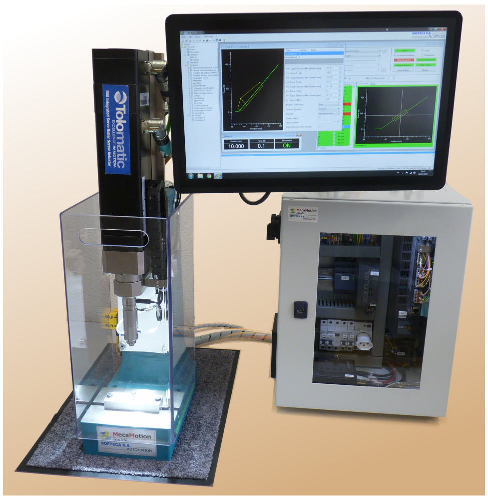
Universal press
MecaMotion

SOFTECA S.A.
AUTOMATION
ETUDE ET REALISATION DE COMMANDES INDUSTRIELLES

Champs-Montants 16 b/c
Case postale 184
2074 Marin-Epagnier
Tél:032 753 41 22
Fax : 032 753 60 56
E-Mail:softeca@bluewin.ch
http://www.softeca.ch

MecaMotion
Inside

SOFTECA S.A.
AUTOMATION



This help has been performed by:

SOFTECA S.A.
AUTOMATION
ETUDE ET REALISATION DE COMMANDES INDUSTRIELLES

Champs-Montants 16 b/c
Case postale 184
2074 Marin-Epagnier
Tél:032 753 41 22
Fax : 032 753 60 56
E-Mail:softeca@bluewin.ch
http://www.softeca.ch

MecaMotion Inside

SOFTECA S.A.
AUTOMATION

Version of the help document: Version 2.4

Document date : February 19, 2019

Date of the corresponding program : November 26, 2018

Minimum computer requirements to install MecaMotion: Windows XP, .NET framework4.0, CPU 2Ghz, RAM 2 Gb

The universal press

The universal press has been designed to perform assembling, punching, riveting, cutting, bending, stamping, marking, measuring, or clipping operations. For each operation, the press manages the force applied throughout the cycle. Thanks to this, it is also possible to carry out stiffness tests.

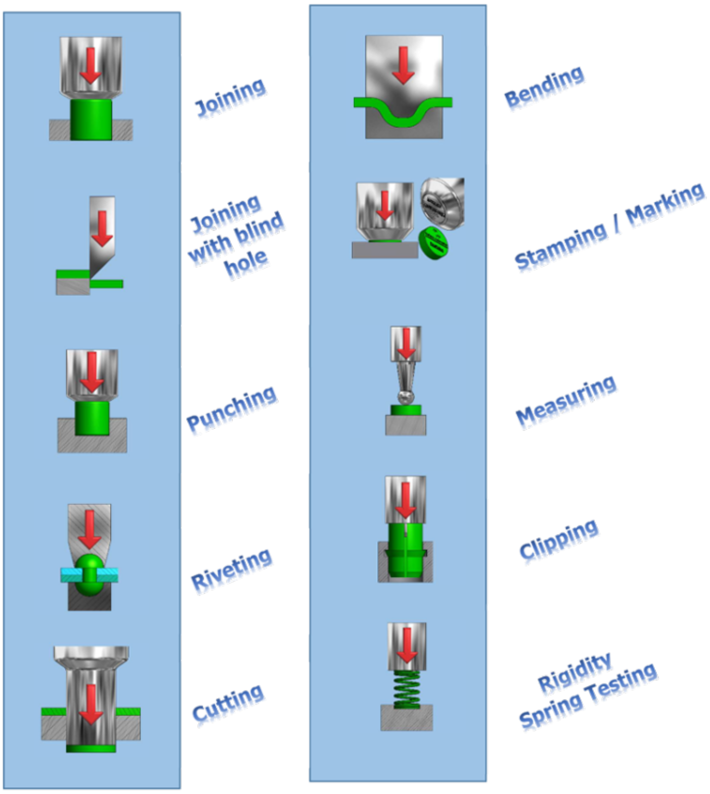


Figure 1: Example of operations achievable with the press

In order to adapt to all types of applications, the relatively intuitive MecaMotion software allows you to program the cycle to be performed yourself.

