ADV400 Integration Manual
Version 2.00 Rev 1.17

TABLE OF CONTENTS

1) Introduction 2

2) Passwords 2

3) Integration Menu 2

4) Machine settings 4
   a) Setting page 1: general axis setup 1 4
   b) Setting page 2: general axis setup 2 6
   c) Setting page 3: axis homing setup 7
   d) Setting page 4: miscellaneous 8

5) Axis Tuning 10

6) Custom PLCs 11
   a) PLCs management 11
   b) Writing a PLC 12
   c) Specific PLC25 14
   d) Messages 15
   e) Table of Symbols 17

7) Custom M-codes programming 17

8) Tool Changer 18
   a) Sub-routine 998 18
   b) Tool changer initialisation 20

Annexe 1 : Table of Symbols 21

Annexe 2 : Hard Disk CNC architecture 30

1) Introduction

The ADV400 is a flexible CNC system, able to run different types of Milling or Lathe machines.

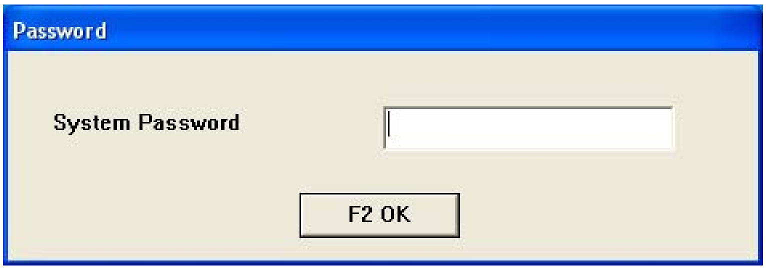
This manual gives the different options for setting the CNC controller to a dedicated machine:

- Setting machine and axis specifications
- Tuning the axis
- Writing custom M-codes
- Writing tool changer code
- And more

2) Passwords

The system contains 3 levels of use, protected with passwords.

The system password is asked every power-up of the system.
It also can be called at any time from the File menu.



With no password (pressing F2), the system will be at machine User level.
The User level permits to home the axis, to move manually the axis and to run the present part program.

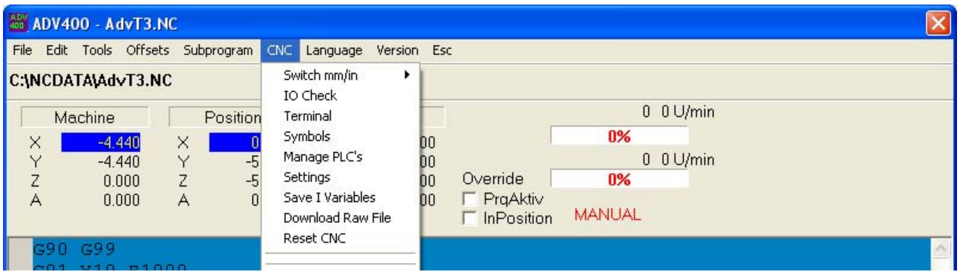
With the first password **400USER**, the system will be at machine Manager level.
This level permits also to modify the part program and to access to the different menus for managing the production (Tool menu, Work Offset menu, Sub-programs).

With the second password **400MC**, the system will be at machine Integration level.
This level permits to access all menus, permitting to perform the machine integration (PLCs, tuning...). This level has to be used for the machine integration.

It's the machine integrator decision to give only the first password or both passwords to the final machine user.

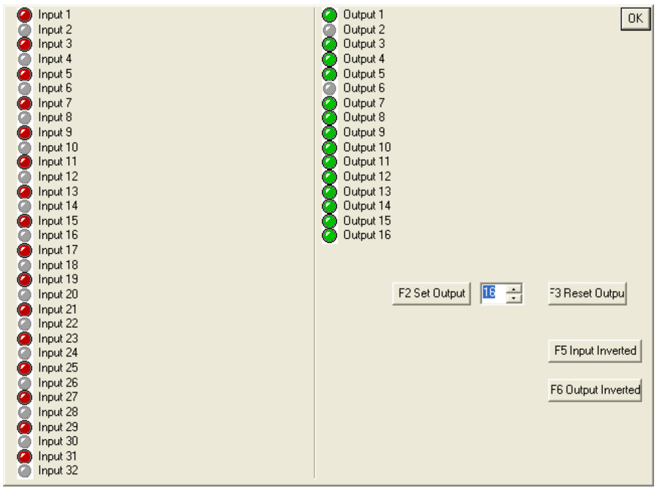
3) Integration Menu

On the Menu bar, the CNC menu permits to access to the different configuration menus.

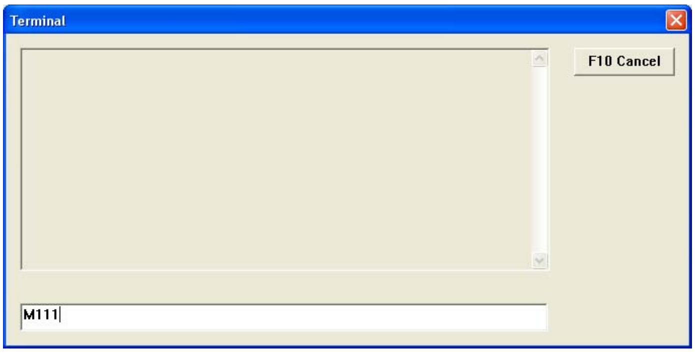


This menu gives access to different tools :

- **Switch mm/in** permits to put the CNC system in millimeters or in inches.
- **IO-check** shows the digital Inputs and Outputs state



- **Terminal** permits to send commands to the CNC controller



```
M111
```

This is mainly used for checks and debugging. The commands to be sent here are the motion controller commands.

- **Symbols** and **Manage-PLCs** are necessary tools to create and manage customs PLCs for the machine. Please, refer to “Custom PLCs” section for details.

- **Settings** opens the machine setting pages. Please, refer to “Machine Settings” section for details.

- **Save I Variables** performs a backup of the I variables of the system. The I variables contains several settings of the machine (like PID gains of axis, for example) and any change in one of these variables must be saved in order to recover the same setting after a shutdown of the system.

- **Download Raw File** permits to send a text file to the controller. This is used mainly for maintenance, as this file must contain only pure Motion Controller code.

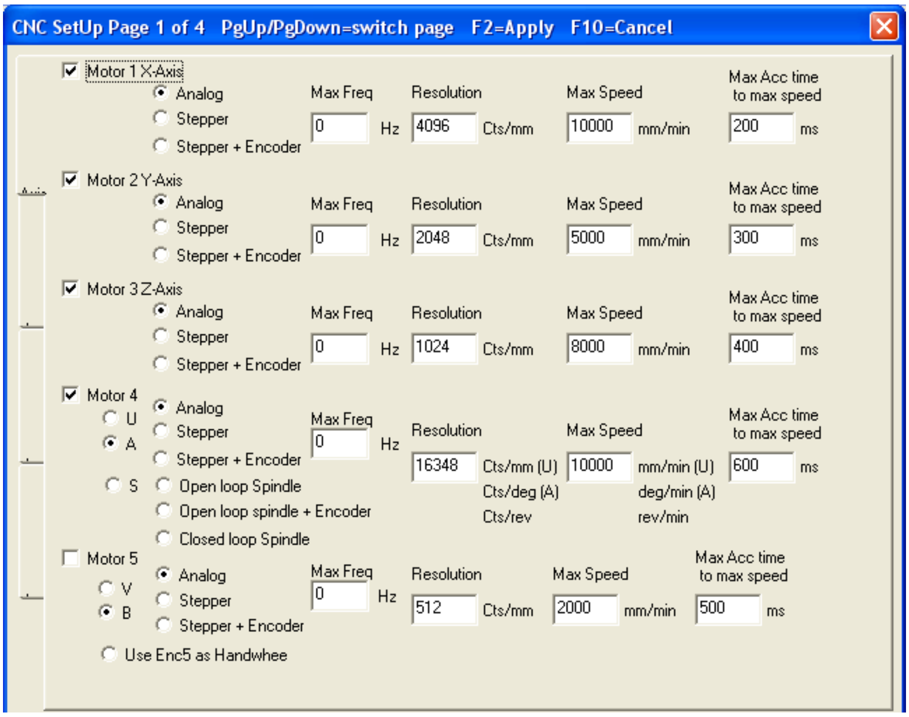
- **Reset CNC** just restart the CNC system, like after a shutdown of the system. If, for any reason, all the configuration has to be resent to the controller, this menu will perform this job. It’s necessary to remove power of drives or press the E-stop button before performing this action.

3) Machine settings

The setting of the machine is the first step to accomplish before using the axis.

There is four pages for the machine setting, permitting to define all details about the machine that has to be manage by this CNC controller (number of axis, type of each axis, spindle, limits, homing procedure...).

a) Setting page 1: general axis setup 1



CNC SetUp Page 1 of 4 PgUp/PgDown=switch page F2=Apply F10=Cancel

☑ Motor 1 X-Axis
● Analog ○ Stepper ○ Stepper + Encoder
Max Freq 0 Hz — Resolution 4096 Cts/mm — Max Speed 10000 mm/min — Max Acc time to max speed 200 ms

☑ Motor 2 Y-Axis
● Analog ○ Stepper ○ Stepper + Encoder
Max Freq 0 Hz — Resolution 2048 Cts/mm — Max Speed 5000 mm/min — Max Acc time to max speed 300 ms

☑ Motor 3 Z-Axis
● Analog ○ Stepper ○ Stepper + Encoder
Max Freq 0 Hz — Resolution 1024 Cts/mm — Max Speed 8000 mm/min — Max Acc time to max speed 400 ms

☑ Motor 4 ○ U ● A ○ S
● Analog ○ Stepper ○ Stepper + Encoder ○ Open loop Spindle ○ Open loop spindle + Encoder ○ Closed loop Spindle
Max Freq 0 Hz — Resolution 16348 Cts/mm (U) Cts/deg (A) Cts/rev — Max Speed 10000 mm/min (U) deg/min (A) rev/min — Max Acc time to max speed 600 ms

□ Motor 5 ○ V ● B
● Analog ○ Stepper ○ Stepper + Encoder
Max Freq 0 Hz — Resolution 512 Cts/mm — Max Speed 2000 mm/min — Max Acc time to max speed 500 ms
○ Use Enc5 as Handwhee

The ADV400 is capable of doing up to five axis.

