle control with servo motor:

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
A	N01006#0	ROTx	1		Rotary axis (A type)
A	N01006#1	ROSx	0		
A	N01008#0	ROA	1		Rotary axis roll over is valid
A	N01008#2	RLLx	1		Relative coordinates are rounded
A	N01260		360.0	[°]	Shift amount per rotation
A	N01408#3	IRCx	1		maximum cutting feedrate is multiplied by 10
A	N02422#0	SVSAR2	1		Detection level of speed arrival coefficient 2
A	N02455			[pulse]	Number of pulses per detector rotation: Integer part(α) [16384 semi-closed]
A	N02456			[pulse]	Number of pulses per detector rotation: Index part(ß) [7 semi-closed]]
A	N02482		50	[%]	Detection level of speed arrival (SVSAR)
A	N02483		5	[min^{-1}]	Detection level of speed zero (SVSST)
P	N03411		5		M code preventing buffering
P	N03702#1	EMS	1		The multi-spindle control function is not used 0: Used. 1: Not used.
P	N03705#4	EVS	1		Output of S code and SF signal (T type)
P	N03705#6	SFA	1		SF signal is always output (M type)
S	N03717		0		Spindle amplifier number
S	N03718		1		Subscript for display
S	N03741			[min^{-1}]	Maximum speed for gear 1
S	N03772			[min^{-1}]	Maximum spindle speed
A	N11000#4	PCAx			For parameter PCE(No.11006#0)=0, to each axis, positional control under spindle control with servo motor is: 0: Disabled. 1: Enabled.
A	N11000#5	SOAx			The servo axis used by the spindle control with servo motor is: 0: used as a control axis (move command is enabled). 1: used as a servo axes for spindle use (move command is disabled). This parameter is only effective on FANUC Series 32*i*-B.
A	N11000#7	SRV	1		Spindle control with servo motor is performed (set bit of related axis)
A	N11001#1	TCR	1		Parameter N11016 is used for acc. / dec.
A	N11010		1		Used spindle number
A	N11011			[°]	Movement of spindle per servo motor revolution
A	N11012			[min^{-1}]	Spindle indexing speed
A	N11013			[detection unit]	Position deviation limit in movement
A	N11014			[detection unit]	Position deviation limit in the stopped state
A	N11015			[min^{-1}]	Maximum motor speed
A	N11016			[ms]	Time constant of acceleration/deceleration in SV speed control mode for each axis
A	N11020			[min^{-1}]	Acc. / dec. switching speed (S0)
A	N11030			[min^{-1}/s]	Individual acceleration / deceleration 1 (Aa)

0*i*-D:

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
A	N01024		8		90C8 / 90E8 servo software is used
P	N03706#4	GTT	1		Spindle gear selection is Type T (0*i*-MD)
S	N03716#0	A/S	1		Spindle motor type is serial spindle
C	N08133#5	SSN	0		Serial spindle used

Required software: System software D6F1/25.0 (0*i*-TD), D4F1/25.0 (0*i*-MD) or later
Servo software 90C8 / 4.0, 90E8 / 4.0 or later

0*i*-F:

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
P	N03706#4	GTT	1		Spindle gear selection is Type T (0*i*-MF)
S	N03716#0	A/S	1		Spindle motor type is serial spindle
C	N08133#5	SSN	0		Serial spindle used

9.10 Spindle orientation (serial spindle)

In case of 3x*i*-A/B and PM*i*-A controls, the following option is required:

-J853 Spindle Orientation for one Spindle

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
S	N03729#0	ORTs			Spindle orientation (External stop position) 0: Not performed (one position) 1: performed (external stop position)
S	N04031				Stop position (if N03729#0 = 0)
S	N04077				Spindle orientation stop position shift (if N03729#0 = 1)