The universal press can be controlled using a Beckhoff, Siemens, B&R, Codesys, Schneider Electric PLC via a PROFINET link, or with a computer and MecaMotion software that uses the Ethernet link.

Schematic diagram of the use of a PLC controlled press

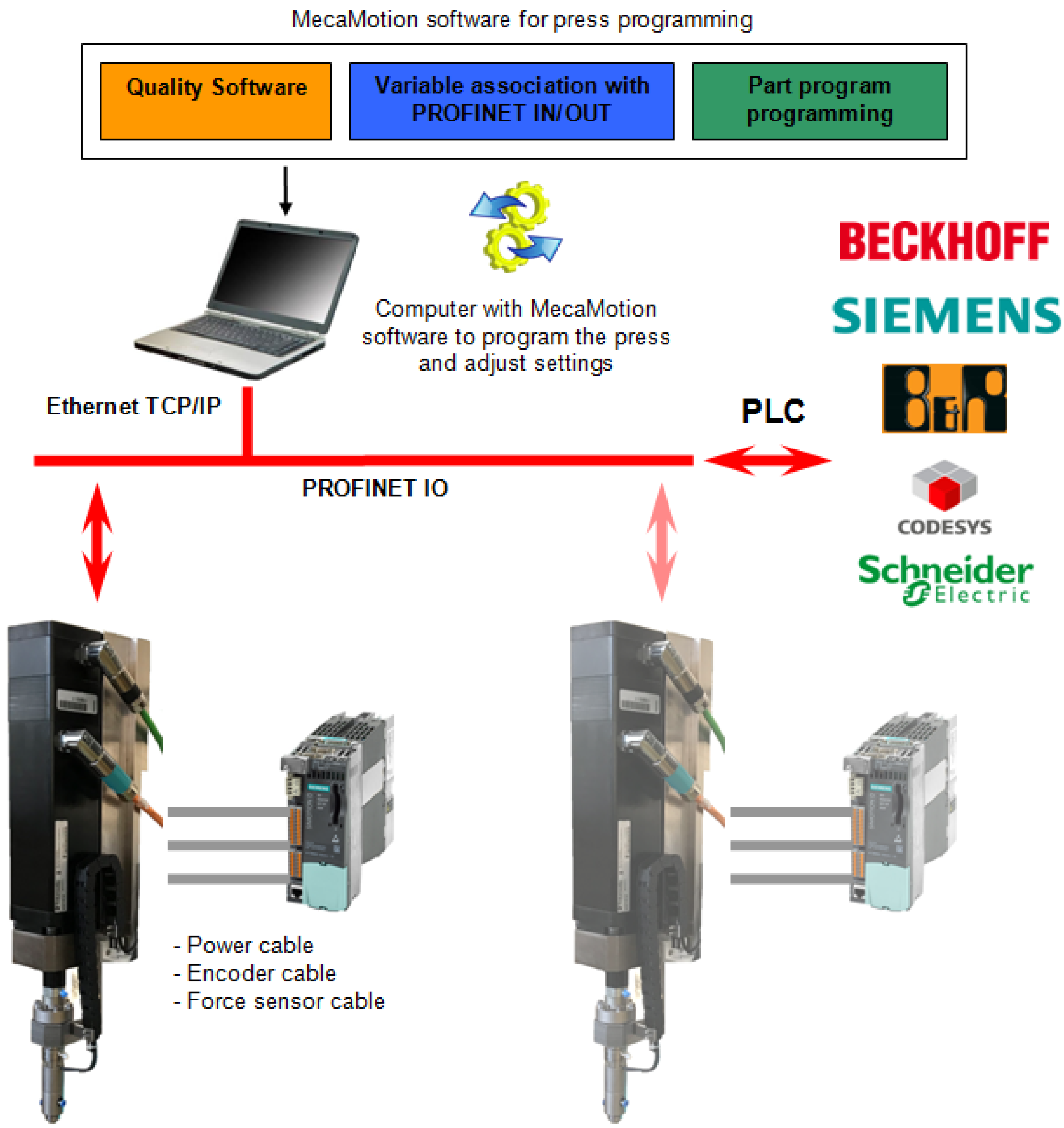


Figure 2: Schematic diagram of the use of the universal press

Overview of the universal press system

The universal press is built around a SIMOTION architecture (Siemens). This architecture is composed of the following hardware:

- D410-2 control unit specific to the axis control
- PM240-2 power part

In case the motor is not a Siemens brand and the encoder returns ENDAT signals, the following module must be added:

- Encoder feedback: SME25 encoder converter (ENDAT to Drive-CliQ)

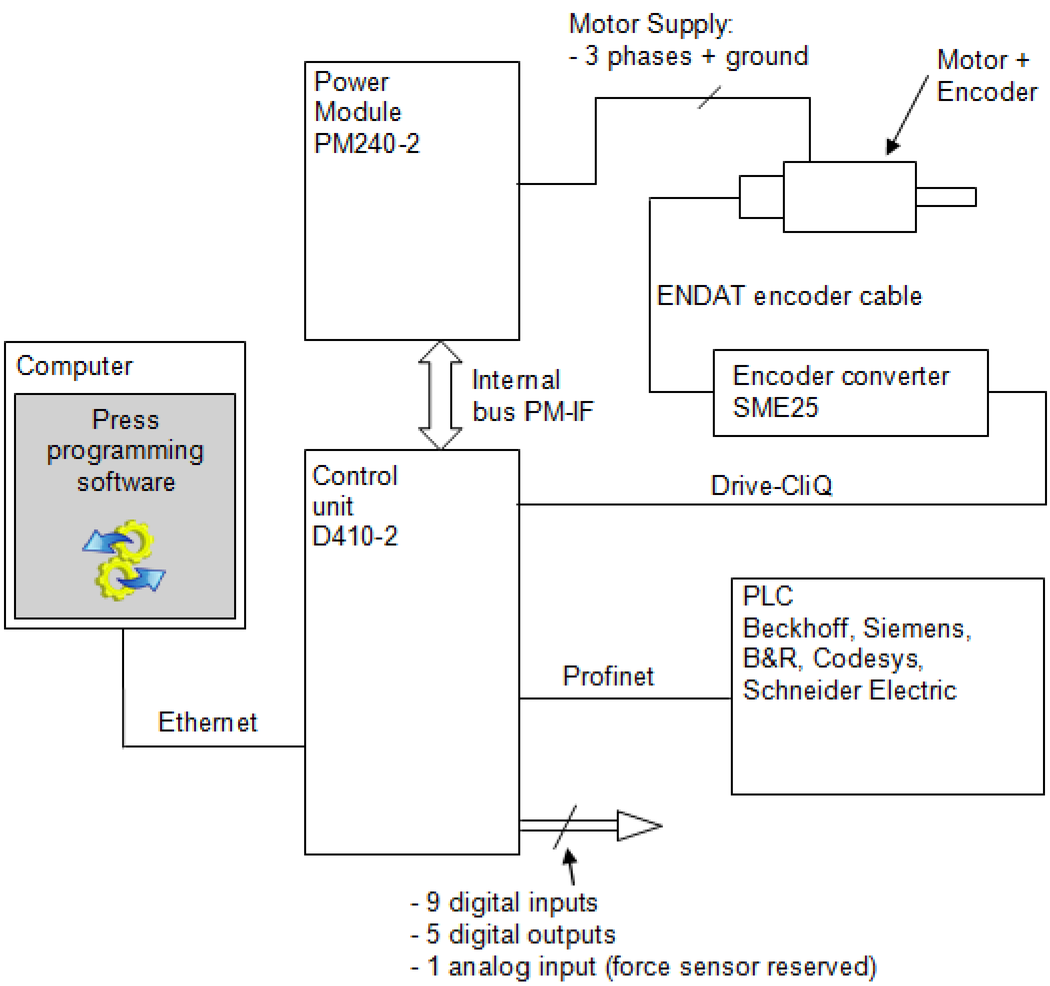


Figure 1: Block diagram of the press in case the motor is not Siemens brand

Press programming

The press is programmed by means of MecaMotion software via an Ethernet link.