To use an axis, the check box in front of motor number (Motor x) must be checked.
Motors 1, 2 and 3, respectively names X, Y and Z, always linear axis, can be analog (+/-10V) or stepper (pulses and direction) type of axis.

If Analog axis is selected, an encoder will have to be present on this axis.

In case of stepper axis, the ADV400 can work with or without real encoder feedback. The maximum frequency for pulse output must also be entered when an axis is used in stepper.

In both cases (analog or stepper), the resolution (counts per millimeters), the maximum speed (millimeters per minute) and the maximum acceleration (time in milliseconds from 0 to maximum speed) must be indicated.

If analog axis or stepper + encoder is selected, a position-loop tuning must be performed before this axis can be used (see section “Axis Tuning”).
If stepper (meaning with no encoder) is selected, the position-loop tuning must NOT be done. The encoder is simulated internally by the controller and the axis PID gains are automatically calculated.

Motor 4 can be an axis (linear named U or rotary named A). In this case, like motors 1, 2 and 3, this motor can be analog or stepper with the same setup.

Motor 4 can be also a spindle named S (open-loop with no encoder, open-loop with encoder, or closed-loop automatically with encoder). In this case, the analog type is used (no stepper).

The resolution (counts per millimeters if linear axis U, counts per degree if rotary axis A or counts per revolution in case of spindle), the maximum speed (millimeters per minute if linear axis U, degrees per minute if rotary axis A or revolution per minute in case of spindle) and the maximum acceleration (time in milliseconds from 0 to maximum speed) must be indicated.

If analog axis, stepper + encoder or closed-loop spindle is selected, a position-loop tuning must be performed before this axis can be used (see section “Axis Tuning”).

If stepper (meaning with no encoder) or open-loop spindle (with or without encoder) is selected, the position-loop tuning must NOT be done. For stepper case, the encoder is simulated internally by the controller and the axis PID gains are automatically calculated. For open-loop spindle case, the amplifier must be an inverter or an amplifier closing the velocity-loop internally.

Motor 5 can be an axis (linear named V or rotary named B). This motor is always an analog axis with encoder feedback.

The resolution (counts per millimeters in linear axis V or counts per degree if rotary axis B), the maximum speed (millimeters per minute if linear axis V, degrees per minute if rotary axis B) and the maximum acceleration (time in milliseconds from 0 to maximum speed) must be indicated.

A position-loop tuning must be performed before this axis can be used (see section “Axis Tuning”).

When the motor 5 is not used, the encoder feedback can be used for an external handwheel. To enable this function, uncheck to box “Motor 5” and check the box “Use ENC5 as Handwheel”.

In this case, some Inputs/Outputs will not be for general-purpose use anymore, but will have dedicated functions in order to be able to use an external box for manual movements of the axis:

INPUT7 will tell if we use the internal ADV400 functions or the external box buttons:
If INPUT7 is OFF -> use of internal ADV400 functions
If INPUT7 is ON -> use of external box buttons

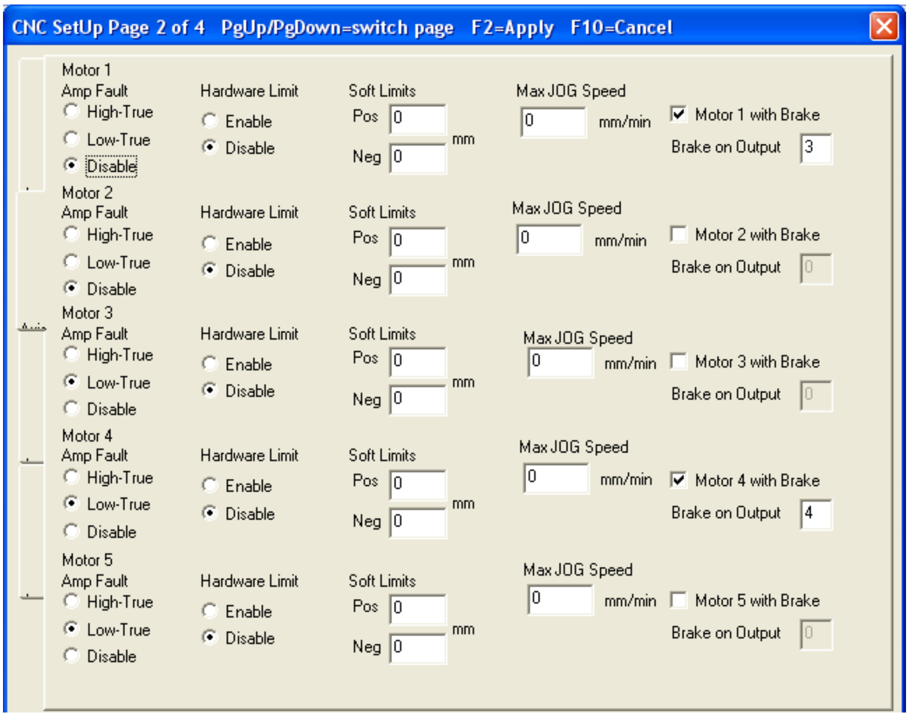
The external box buttons are:

INPUT9, INPUT10 and INPUT10 for axis select
With INPUT9=Off and INPUT10=Off and INPUT11=Off axis X is selected
With INPUT9=On and INPUT10=Off and INPUT11=Off axis Y is selected
With INPUT9=On and INPUT10=On and INPUT11=Off axis Z is selected
With INPUT9=Off and INPUT10=On and INPUT11=Off axis AorU is selected

INPUT13, INPUT14 and INPUT15 for step move size with handwheel
With INPUT13=On and INPUT14=Off and INPUT15=Off, axis moves of 0.001mm per Handwheel step
With INPUT13=On and INPUT14=On and INPUT15=Off, axis moves of 0.01mm per Handwheel step
With INPUT13=Off and INPUT14=On and INPUT15=Off, axis moves of 0.1mm per Handwheel step
With INPUT13=Off and INPUT14=On and INPUT15=On, axis moves of 0.2 mm per Handwheel step

INPUT12 for JOG axis in Minus direction
INPUT16 for JOG axis in Plus direction

b) Setting page 2: general axis setup 2



CNC SetUp Page 2 of 4 PgUp/PgDown=switch page F2=Apply F10=Cancel

	Amp Fault	Hardware Limit	Soft Limits		Max JOG Speed		
			Pos	Neg	mm/min		Brake on Output
Motor 1	○ High-True ○ Low-True ● Disable	○ Enable ● Disable	0	0	0	☑ Motor 1 with Brake	3
Motor 2	○ High-True ○ Low-True ● Disable	○ Enable ● Disable	0	0	0	☐ Motor 2 with Brake	0
Motor 3	○ High-True ● Low-True ○ Disable	○ Enable ● Disable	0	0	0	☐ Motor 3 with Brake	0
Motor 4	○ High-True ● Low-True ○ Disable	○ Enable ● Disable	0	0	0	☑ Motor 4 with Brake	4
Motor 5	○ High-True ● Low-True ○ Disable	○ Enable ● Disable	0	0	0	☐ Motor 5 with Brake	0

Soft Limits in mm.

If an axis is used (check box activated in Setting page 1), informations on this page must be entered.

The **Amp Fault** (amplifier fault signal) coming back from the drive can be true (Amplifier on default) at level high or at level low. Also, it's possible to disable the management of the Amplifier fault signal if the drive doesn't have one.

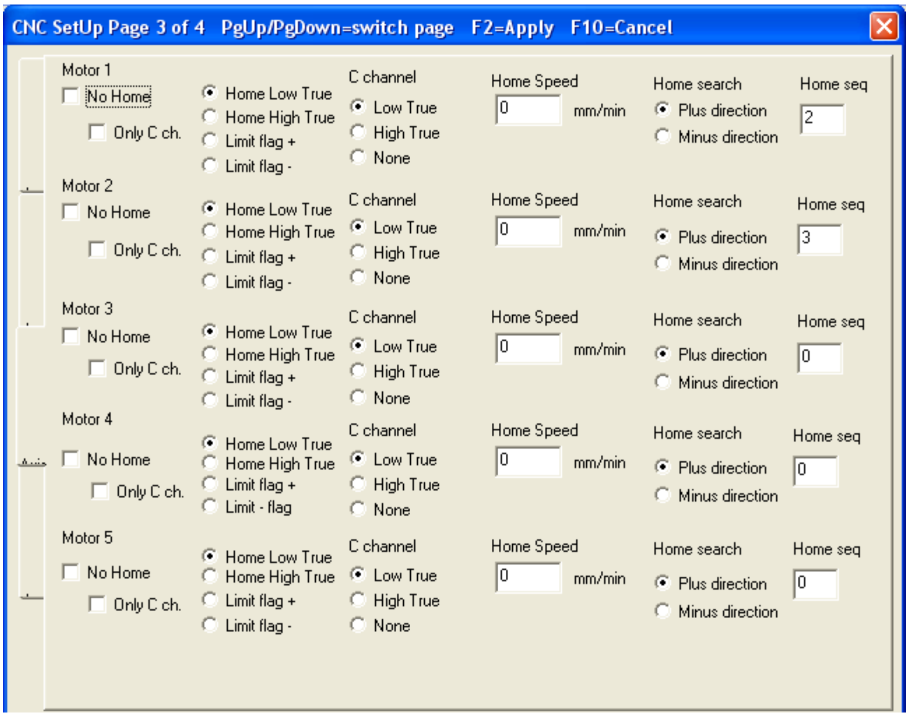
The **Hardware Limits** (plus and minus) can be enabled or disabled. If enabled, an hitting of one of this limits will automatically stop the axis.