9.11 Reference position return

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
A	N01005#1	DLZx			Reference position return sequence 0: with dog 1: without dog
A	N01006#5	ZMI			Direction of reference position return 0: positive direction (dog at negative axis limit) 1: negative direction (dog at positive axis limit)
A	N01425			[mm/min.]	FL rate of the reference position return for each axis
A	N01428			[mm/min.]	Reference position return feed rate for each axis
A	N01836			[detection unit]	Servo error amount where reference return is possible (in case of demo_control equal 1 and no servo motors mounted set value **-128** for the axes which should make the reference position return) When 0 is set, 128 is assumed as the default.
P	N03003#5	DEC	0		Deceleration signal for reference position return (signal = 0)
C	N03006#0	GDC	1		Gn196.x is used as deceleration signal for reference position return

9.12 Message management

In case of 30*i*/31*i*/32*i* controls, one of the following options is required:

-J911 External message or
-J913 External data input

In case of 35*i*-B and PM*i*-A, basic option 2 (-R703) is required.

Useful additional options for 3x*i*-B, 0*i*-F and PM*i*-A controls:

-R856#512K PMC symbol, comment, and message capacity expansion (512KB)
-R856#1M PMC symbol, comment, and message capacity expansion (1MB)
-R856#2M PMC symbol, comment, and message capacity expansion (2MB)

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
C	N11931#1	M16			Number of displayed alarm and operator messages 0: 4 1: 16

9.13 *i*Pendant

In case of *i*Pendant **without** manual pulse generator set the following parameters:

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
C	N03206#7	NS2			CNC screen dual display function is: 0: disabled 1: enabled
C	N11539#4	OEN	0		*i*Pendant operation ON/OFF switch <ENBC> is: 0: disabled 1: enabled
C	N11539#6	MCD	0		*i*Pendant operation operation mode control signal <MOPEC> is: 0: disabled 1: enabled
C	N11540		0		Address type of the operation mode control signal
C	N11541		0		Address number of the operation mode control signal
C	N11542		3		Address type of the key signal area (E area)
C	N11543		9900		Address number of the key signal area
C	N11544		64		Refresh interval time of key signal data
C	N11546		0		Deactivate parameters N12300 / N12340 for the 1st manual pulse generator

In case of *i*Pendant **with** manual pulse generator (hand wheel) set the following parameters:

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
C	N03206#7	NS2			CNC screen dual display function is: 0: disabled 1: enabled
C	N11539#4	OEN	1		*i*Pendant operation ON/OFF switch <ENBC> is: 0: disabled 1: enabled
C	N11539#6	MCD	1		*i*Pendant operation operation mode control signal <MOPEC> is: 0: disabled 1: enabled
C	N11540		3		Address type of the operation mode control signal
C	N11541		9928		Address number of the operation mode control signal
C	N11542		3		Address type of the key signal area (E area)
C	N11543		9900		Address number of the key signal area
C	N11544		64		Refresh interval time of key signal data
C	N11546		1		Deactivate parameters N12300 / N12340 for the 1st manual pulse generator

In case of 30*i* / 31*i* and 32*i* controls, the following option is required for the hand wheel:

-J835 Manual Handle Feed 1 Unit

In case of 35*i*-B and PM*i*-A control, at least basic option 1 (-R702) is required for the hand wheel.

9.14 Dual Check Safety

For the Dual Check Safety related parameters please check the manual **DCS_user_manual_en.pdf**.

9.15 Software operator's panel

In case of 30*i* / 31*i* and 32*i* controls, the following options are required:

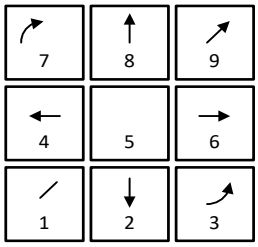
-J960 Software operator's panel
-J961 Software operator's panel general purpose switch

In case of 35*i*-B and PM*i*-A controls, Basic option 2 (-R703) is required.