The **Software Limits** in millimetres can be used to limit the travel of the axis. These limits are only for linear axis (X, Y, Z, U and V). A zero (0) value disables the software limit.

The **Max JOG Speed** indicated the maximum speed of the axis on manual JOG moves.

If a motor has an internal **Brake** (vertical axis, for example), a digital output must be affected to the brake and the ADV400 will automatically manage this output with the status (open-loop or closed-loop) of this motor. In this case, the check box "Motor x with Brake" must be activated and a number of digital output between 1 and 16 (see digital outputs on ADV400, connector OUT1) must be entered.

c) Setting page 3: axis homing setup



CNC SetUp Page 3 of 4 PgUp/PgDown=switch page F2=Apply F10=Cancel

			C channel	Home Speed	Home search	Home seq
Motor 1 □ No Home □ Only C ch.	● Home Low True ○ Home High True ○ Limit flag + ○ Limit flag -		● Low True ○ High True ○ None	0 mm/min	● Plus direction ○ Minus direction	2
Motor 2 □ No Home □ Only C ch.	● Home Low True ○ Home High True ○ Limit flag + ○ Limit flag -		● Low True ○ High True ○ None	0 mm/min	● Plus direction ○ Minus direction	3
Motor 3 □ No Home □ Only C ch.	● Home Low True ○ Home High True ○ Limit flag + ○ Limit flag -		● Low True ○ High True ○ None	0 mm/min	● Plus direction ○ Minus direction	0
Motor 4 □ No Home □ Only C ch.	● Home Low True ○ Home High True ○ Limit flag + ○ Limit - flag		● Low True ○ High True ○ None	0 mm/min	● Plus direction ○ Minus direction	0
Motor 5 □ No Home □ Only C ch.	● Home Low True ○ Home High True ○ Limit flag + ○ Limit flag -		● Low True ○ High True ○ None	0 mm/min	● Plus direction ○ Minus direction	0

If an axis is used (check box activated in Setting page 1), the informations on this page must be entered.

This page gives the setting of the way to do the homing (home reference) of each axis.

The Check box “No Home” indicates that no homing routine has to be performed on this axis. In this case, during the homing sequence, the zero position will be forced to this axis where it is positioned.
If this check box is activated, all informations for this axis on this page are useless.

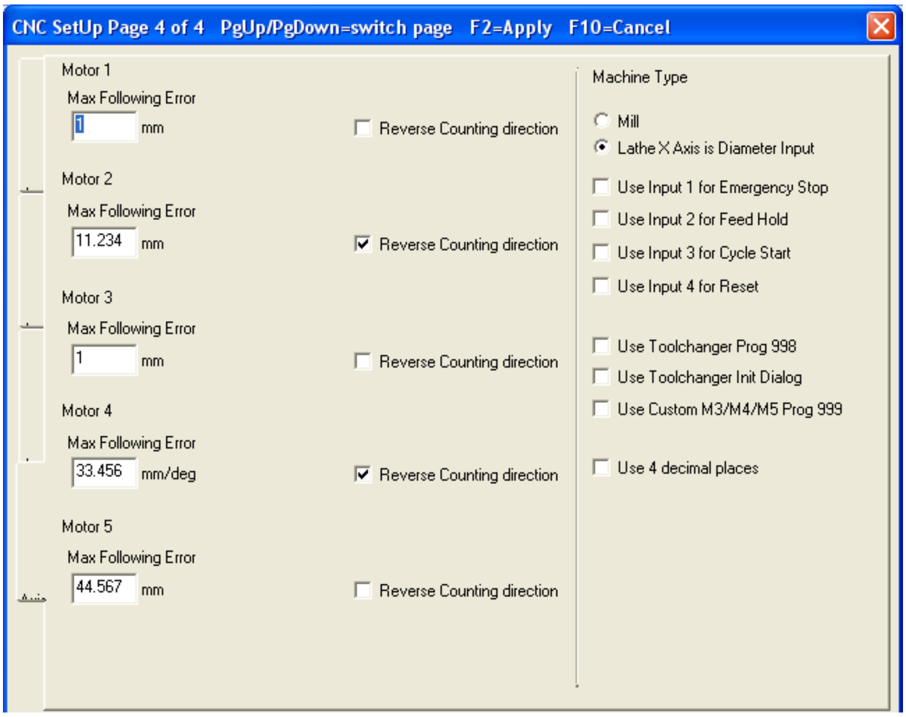
An axis reference can be done on the **C channel** (zero encoder) or on one of the fast inputs **Home Flag** (high-true or low-true), **Limit+** or **Limit-**. If one of these three fast inputs is selected, it’s also possible to use it with in combinaison with the **C channel**.
Depending of the encoder used, the **C channel** active can be **Low True** or **High True**.
- To do the homing routine on the **C channel** only, check the box “Only C ch.”.
- To do the homing on one of the fast input flags only, uncheck the box “Only C ch.”, check the box “None” under the C channel section and check the desired fast input (**Home Low True**, **Home High True**, **Limit flag** + or **Limit flag -).**
- To do the homing on a combinaison of the **C channel** and one of the fast input flags, uncheck the box “Only C ch.”, check the box “Low True” or “High True” under the C channel section and check the desired fast input (**Home Low True**, **Home High True**, **Limit flag** + or **Limit flag -).**

The **Home Speed** must be entered, in millimetres per minute.

Also, the direction for the **Home Search** has to be entered. The direction is the one when the axis is going to the selected flag direction.
The **Home Seq** box must be setted with a number between 1 and 5, indicating the desired sequence of the homing. Axis homing will be done one by one following this sequence (first axis will be this with number 1, second axis will be this with number 2… an so on).

If the Motor 4 is used as a spindle, no homing sequence will be done on this motor.

d) Setting page 4: miscellaneous



CNC SetUp Page 4 of 4 PgUp/PgDown=switch page F2=Apply F10=Cancel

Motor 1
Max Following Error 1 mm
□ Reverse Counting direction

Motor 2
Max Following Error 11.234 mm
☑ Reverse Counting direction

Motor 3
Max Following Error 1 mm
□ Reverse Counting direction

Motor 4
Max Following Error 33.456 mm/deg
☑ Reverse Counting direction

Motor 5
Max Following Error 44.567 mm
□ Reverse Counting direction

Machine Type
○ Mill
● Lathe X Axis is Diameter Input

□ Use Input 1 for Emergency Stop
□ Use Input 2 for Feed Hold
□ Use Input 3 for Cycle Start
□ Use Input 4 for Reset

□ Use Toolchanger Prog 998
□ Use Toolchanger Init Dialog
□ Use Custom M3/M4/M5 Prog 999

□ Use 4 decimal places

This page permits to indicate the **Maximum Following Error** allowed for an axis, in millimeters.
If for any reason, during operation, an axis has more following error than the one indicated here, all axis will be stopped and disabled and an error message will appear.

The **Reverse Counting direction** box permits to change the encoder counting direction. As the encoder counting direction must always be in correlation with the command output (a positive command on the analog or stepper output must move the motor in the direction where the encoder counts positive, and a negative command on the analog or stepper output must move the motor in the direction where the encoder counts negative), this check box permits to create this correlation if it's not the case by inverting the encoder counting direction.

On the right of this setting page, we have to indicate the **Machine Type** we want to manage (Mill or Lathe). A Lathe selection will make the X axis with a diameter input.

The check box "Use Input 1 for Emergency Stop" permits to use the INPUT1 as the ESTOP input, low true. If this box is checked, the INPUT1 must be high to have the machine running.

The check box "Use Input 2 for Feed Hold" permits to give the function Feed Hold to INPUT2.

The check box "Use Input 3 for Cycle Start" permits to give the function Cycle Start to INPUT3.

The check box "Use Input 4 for Reset " permits to give the function Reset to INPUT4.

The check box "Use Toolchanger Prog998" will create an automatic jump to sub-routine 998 when a tool code Txxyy is programmed. This permits to create some tool changing routines. Please, refer to Tool Changer programming for details.

The check box "Use Toolchanger Init Dialog" will give a special dialog menu at power-up of ADV400-CNC, allowing to initialise some special tool changer positions.
Please, refer to Tool Changer programming for details.

The check box "Use Custom M3/M4/M5 Prog999" allows to write custom M-codes for spindle in sub-routine 999. If this box is checked, these custom M-codes have to be written, otherwise the normal M3/M4/M5 codes of the system will be used.
Please, refer to custom M-codes programming for details.

The check box "Use 4 decimal places" permits to have 4 decimal digits in the axis position windows. Default is 3 digits.

5) Axis Tuning

For time being, the tuner tool of the Executive Program (Pewin32 or Pewin32Pro) has to be used to tune the axis.

The following steps have to be performed to performs this job:

- Boot you ADV400 system and let the HMI start
- Exit from the HMI with the "File /Exit" menu.
- Connect an external computer with Executive program on it to the RS232 connector of ADV400.
- Start Executive Program and tune axis with this external computer.
- When the tuning is finished, quit the executive program and, without shutting down the ADV400 controller, start again the ADV400 HMI (the program is called ADV400.EXE and is located on the "Hard Disk/CNC/" directory.
- Perform a saving of the I variables with the "CNC/Save_I_variables" menu of ADV400 HMI
- Reboot the ADV400 unit.

6) Custom PLCs

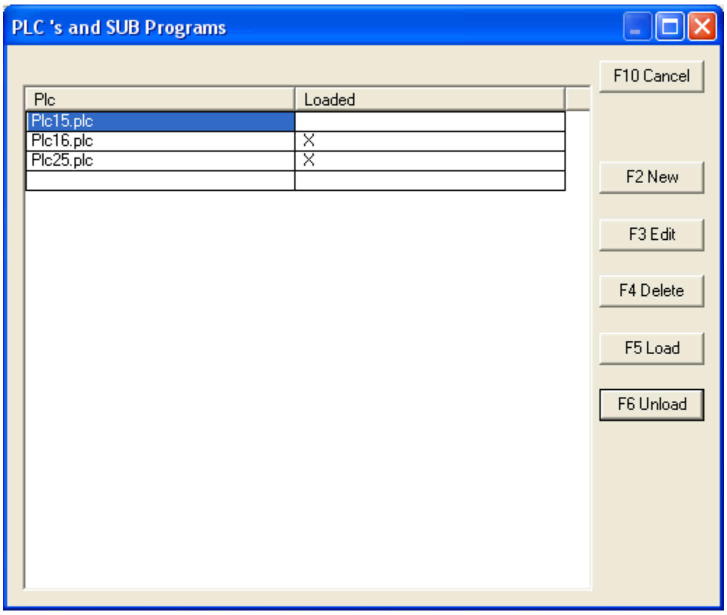
With this feature, it is possible to create customs PLCs for managing certain functions of the machine.

PLCs numbers 15 to 24 are available for these customs PLCs.

PLC25 is already present and permits to create some additional conditions for the use of some machine control buttons.

a) PLCs management

Under the CNC menu, a management page permits to create (New), Edit, Delete, Load and unload a custom PLC.



Plc	Loaded
Plc15.plc	
Plc16.plc	X
Plc25.plc	X

New (or F2 on the keyboard) permits to create a new custom PLC (opening text editor with blank page).

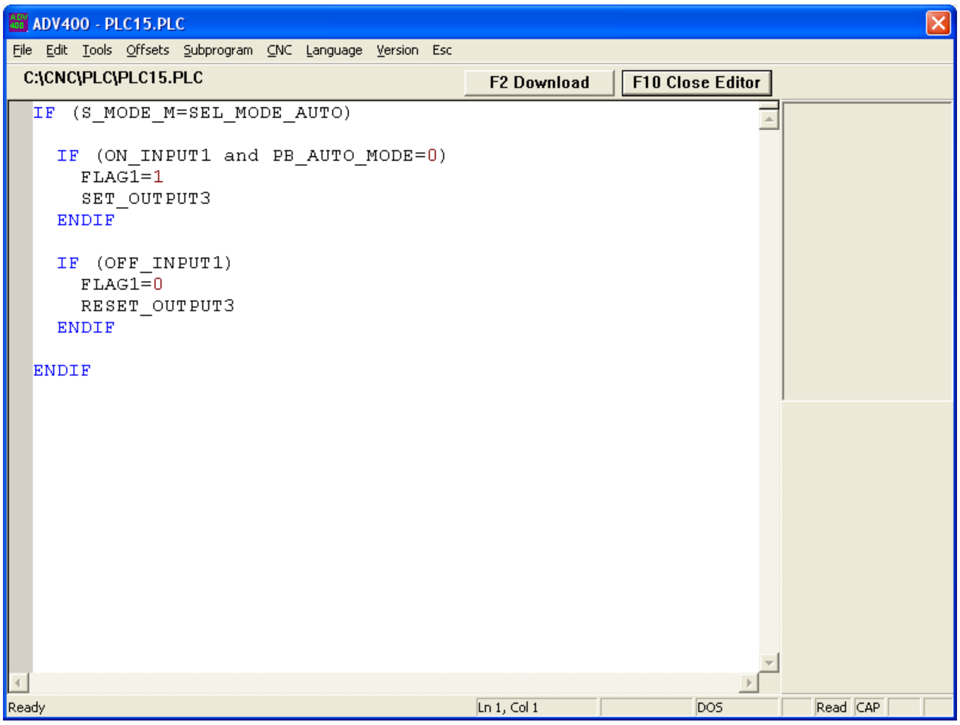
Edit (or F3 on the keyboard) open an existing custom PLC (opening text editor with this PLC inside) for consulting or modification.

Delete (or F4 on the keyboard) remove an existing custom PLC from the list.

Load (or F5 on the keyboard) loads an existing custom PLC in the controller and enables it (running). An “X” appears then in the Loaded section for this PLC and it will be used until it will be unloaded.

Unload (or F6 on the keyboard) removes an existing custom PLC from the controller. The “X” does not appear then in the Loaded section for this PLC and it will not be used until it will be loaded again.

When creating a new PLC or editing an existing PLC, an text editor is opening and the PLC code can be entered.



```
IF (S_MODE_M=SEL_MODE_AUTO)

  IF (ON_INPUT1 and PB_AUTO_MODE=0)
    FLAG1=1
    SET_OUTPUT3
  ENDIF

  IF (OFF_INPUT1)
    FLAG1=0
    RESET_OUTPUT3
  ENDIF

ENDIF
```

Download (or F2 on the keyboard) is sending the PLC to the controller and automatically uses it (like the Load button on the Managing PLC page)

Close Editor (or F10 on the keyboard) is providing a quit of this page, asking to save to entered code if not done.

b) Writing a PLC

As the Part Program, the Custom PLCs are using the symbol table.
Please, refer to "Table of Symbols" section for list of all available symbols.

It's also possible to create personal symbols with ADV400 free variables.

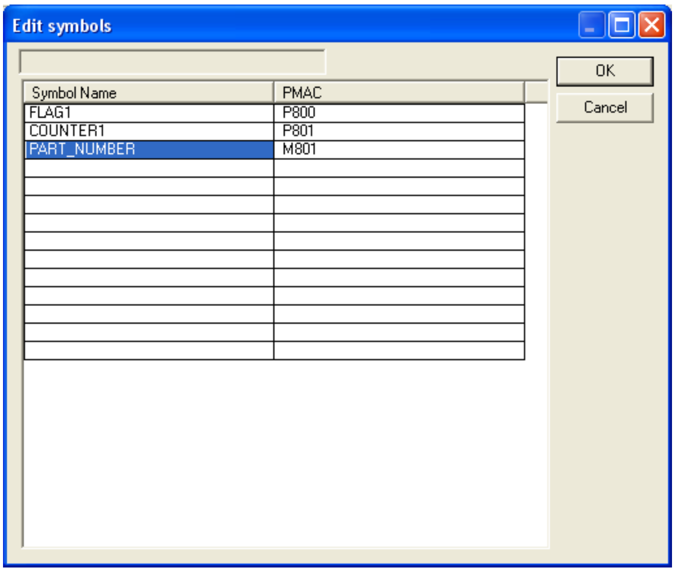
The free variables are:

- P800 to P1023
- Q800 to Q1023
- M800 to M1023

P and Q variables are 48-bits floating point format and permit to do any calculation or flag.

M variables are 24-bits format and permit also to do calculation of flags. They are normally used for pointing any memory of the ADV400 system, but the needed memories for writing applications are already done, accessible with the Table of Symbols.

Under the CNC menu, there is a table to create symbols for free variables:



Symbol Name	PMAC
FLAG1	P800
COUNTER1	P801
PART_NUMBER	M801

PLC15 to PLC24 are available for customs PLCs.

A special PLC25 is already present in the system and HAS TO BE DOWNLOADED IN THE CNC in order to have all buttons (Cycle Start, Feed Hold....) working.
Please, refer to the “Specific PLC25” section for details.

A PLC is scanned all the time, asynchronously of the part program.
It permits to read inputs, write outputs, test conditions.....

- testing an input

```
............................................
If (ON_INPUT1)   ; test is Input1 true
  ; action
Endif
```

```
............................................
If (OFF_INPUT4)   ; test is Input4 false
  ; action
Else
  ; other action
Endif
```

- waiting state of an input

```
.............................................
While (ON_INPUT2)  ; wait as long as Input2 is true
  ; action
Endw
```

- setting an output

```
.............................................
SET_OUTPUT2        ; set Ouput2

.............................................
RESET_OUTPUT3   ; Reset Ouput3
```

- testing or waiting an information

  With a If condition or with a While loop, it's also possible to test or wait some other information coming from the CNC.

  The "Annexe 5 : Table of Symbols" gives the list of information available.

```
.............................................
If (CS_SPND_AT_ZERO != 0)  ; test is spindle is at zero speed
 ; action
Endif

.............................................
While (CS_SPND_AT_SPEED = 0)   ; wait that spindle is at programmed speed.
Endif
```

- Using a timer

```
SET_OUTPUT2                    ; set ouput2
USER_TIMER_1=150               ; timer of 150ms
While (USER_TIMER_1>0)         ; wait timer finished
Endw
RESET_OUTPUT2                  ; reset ouput2
```

c) Specific PLC25

The PLC25 permits to add some conditions to the use of buttons.