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
P	N07200#0	OP1	1		Mode selection
P	N07200#1	OP2	1		JOG feed axis selection and JOG rapid traverse buttons
P	N07200#2	OP3	1		Hand wheel
P	N07200#3	OP4	1		JOG and rapid traverse override
P	N07200#4	OP5	1		Block skip, single block, machine lock, dry run
P	N07200#5	OP6	1		Protection key
P	N07200#6	OP7	1		Feed hold

Axes direction keys:

Arrow keys on the MDI panel



Parameter value	Axis direction
0	No movement
1	First axis positive direction
2	First axis negative direction
3	Second axis positive direction
4	Second axis negative direction
5	Third axis positive direction
6	Third axis negative direction
7	Fourth axis positive direction
8	Fourth axis negative direction

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION	
P	N07210		3		MDI key 8:	(Second axis positive direction)
P	N07211		4		MDI key 2:	(Second axis negative direction)
P	N07212		1		MDI key 6:	(First axis positive direction)
P	N07213		2		MDI key 4:	(First axis negative direction)
P	N07214		5		MDI key 1:	(Third axis positive direction)
P	N07215		6		MDI key 9:	(Third axis negative direction)
P	N07216		7		MDI key 3:	(Fourth axis positive direction)
P	N07217		8		MDI key 7:	(Fourth axis negative direction)

Used general purpose switches:

KEY	PARAMETER	VALUE	CHARACTER	KEY	PARAMETER	VALUE	CHARACTER	KEY	PARAMETER	VALUE	CHARACTER
P	N07220	69	**E**	**P**	N07228	82	**R**	**P**	N07236	78	**N**
P	N07221	77	**M**	**P**	N07229	69	**E**	**P**	N07237	67	**C**
P	N07222	71	**G**	**P**	N07230	83	**S**	**P**	N07238	32	**SPACE**
P	N07223	32	**SPACE**	**P**	N07231	69	**E**	**P**	N07239	83	**S**
P	N07224	83	**S**	**P**	N07232	84	**T**	**P**	N07240	84	**T**
P	N07225	84	**T**	**P**	N07233	0		**P**	N07241	65	**A**
P	N07226	79	**O**	**P**	N07234	0		**P**	N07242	82	**R**
P	N07227	80	**P**	**P**	N07235	0		**P**	N07243	84	**T**

KEY	PARAMETER	VALUE	CHARACTER	KEY	PARAMETER	VALUE	CHARACTER	KEY	PARAMETER	VALUE	CHARACTER
P	N07244	68	**D**	**P**	N07252	79	**O**	**P**	N07260	67	**C**
P	N07245	78	**N**	**P**	N07253	80	**P**	**P**	N07261	79	**O**
P	N07246	67	**C**	**P**	N07254	84	**T**	**P**	N07262	79	**O**
P	N07247	0		**P**	N07255	32	**SPACE**	**P**	N07263	76	**L**
P	N07248	0		**P**	N07256	83	**S**	**P**	N07264	65	**A**
P	N07249	0		**P**	N07257	84	**T**	**P**	N07265	78	**N**
P	N07250	0		**P**	N07258	79	**O**	**P**	N07266	84	**T**
P	N07251	0		**P**	N07259	80	**P**	**P**	N07267	0	

KEY	PARAMETER	VALUE	CHARACTER	KEY	PARAMETER	VALUE	CHARACTER	KEY	PARAMETER	VALUE	CHARACTER
P	N07268	83	**S**	**P**	N07276	83	**S**				
P	N07269	80	**P**	**P**	N07277	80	**P**				
P	N07270	73	**I**	**P**	N07278	73	**I**				
P	N07271	78	**N**	**P**	N07279	78	**N**				
P	N07272	68	**D**	**P**	N07280	32	**SPACE**				
P	N07273	76	**L**	**P**	N07281	67	**C**				
P	N07274	69	**E**	**P**	N07282	67	**C**				
P	N07275	0		**P**	N07283	87	**W**				

9.16 Incremental feed and hand wheel

In case of 30*i* / 31*i* and 32*i* controls, the following option is required for the hand wheel:

-J835 Manual Handle Feed 1 Unit

In case of 35*i*-B and PM*i*-A control, basic option 1 (-R702) is required for the hand wheel.