The buttons managed in this PLC25 where conditions are:

- MANUAL mode button

- AUTO mode button
- MDI mode button
- HOME mode button
- CYCLE START button
- FEED HOLD button
- JOG PLUS button
- JOG MINUS button
- SINGLE MODE button
- OPTIONNAL MODE button
- BLOCK DELETE button

As an example, here is the standard code (no conditions) for the CYCLE START button:

```
If (PB_CYCLE_START!=0)
  IPB_CYCLE_START = 1
Else
  IPB_CYCLE_START = 0
Endif
```

If, for example, we want to add the condition that Input 1 is ON to allow to do a Cycle Start of the program, we would modify this section like this:

```
If (PB_CYCLE_START!=0 and ON_INPUT3)
  IPB_CYCLE_START = 1
Else
  IPB_CYCLE_START = 0
Endif
```

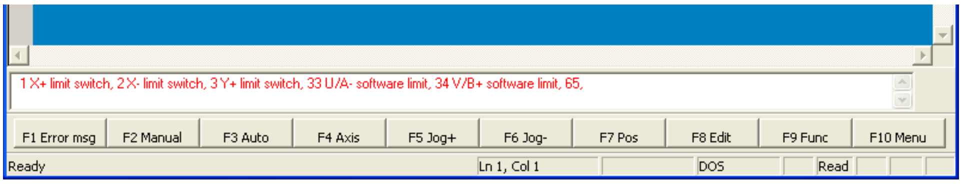
As another example, if we need to reset the Output 2 and 3 when we go in manual mode, we would do:

```
        If (PB_MANUAL_MODE!=0)
          IPB_MANUAL_MODE = 1
          RESET_OUTPUT2
          RESET_OUTPUT3
        Else
          IPB_MANUAL_MODE = 0
Endif
```

d) Messages

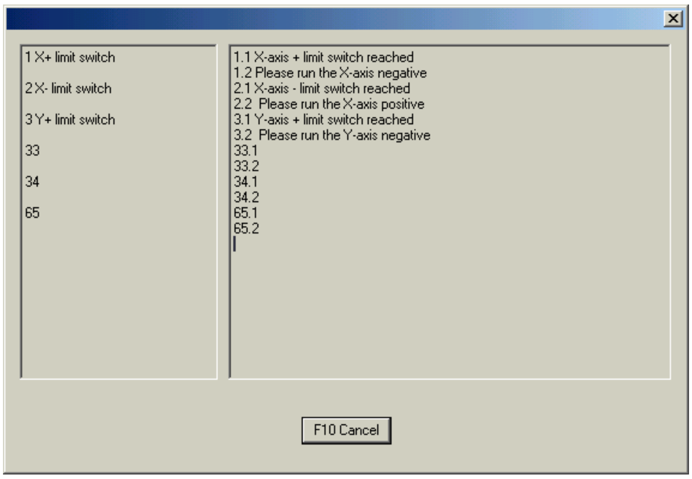
From PLCs, it’s possible to create messages (information, warning or error) for the machine user.

These messages will then appear in the message window on the bottom of the main screen



1 X+ limit switch, 2 X- limit switch, 3 Y+ limit switch, 33 U/A- software limit, 34 V/B+ software limit, 65,

When any message is present, the F1 key permits to enter to the detailed message window.



1 X+ limit switch
2 X- limit switch
3 Y+ limit switch
33
34
65

1.1 X-axis + limit switch reached
1.2 Please run the X-axis negative
2.1 X-axis - limit switch reached
2.2 Please run the X-axis positive
3.1 Y-axis + limit switch reached
3.2 Please run the Y-axis negative
33.1
33.2
34.1
34.2
65.1
65.2

A maximum of 96 messages are possible, but messages 1 to 59 are reserved for the system.

So, messages 60 to 96 can be used by customes PLCs.

To display or remove a message, the following code has to be used:

```
SET_ER_60           ; this will display message number 60
RESET_ER_71         ; this will remove message number 71
```

It is also possible to test if a message is displayed (present) or not:

```
IF (ERH_81)         ; test if message 81 is display (present)
IF (ERL_68)         ; test if message 68 is not present
```

The text message has to be entered in the file “Adv4Err.dtx” which is a pure text file, present on the system under the directory “Hard Disk/CNC/01” for the first language English, or under “Hard Disk/CNC/02” for the second language.

On this file, any message is contained in 3 lines.

For the message 60, for example, we find lines:

60
60.1
60.2

The first line of a message is the one displayed on the bottom of the main screen.

In the detailed message window, all three lines will appear, permitting to give detailed messages.

For example, if we want that message 65 gives an error about a door open:

65 DOOR OPEN
65.1 Please, close the door
65.2 to allow cycle starting

e) Table of Symbols

The Table of Symbols present of the system contains all symbols permitting to access to all system information, like Input and Output, status of axis, status of spindle...

These symbols can be used mainly in custom PLCs, but it's also possible to use them in part programs or in the sub-routines.

Please, refer to "Annexe1: Table of Symbols" to have the complete list.

7) Custom M-codes programming

In several applications, it's needed to create custom M-codes to use them in the part program in order to perform some specific actions.

This sub-routine 999 allows to do this.

Some M-codes are reserved, already used by the system, and then cannot be create in this sub-routine.

List of reserved M-codes:

M00	Program Stop
M01	Optional Stop
M02	Program End & Rewind
M19	Spindle orient
M30	Program End & Rewind
M50	C-axe call
M51	Spindle call
M98	Subprogram Call

The M-codes M03, M04, and M05 are usually used for a spindle. If a spindle is present in the system, these codes are reserved and cannot be created in this sub-routine.

If no spindle is present, it's possible to use M03, M04 and M05 codes in this sub-routine, by checking the box "Use Custom M3/M4/M5 Prog999" in the machine Setting Page 4.

The structure of sub-routine 999 is very simple.

The part program M-code will just call the same label number than the code number:

M06 code will call sub-routine 999 at label N6
M09 code will call sub-routine 999 at label N9
M10 code will call sub-routine 999 at label N10
M12 code will call sub-routine 999 at label N12
M252 code will call sub-routine 999 at label N252

And so on.

In this sub-routine 999, a "RET" command has to be putted at the end of the programmed code to go out of this sub-routine and go back in the main program.

As an example, here is the code for using M07 code setting the output 1 and the M08 code for resetting the output 1.

The "Dwell0" command permits to wait that a previous move is finished before making the action of this M-code.
If no "Dwell0" is putted in front of the action, the action will be performed before the previous move is finished, because of the ahead calculation of the CNC.

```
// Prog 999
;
 ;
// M07 code
N7
  Dwell0
 SET_OUTPUT1        ; set Ouput1
RET

// M08 code
N8
 Dwell0
 RESET_OUTPUT1       ; reset Ouput1
RET
```

8) Tool Changer

a) Sub-routine 998

The sub-routine 998 has to be used for creating the code for a tool changer.

To have this function working, the check box "Use Toolchanger Prog998" must be checked in the machine Setting Page 4.

In this case, when the T-code Txxyy is programmed in the part program, two jumps to this sub-routine will automatically be done.

A first jump on label N1000 will be done with the actual Tool offset, permitting to create some axis movements out with the actual tool offset.

A second jump on label N2000 will then be done with the new tool offset, the one called by Txxyy T-code, permitting to create axis movements in with the new tool offset.

At the end of these two sections, a "RET" command must be present to tell that this section is finished (jump back to main program). Even if one of the two labels has no code because one of thes two jumps is not needed, the "RET" command must be present.

Structure of empty sub-routine 998:

```
// Prog 998
// Create your tool changer in this file
// put your code before new tool offset at label N1000
// put your code with new tool offset at label N2000

// here your code before tacking new tool offset
N1000
   ; put your code here
RET

// here your code after tacking new tool offset
N2000
   ; put your code here
RET
```

As an example, we decide that we move X and Y axis to positions 0 before we change tool, than we ask to a PLC with variable P810 to make the tool change, then we move back X and Y axis to position 10.

In this example, there is some "Dwell0" commands.

A Dwell0 permits to wait that a previous axis movement is finished before douing the job after it.
A Dwell0 must be always present in a "While" loop of a part program (as sub-routines are running in part programs).

```
// Prog 998
// Create your tool changer in this file
// put your code before new tool offset at label N1000
// put your code with new tool offset at label N2000

// here your code before tacking new tool offset
```

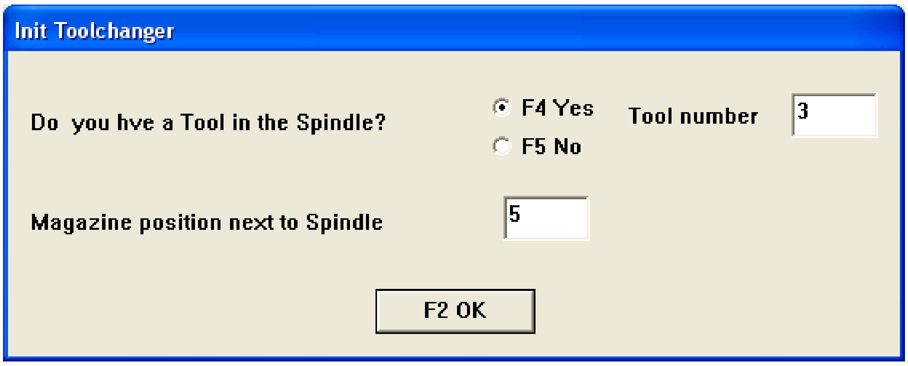
```
N1000
   G00 X0 Y0          ; move axis with actual tool offset
   Dwell0             ; this waits previous movement is finished
   P801=1             ; ask PLC to make tool change
   While (P810=1)      ; wait that the tool change is finished
     Dwell0           ; this means that PLC puts back P801 to 0
   Endwhile
RET

// here your code after tacking new tool offset
N2000
   G00 X10 Y10        ; move axis with new tool offset
RET
```

b) Tool changer initialisation

With certain types of Tool Changer, it is necessary to initialise some datas every power-up of the CNC to indicate to the system what tool is present....

For this matter, when the box "Use Toolchanger Init Dialog" is checked in machine Setting page 4, a dialog menu is appearing at power-up the ADV400-CNC software:



Init Toolchanger

Do you hve a Tool in the Spindle? ● F4 Yes ○ F5 No Tool number: 3

Magazine position next to Spindle: 5

These two values, tool number and magazine position, are just memorized on some variables that the tool changing code can then use later.

The first value is going in the variable P355, named TOOL_IN_SPINDLE

The second value is going in the variable P356, named MAGAZINE_POSITION

The first idea is to use these two variables for a tool changer where an initialisation has to be made every power-up of the system, telling what tool is in the spindle at what is the magazine position next to the spindle. But for any other type of tool changer, these two variables can be used in another way, like telling what tool is active at power-up or whatelse.

Annexe 1 : Table of Symbols

```
;************************************************************************
; INPUTS / OUTPUTS
;************************************************************************
;      for PLCs and Part Programs
; Test an INPUT true   : If (ON_INPUTx)
; Test an INPUT false  : If (OFF_INPUTx)
; Set an OUTPUT        : SET_OUTPUTx
; Reset an OUTPUT      : RESET_OUTPUTx
; Test an OUTPUT true  : If (ON_OUTPUTx)
; Test an OUTPUT false : If (OFF_OUTPUTx)
;      for Part Programs only
; Set an OUTPUT Sync   : SETS_OUTPUTx
; Reset an OUTPUT Sync : RESETS_OUTPUTx
;************************************************************************
```

;; INPUTS TRUE

SYMBOL	Comment
ON_INPUT1	Input 1 true
ON_INPUT2	Input 2 true
ON_INPUT3	Input 3 true
ON_INPUT4	Input 4 true
ON_INPUT5	Input 5 true
ON_INPUT6	Input 6 true
ON_INPUT7	Input 7 true
ON_INPUT8	Input 8 true
ON_INPUT9	Input 9 true
ON_INPUT10	Input 10 true
ON_INPUT11	Input 11 true
ON_INPUT12	Input 12 true
ON_INPUT13	Input 13 true
ON_INPUT14	Input 14 true
ON_INPUT15	Input 15 true
ON_INPUT16	Input 16 true
ON_INPUT17	Input 17 true
ON_INPUT18	Input 18 true
ON_INPUT19	Input 19 true
ON_INPUT20	Input 20 true
ON_INPUT21	Input 21 true
ON_INPUT22	Input 22 true
ON_INPUT23	Input 23 true
ON_INPUT24	Input 24 true
ON_INPUT25	Input 25 true
ON_INPUT26	Input 26 true
ON_INPUT27	Input 27 true
ON_INPUT28	Input 28 true
ON_INPUT29	Input 29 true
ON_INPUT30	Input 30 true
ON_INPUT31	Input 31 true
ON_INPUT32	Input 32 true

;; INPUTS FALSE

SYMBOL	Comment
OFF_INPUT1	Input 1 false

OFF_INPUT2	Input 2 false
OFF_INPUT3	Input 3 false
OFF_INPUT4	Input 4 false
OFF_INPUT5	Input 5 false
OFF_INPUT6	Input 6 false
OFF_INPUT7	Input 7 false
OFF_INPUT8	Input 8 false
OFF_INPUT9	Input 9 false
OFF_INPUT10	Input 10 false
OFF_INPUT11	Input 11 false
OFF_INPUT12	Input 12 false
OFF_INPUT13	Input 13 false
OFF_INPUT14	Input 14 false
OFF_INPUT15	Input 15 false
OFF_INPUT16	Input 16 false
OFF_INPUT17	Input 17 false
OFF_INPUT18	Input 18 false
OFF_INPUT19	Input 19 false
OFF_INPUT20	Input 20 false
OFF_INPUT21	Input 21 false
OFF_INPUT22	Input 22 false
OFF_INPUT23	Input 23 false
OFF_INPUT24	Input 24 false
OFF_INPUT25	Input 25 false
OFF_INPUT26	Input 26 false
OFF_INPUT27	Input 27 false
OFF_INPUT28	Input 28 false
OFF_INPUT29	Input 29 false
OFF_INPUT30	Input 30 false
OFF_INPUT31	Input 31 false
OFF_INPUT32	Input 32 false

;; SET OUTPUTS NON SYNCHRONE

SYMBOL	Comment
SET_OUTPUT1	Set Output 1 true
SET_OUTPUT2	Set Output 2 true
SET_OUTPUT3	Set Output 3 true
SET_OUTPUT4	Set Output 4 true
SET_OUTPUT5	Set Output 5 true
SET_OUTPUT6	Set Output 6 true
SET_OUTPUT7	Set Output 7 true
SET_OUTPUT8	Set Output 8 true
SET_OUTPUT9	Set Output 9 true
SET_OUTPUT10	Set Output 10 true
SET_OUTPUT11	Set Output 11 true
SET_OUTPUT12	Set Output 12 true
SET_OUTPUT13	Set Output 13 true
SET_OUTPUT14	Set Output 14 true
SET_OUTPUT15	Set Output 15 true
SET_OUTPUT16	Set Output 16 true

;; SET OUTPUTS SYNCHRONE for prog only

SYMBOL	Comment
SETS_OUTPUT1	Set Output 1 true synchronously (for Part Prog only)
SETS_OUTPUT2	Set Output 2 true synchronously (for Part Prog only)
SETS_OUTPUT3	Set Output 3 true synchronously (for Part Prog only)
SETS_OUTPUT4	Set Output 4 true synchronously (for Part Prog only)
SETS_OUTPUT5	Set Output 5 true synchronously (for Part Prog only)
SETS_OUTPUT6	Set Output 6 true synchronously (for Part Prog only)

SETS_OUTPUT7	Set Output 7 true synchronously (for Part Prog only)
SETS_OUTPUT8	Set Output 8 true synchronously (for Part Prog only)
SETS_OUTPUT9	Set Output 9 true synchronously (for Part Prog only)
SETS_OUTPUT10	Set Output 10 true synchronously (for Part Prog only)
SETS_OUTPUT11	Set Output 11 true synchronously (for Part Prog only)
SETS_OUTPUT12	Set Output 12 true synchronously (for Part Prog only)
SETS_OUTPUT13	Set Output 13 true synchronously (for Part Prog only)
SETS_OUTPUT14	Set Output 14 true synchronously (for Part Prog only)
SETS_OUTPUT15	Set Output 15 true synchronously (for Part Prog only)
SETS_OUTPUT16	Set Output 16 true synchronously (for Part Prog only)

;; RESET OUTPUTS

SYMBOL	Comment
RESET_OUTPUT1	Reset Output 1
RESET_OUTPUT2	Reset Output 2
RESET_OUTPUT3	Reset Output 3
RESET_OUTPUT4	Reset Output 4
RESET_OUTPUT5	Reset Output 5
RESET_OUTPUT6	Reset Output 6
RESET_OUTPUT7	Reset Output 7
RESET_OUTPUT8	Reset Output 8
RESET_OUTPUT9	Reset Output 9
RESET_OUTPUT10	Reset Output 10
RESET_OUTPUT11	Reset Output 11
RESET_OUTPUT12	Reset Output 12
RESET_OUTPUT13	Reset Output 13
RESET_OUTPUT14	Reset Output 14
RESET_OUTPUT15	Reset Output 15
RESET_OUTPUT16	Reset Output 16

;; RESET OUTPUTS SYNCHRONE for prog only

SYMBOL	Comment
RESETS_OUTPUT1	Reset Output 1 synchronously
RESETS_OUTPUT2	Reset Output 2 synchronously
RESETS_OUTPUT3	Reset Output 3 synchronously
RESETS_OUTPUT4	Reset Output 4 synchronously
RESETS_OUTPUT5	Reset Output 5 synchronously
RESETS_OUTPUT6	Reset Output 6 synchronously
RESETS_OUTPUT7	Reset Output 7 synchronously
RESETS_OUTPUT8	Reset Output 8 synchronously
RESETS_OUTPUT9	Reset Output 9 synchronously
RESETS_OUTPUT10	Reset Output 10 synchronously
RESETS_OUTPUT11	Reset Output 12 synchronously
RESETS_OUTPUT12	Reset Output 13 synchronously
RESETS_OUTPUT13	Reset Output 14 synchronously
RESETS_OUTPUT14	Reset Output 15 synchronously
RESETS_OUTPUT15	Reset Output 16 synchronously
RESETS_OUTPUT16	Reset Output 17 synchronously

;; OUTPUTS ON

SYMBOL	Comment
ON_OUTPUT1	Output 1 true
ON_OUTPUT2	Output 2 true
ON_OUTPUT3	Output 3 true
ON_OUTPUT4	Output 4 true
ON_OUTPUT5	Output 5 true
ON_OUTPUT6	Output 6 true
ON_OUTPUT7	Output 7 true
ON_OUTPUT8	Output 8 true

ON_OUTPUT9	Output 9 true
ON_OUTPUT10	Output 10 true
ON_OUTPUT11	Output 11 true
ON_OUTPUT12	Output 12 true
ON_OUTPUT13	Output 13 true
ON_OUTPUT14	Output 14 true
ON_OUTPUT15	Output 15 true
ON_OUTPUT16	Output 16 true

;; OUTPUTS OFF

SYMBOL	Comment
OFF_OUTPUT1	Output 1 false
OFF_OUTPUT2	Output 2 false
OFF_OUTPUT3	Output 3 false
OFF_OUTPUT4	Output 4 false
OFF_OUTPUT5	Output 5 false
OFF_OUTPUT6	Output 6 false
OFF_OUTPUT7	Output 7 false
OFF_OUTPUT8	Output 8 false
OFF_OUTPUT9	Output 9 false
OFF_OUTPUT10	Output 10 false
OFF_OUTPUT11	Output 11 false
OFF_OUTPUT12	Output 12 false
OFF_OUTPUT13	Output 13 false
OFF_OUTPUT14	Output 14 false
OFF_OUTPUT15	Output 15 false
OFF_OUTPUT16	Output 16 false

```
;***************************************************************************
; TIMERS
;***************************************************************************
; example of a timer of 100ms
;   USER_TIMER_1=100
;   While (USER_TIMER_1>0)
;   Endwhile
;***************************************************************************
```

SYMBOL	Comment
USER_TIMER_1	User timer 1 (for PLC)
USER_TIMER_2	User timer 2 (for PLC)
USER_TIMER_3	User timer 3 (for PLC)
USER_TIMER_4	User timer 4 (for PLC)
USER_TIMER_5	User timer 5 (for PLC)
USER_TIMER_6	User timer 6 (for PLC)
USER_TIMER_7	User timer 7 (for PLC)
USER_TIMER_8	User timer 8 (for PLC)

```
;***************************************************************************
; STATUS
;***************************************************************************
; test if a status is false : If (CS_SPND_AT_SPEED=0)
; test if a status is true  : If (CS_SPND_AT_SPEED!=0)
;***************************************************************************
```