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
P	N07100#0	JHD	1		Activation of the incremental feed
C	N07105#1	HDX	1		Hand wheel I/O Link address manual setting
P	N07113		100		Hand wheel factor m
P	N07114		1000		Hand wheel factor n
C	N08131#0	HPG	1		Manual handle feed is used (only 0*i*-D/F type)
C	N12300				X address of 1st MPG **18** (`start_address_op` equal 2) **80** (`hand wheel_support` equal 2) **112** (`start_address_op` equal 1)
C	N12301		-1		X address of 2nd MPG
C	N12302		-1		X address of 3rd MPG
C	N12303		-1		X address of 4th MPG
C	N12304		-1		X address of 5th MPG

9.17 One Touch Macro Call

The following option is required:

-S655 One touch macro call

KEY	PARAMETER	CODE	VALUE	UNIT	DESCRIPTION
P	N06095		4		Number of programs used by the one-touch macro call function
C	N06096				Number of the first program in the program group used by the one touch macro call function

Not available for 35*i*-B and PM*i*-A.

10 PMC parameter setting

10.1 Keep Relay

Activate / deactivate reference position return

NO.	ADDRESS	DATA	UNIT	DESCRIPTION
	Kn00.0			No reference position return for the 1st axis
	Kn00.1			No reference position return for the 2nd axis
	Kn00.2			No reference position return for the 3rd axis
	Kn00.3			No reference position return for the 4th axis
	Kn00.4			No reference position return for the 5th axis
	Kn00.5			No reference position return for the 6th axis
	Kn00.6			No reference position return for the 7th axis
	Kn00.7			No reference position return for the 8th axis

n = 0 to 14 (path number -1)

10.2 Timer

If the central lubrication (`lubrication_support` equal `1`) was defined in the config.def file, the used variable timers (TMR) must be specified.
The following conditions must be fulfilled: value T18 < value T16 < value T14 < value T12

NO.	ADDRESS	DATA	UNIT	DESCRIPTION
07	T12			Stand-by central lubrication
08	T14			Time between two lubrication impulses
09	T16			Time out central lubrication
10	T18			Lubrication time

For the analogue spindle (`spindle_support` 2 or 3) the acceleration and deceleration time must be specified.

NO.	ADDRESS	DATA	UNIT	DESCRIPTION
n*10+1	n*10+10			Acceleration time analogue spindle
n*10+2	n*10+12			Deceleration time analogue spindle

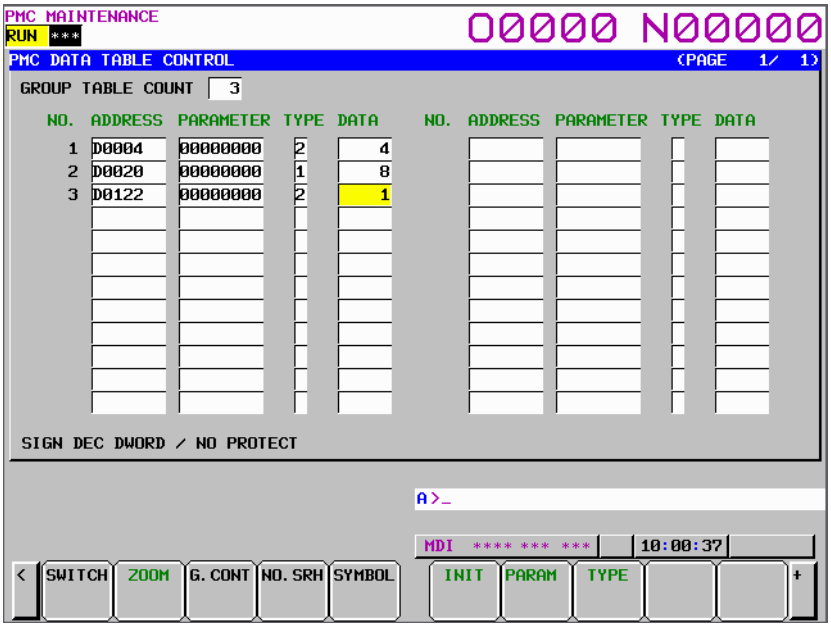
n = 1 to 15 (path number)