SYMBOL	Comment

;; CS_MACHINE_LOCK	Not used
;; CS_RESET	Not used
CS_SINGLE_BLOCK	Part program single block status
CS_OPT_STOP	Part program optional stop status
CS_BLOCK_DELETE	Part program block delete status
;; CS_CLNT_FLOOD	Not used
;; CS_CLNT_MIST	Not used
;; CS_JOG_PLUS	Not used
;; CS_JOG_MINUS	Not used
;; CS_JOG_STOP	Not used
;; CS_JOG_RETURN	Not used
;; CS_HOME	Not used
;; CS_CYCLE_START	Not used
;; CS_CYCLE_RESTART	Not used
;; CS_FEED_HOLD	Not used
;; CS_DRY_RUN	Not used
;; CS_SPND_CW	Not used
;; CS_SPND_CCW	Not used
;; CS_SPND_BRAKE	Not used
;; CS_SPND_NEUTRAL	Not used
;; CS_PRG_REWIND	Not used
;; CS_CHUCK_OPEN	Not used
;; CS_CHUCK_CLOSE	Not used
;; CS_CHUCK_OD_GRIP	Not used
;; CS_CHUCK_ID_GRIP	Not used
;; CS_TOOL_RELEASE	Not used
;; CS_TOOL_ENGAGE	Not used
CS_SPND_DETECT	Spindle speed detection status
CS_SPND_CSS	Spindle constant surface speed status
CS_SPND_AT_SPEED	Spindle at speed status
CS_SPND_AT_ZERO	Spindle at zero speed status
CS_SPND_FEED	Spindle feed per revolution status

```
;**************************************************************************
; BUTTONS
;**************************************************************************
; test if a button is not pressed  : If (PB_AUTO_MODE=0)
; test if a button is pressed      : If (PB_AUTO_MODE!=0)
;**************************************************************************
```

SYMBOL	Comment
;; PB_MACHINE_LOCK	Not used
PB_RESET	Reset button
PB_SINGLE_BLOCK	Single block button
PB_OPT_STOP	Optional stop button
PB_BLOCK_DELETE	Block delete button
;; PB_CLNT_FLOOD	Not used
;; PB_CLNT_MIST	Not used
PB_JOG_PLUS	Jog Plus button
PB_JOG_MINUS	Jog Minus button
;; PB_JOG_STOP	Not used
;; PB_JOG_RETURN	Not used
PB_HOME	Home mode button

PB_CYCLE_START	Cycle start button
;;PB_CYCLE_RESTART	Not used
PB_FEED_HOLD	Feed Hold button
;; PB_DRY_RUN	Not used
;; PB_SPND_CW	Not used
;; PB_SPND_CCW	Not used
;; PB_SPND_BRAKE	Not used
;; PB_SPND_NEUTRAL	Not used
;; PB_PRG_REWIND	Not used
;; PB_CHUCK_OPEN	Not used
;; PB_CHUCK_CLOSE	Not used
;;PB_CHUCK_OD_GRIP	Not used
;;PB_CHUCK_ID_GRIP	Not used
;; PB_TOOL_RELEASE	Not used
;; PB_TOOL_ENGAGE	Not used
PB_AUTO_MODE	Auto mode button
PB_MANUAL_MODE	Manual mode button
PB_MDI_MODE	MDI mode button
PB_HOME_MODE	Home mode button
;; PB_FREE_TWO	Not used

```
;***************************************************************************
; Informations
;***************************************************************************
; just read a number value
;***************************************************************************
```

SYMBOL	Comment
; VS_SPINDLE_RPM_M	Not used
VS_SPINDLE_MAX_RPM_M	Maximum spindle speed
VS_SPINDLE_MAX_LIM_M	Maximum spindle speed
VS_SPINDLE_CMD_RPM_M	Actual commanded spindle speed
VS_SPINDLE_ACT_RPM_M	Actual spindle speed
; VS_SPINDLE_COUNTS_REV_M	Not used
; VS_SPINDLE_CSS_M	Not used
; VS_SPINDLE_CSS_UNITS_M	Not used
VS_SPINDLE_OVERRIDE_M	Actual spindle override
; VS_HAND_STEP_M	Not used
VS_FEED_OVERRIDE_M	Actual feed override
VS_RAPID_OVERRIDE_M	Actual rapid override

```
;***************************************************************************
; Mode, axis selected....
;***************************************************************************
; test the actual mode   : If (S_MODE_M=SEL_MODE_AUTO)
; test the axis selected : If (S_AXIS_M=SEL_AXIS_X)
; test of spindle status : If (S_SPND_M=SEL_SPND_CCW)
;***************************************************************************
```

SYMBOL	Comment
S_MODE_M	Tell actual mode
SEL_MODE_AUTO	Auto mode
SEL_MODE_MANUAL	Manual mode
SEL_MODE_MDI	MDI mode

SEL_MODE_HOME	Home mode

SYMBOL	Comment
S_AXIS_M	Tell axis selected
SEL_AXIS_X	X axis selected
SEL_AXIS_Y	Y axis selected
SEL_AXIS_Z	Z axis selected
SEL_AXIS_A	A/U axis selected
SEL_AXIS_B	B/V axis selected

SYMBOL	Comment
S_SPND_M	Tell spindle status
SEL_SPND_CCW	Spindle turning CCW
SEL_SPND_OFF	Spindle stopped
SEL_SPND_CW	Spindle turning CW
;SEL_SPND_ORIENT1	Not used
;SEL_SPND_ORIENT2	Not used
;SEL_SPND_LOCK	Not used

```
;**************************************************************************
; Informations
;**************************************************************************
; just read a number value
;**************************************************************************
```

SYMBOL	Comment
VS_TOOL_NUM_M	Give active tool and correction numbers
VS_X_ABS_M	Give actual offset (G54-G59 + G52 + Tool) of X axis
VS_Y_ABS_M	Give actual offset (G54-G59 + G52 + Tool) of Y axis
VS_Z_ABS_M	Give actual offset (G54-G59 + G52 + Tool) of Z axis

```
;**************************************************************************
; Informations
;**************************************************************************
; just read a number value or a bit
; possibility to write DAC5 or DAC6 from -512 to +512 for -10V to +10V
;**************************************************************************
```

SYMBOL	Comment
SERVO_COUNTER_M	Servo counter incremented by one every Servo inter.
HANDLE5IN_COUNT_M	Encoder 5 value (if motor 5 not used)
HANDLE6IN_COUNT_M	Encoder 6 value
ADC1_M	ADC1 value (feed potentiometer)
ADC2_M	ADC2 value
ADC3_M	ADC3 value
ADC4_M	ADC4 value
; TIMER_1_M	
; TIMER_2_M	
; TIMER_3_M	
; TIMER_4_M	
; FEED_HOLD_M	
PROG_RUNNING_M	Program running bit

HOME_COMPLETE_1_M	Axis 1 (X) home completed bit
HOME_COMPLETE_2_M	Axis 2 (Y) home completed bit
HOME_COMPLETE_3_M	Axis 3 (Z) home completed bit
HOME_COMPLETE_4_M	Axis 4 (A/U) home completed bit
HOME_COMPLETE_5_M	Axis 5 (B/V) home completed bit
; PROG_STEPPING_M	
; MOTION_MODE_M	
IN_POSITION_M	Coordinate system one all axis in position
; CONT_MOTION_REQ_M	
; SPINDLE_MOTOR_VEL	
; SPINDLE_CSS_POS	
; SPINDLE_DAC_M	
; SPINDLE_ENA_M	
; ABORT_DECEL_1_M	
; ABORT_DECEL_2_M	
; ABORT_DECEL_3_M	
DES_VEL_ZERO_1_M	Axis 1 (X) desired velocity zero bit
DES_VEL_ZERO_2_M	Axis 2 (Y) desired velocity zero bit
DES_VEL_ZERO_3_M	Axis 3 (Z) desired velocity zero bit
DES_VEL_ZERO_4_M	Axis 4 (A/U) desired velocity zero bit
DES_VEL_ZERO_5_M	Axis 5 (B/V) desired velocity zero bit
POS_BIAS1_M	Axis 1 (X) position bias (G92 offset)
POS_BIAS2_M	Axis 2 (Y) position bias (G92 offset)
POS_BIAS3_M	Axis 3 (Z) position bias (G92 offset)
POS_BIAS4_M	Axis 4 (A/U) position bias (G92 offset)
POS_BIAS5_M	Axis 5 (B/V) position bias (G92 offset)
PLUS_LIMIT1_M	Axis 1 (X) plus limit active (hard or soft)
PLUS_LIMIT2_M	Axis 2 (Y) plus limit active (hard or soft)
PLUS_LIMIT3_M	Axis 3 (Z) plus limit active (hard or soft)
PLUS_LIMIT4_M	Axis 4 (A/U) plus limit active (hard or soft)
PLUS_LIMIT5_M	Axis 5 (B/V) plus limit active (hard or soft)
NEG_LIMIT1_M	Axis 1 (X) minus limit active (hard or soft)
NEG_LIMIT2_M	Axis 2 (Y) minus limit active (hard or soft)
NEG_LIMIT3_M	Axis 3 (Z) minus limit active (hard or soft)
NEG_LIMIT4_M	Axis 4 (A/U) minus limit active (hard or soft)
NEG_LIMIT5_M	Axis 5 (B/V) minus limit active (hard or soft)
MOT1_ACTUAL_POS_M	Axis 1 (X) actual position
MOT2_ACTUAL_POS_M	Axis 2 (Y) actual position
MOT3_ACTUAL_POS_M	Axis 3 (Z) actual position
MOT4_ACTUAL_POS_M	Axis 4 (A/U) actual position
MOT5_ACTUAL_POS_M	Axis 5 (B/V) actual position
CS1_PROG_FEED_M	
HOME_FLAG_1	Axis 1 (X) home flag hardware input
HOME_FLAG_2	Axis 2 (Y) home flag hardware input
HOME_FLAG_3	Axis 3 (Z) home flag hardware input
HOME_FLAG_4	Axis 4 (A/U) home flag hardware input
HOME_FLAG_5	Axis 5 (B/V) home flag hardware input
HWLIM1_PLUS	Axis 1 (X) limit plus hardware input
HWLIM1_MINUS	Axis 1 (X) limit minus hardware input
HWLIM2_PLUS	Axis 2 (Y) limit plus hardware input
HWLIM2_MINUS	Axis 2 (Y) limit minus hardware input
HWLIM3_PLUS	Axis 3 (Z) limit plus hardware input
HWLIM3_MINUS	Axis 3 (Z) limit minus hardware input
HWLIM4_PLUS	Axis 4 (A/U) limit plus hardware input
HWLIM4_MINUS	Axis 4 (A/U) limit minus hardware input
HWLIM5_PLUS	Axis 5 (B/V) limit plus hardware input
HWLIM5_MINUS	Axis 5 (B/V) limit minus hardware input
USER_FLAG_1	Axis 1 (X) User flag hardware input
USER_FLAG_2	Axis 2 (Y) User flag hardware input

USER_FLAG_3	Axis 3 (Z) User flag hardware input
USER_FLAG_4	Axis 4 (A/U) User flag hardware input
USER_FLAG_5	Axis 5 (B/V) User flag hardware input
AMP1_ENA_M	Axis 1 (X) Amplifier enable hardware output
AMP2_ENA_M	Axis 2 (Y) Amplifier enable hardware output
AMP3_ENA_M	Axis 3 (Z) Amplifier enable hardware output
AMP4_ENA_M	Axis 4 (A/U) Amplifier enable hardware output
AMP5_ENA_M	Axis 5 (B/V) Amplifier enable hardware output
FATAL_FE1_M	Axis 1 (X) Fatal following error bit
FATAL_FE2_M	Axis 2 (Y) Fatal following error bit
FATAL_FE3_M	Axis 3 (Z) Fatal following error bit
FATAL_FE4_M	Axis 4 (A/U) Fatal following error bit
FATAL_FE5_M	Axis 5 (B/V) Fatal following error bit
AMP_FAUT1_M	Axis 1 (X) Amplifier fault hardware input
AMP_FAUT2_M	Axis 2 (Y) Amplifier fault hardware input
AMP_FAUT3_M	Axis 3 (Z) Amplifier fault hardware input
AMP_FAUT4_M	Axis 4 (A/U) Amplifier fault hardware input
AMP_FAUT5_M	Axis 5 (B/V) Amplifier fault hardware input
DAC5_OUT	DAC 5 output value (when motor 5 not used)
DAC6_OUT	DAC 6 output value

```
//*************************************************************************
//P VARIABLES
//*************************************************************************
```

SYMBOL	Comment
tool_geomwear	Actual tool correction number
tool_number	Actual tool number
VERSION_NUMBER	PLC program version number
DATE_VERSION	PLC program date
HOME_COMPLETE_P	All axis Home completed bit
DES_VEL_ZERO_P	All axis Desired Velocity zero bit
USER_DISABLE_OVRD	User bit to force feed to 100% (disable feed pot.)
AMP_ARE_ENABLED_P	All axis amplifier are enabled
LATHE_MACHINE	0 = Milling machine 1 = Lathe machine
MMI_OK	HMI running
TOOL_PREVIOUS	Actual (previous) tool active when tool change called
HAVE_TOOL_CHANGER	Tool changer option checked in setting page 4
S1_IN_C_AXE	Spindle used in C axe
TOOL_IN_SPINDLE	Tool in spindle from tool changer initialisation page
MAGAZINE_POSITION	Magazine posit. from tool changer initialisation page
INCH_MODE	Metric or Inch mode

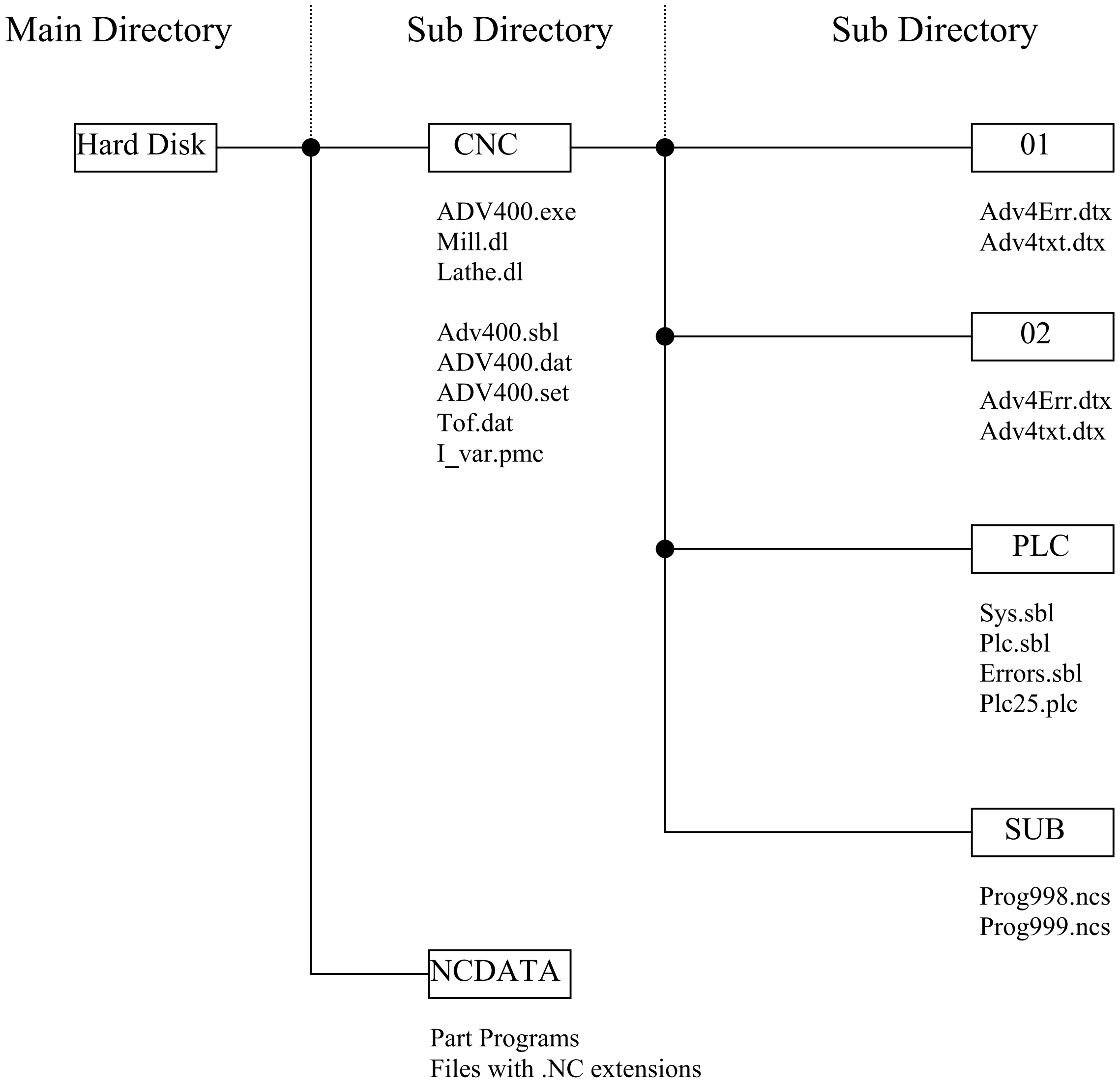
```
//*************************************************************************
//CONSTANTS
//*************************************************************************
```

SYMBOL	Comment
TRUE	Value True
FALSE	Value False
ON	Value ON
OFF	Value OFF

Annexe 2 : Hard Disk CNC architecture

This section explains the structure of directories and files in the flash memory (DOM) of the ADV400, structure needed for the ADV400-HMI software.

DISK ARCHITECTURE FOR ADV400 CNC:



The **main directory** is called **HARD DISK**. It's the root of the disk.
This main directory contains the Windows CE files.

The **sub-directory NCDATA** contains the part programs (G-codes programs) created by the user. These files are managed automatically by the HMI while creations/modifications of part programs. They have a .NC extension.

The **sub-directory CNC** contains sub-directories (01, 02, PLC and SUB) and some files needed by the HMI.

The file “Adv400.exe” is the HMI software.

Files “Mill.dl” and “Lathe.dl” are the system PLCs sent to the controller at power-up.
File “Mill.dl” if a Milling machine was selected, file “Lathe.dl” if a Lathe machine was selected in the machine Setting page 4.
All other files of this sub-directory CNC are created by the HMI software during the setting of the machine.

The **sub-directory 01** contains the error file and the text file used for first language. First language is English by default.

The **sub-directory 02** contains the error file and the text file used for second language. Second language can be any language. Translations have to be made by the machine integrator.

The **sub-directory PLC** will contain the PLC15 to 24 the machine integrator may create. A PLC25 is already existing and can be modified by the machine integrator. It is used to put extra conditions to the use of the ADV400 buttons, like Cycle Start button, JOG buttons…
This sub-directory contains also the different symbol tables (files Sys.sbl, Errors.sbl and Plc.sbl) containing all symbol names used for PLC or Sub-program development.

The **sub-directory SUB** will contain the sub-programs 2-997 the user may create.
The sub-program 998 and 999 are already present and have to be completed by the machine integrator. Program 998 is used for integration of a tool changer. Program 999 is used to create custom M-codes.