Examples	path01	Timer 11	T20	Path01: Acceleration time analogue spindle
		Timer 12	T22	Path01: Deceleration time analogue spindle
	path04	Timer 41	T50	Path04: Acceleration time analogue spindle
		Timer 42	T52	Path04: Deceleration time analogue spindle
	path15	Timer 151	T160	Path15: Acceleration time analogue spindle
		Timer 152	T162	Path15: Deceleration time analogue spindle

10.3 Data

Control table:

Example: Setting of control table for path01



10.3.1 Spindle data

NO.	ADDRESS	DATA	UNIT	DESCRIPTION
0000	Dn004		[min^{-1}]	Maximum spindle speed
0001	Dn008		[min^{-1}]	Maximum motor speed (PMC axis used as a spindle)
0002	Dn012			Gear ratio motor (PMC axis used as a spindle)
0003	Dn016			Gear ratio spindle (PMC axis used as a spindle)

n = 0 to 14 (path number -1)

Example:
Maximum spindle speed = 400 rpm → Dn004 = 400
Maximum motor speed = 4000 rpm → Dn008 = 4000
Motor turns 10 times when the spindle makes 1 turn → Dn012 = 10, Dn016 = 1

NO.	ADDRESS	DATA	UNIT	DESCRIPTION
0000	Dn122		[min^{-1}]	Maximum spindle set-up speed (for Dual Check Safety)

n = 0 to 14 (path number -1)

10.3.2 Reference position return

Length of the software deceleration dog (input units 4/8/16ms) or time for one motor revolution. The value depends on the reference position return feed rate defined in parameter N01428 or N01423 in case of reference position return without dog.

NO.	ADDRESS	DATA	UNIT	DESCRIPTION
0000	Dn020			Length of the software deceleration dog or time for the 1st axis
0001	Dn022			Length of the software deceleration dog or time for the 2nd axis
0002	Dn024			Length of the software deceleration dog or time for the 3rd axis
0003	Dn026			Length of the software deceleration dog or time for the 4th axis
0004	Dn028			Length of the software deceleration dog or time for the 5th axis
0005	Dn030			Length of the software deceleration dog or time for the 6th axis
0006	Dn032			Length of the software deceleration dog or time for the 7th axis
0007	Dn034			Length of the software deceleration dog or time for the 8th axis

n = 0 to 14 (path number -1)

Examples:

Linear axis: Value in parameter N01428 = 2000, travel distance/motor revolution 4mm, PMC cycle time 8ms, reference position return with dog

$$counter\ value = \frac{travel\ distance\ per\ motor\ revolution\ [mm] * 60000}{N01428\left[\frac{mm}{min}\right] * PMC\ cycle\ time[ms] * 2}$$

$$counter\ value = \frac{4\ [mm] * 60000}{2000\left[\frac{mm}{min}\right] * 8[ms] * 2} = 7.5$$

Set a value of about **10** into the data table

Rotary table: Value in parameter N01423 = 3600, gear ratio 1/1, reference position return without dog

$$Time\ for\ one\ motor\ revolution = \frac{360[°] * 60000}{N01423[\frac{°}{min}]}$$

$$Time\ for\ one\ motor\ revolution = \frac{360[°] * 60000}{3600[\frac{°}{min}]} = 6000[ms]$$

Set a value of about **7000** into the data table

11 Appendix

11.1 License of mcpp

```
/*-
 * Copyright (c) 1998, 2002-2008 Kiyoshi Matsui <kmatsui@t3.rim.or.jp>
 * All rights reserved.
 *
 * This software including the files in this directory is provided under
 * the following license.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
```

11.2 GNU GENERAL PUBLIC LICENSE

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
```

your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's

System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

```
  You may convey verbatim copies of the Program's
 source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified
    it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is
    released under this License and any conditions added under section
    7.  This requirement modifies the requirement in section 4 to
    "keep intact all notices".

    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

    a) Convey the object code in, or embodied in, a physical product
    (including a physical distribution medium), accompanied by the
    Corresponding Source fixed on a durable physical medium
```

customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must

suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or

```
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
    authors of the material; or

    e) Declining to grant rights under trademark law for use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys
 the material (or modified versions of
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
```

this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and

propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

  If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

  A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey

the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar
in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF

DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.

For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License. But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.