Using MecaMotion software, the following operations can be performed:

- Program different cycles (up to 254 part program).
- Modify press parameters.

- Receive and visualize the curves "Force = f(Position)".
- Define the association of user variables to inputs/outputs (physical or Profinet/Profibus).
- Compare online / offline projects.
- Check the shape of the "Force = f(Position)" curve by means of an envelope.
- View the progress of the current part program as well as the values of the user variables.
- Create production orders and save the results in a database.
- Control the press manually and execute the part programs (standalone mode).

Press control by a PLC

You can control the press with MecaMotion (standalone mode) or with a PLC and in this case, the various commands that control the press must be sent via a Profinet or Profibus link. Example :

- Execute the part program.
- Choose the press control mode (manual or automatic).
- JOG go up and go down (run on sight).
- View the errors present.
- Visualize the technological data (force, position).
- ...

Hardware supplied with the press

In table 1 below, you will find all the hardware supplied with the universal press.

Component	Description	Representation
Servopress	Press motor	
PM240-2	Power module (Drive)	
D410-2	Press control unit	

Component	Description	Representation
Flash card	1GB flash card containing the firmware	
Power cable	Motor power supply cable	
Drive-CliQ cable	Drive-CliQ encoder cable	
CD-ROM	CD-ROM containing: • MecaMotion platform for press programming • GSDML file to insert the press into a PROFINET network • Block diagram of the press wiring (PDF format)	

Table 1: List of material provided with the press

Options

In the table 2, you will find the optional hardware of the universal press.

Component	Description	Representation
SME25	Encoder converter ENDAT -> Drive-CliQ for motors that do not have a Drive-CliQ connection	
Encoder cable	Cable for encoder type ENDAT	
Optical ruler	Optical ruler for linear measurement of the axis position	
SMC40	Module for processing external encoders ENDAT 2.2 for optical ruler	

Table 2: List of optional hardware

Mounting and wiring of electrical hardware

Mounting the D410-2 control unit and the PM240-2 power module

The D410-2 control unit is mounted directly on the PM240-2 power module, as shown in figure 1.

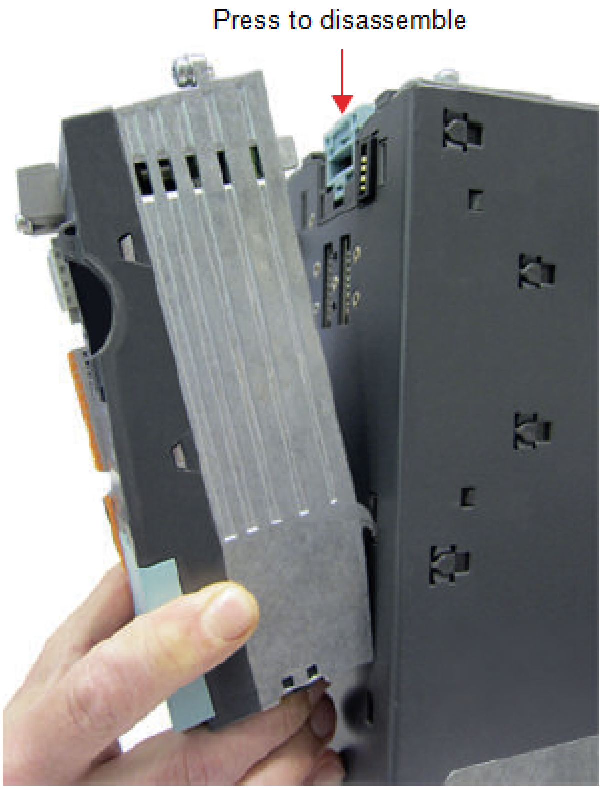


Figure 1: Mounting of D410-2 on the PM240-2

The PM240-2 power section is fixed directly to the mounting plate of the electrical cabinet using three M4 screws. The system shielding kit is attached to the PM240-2. (see figure 2 below)

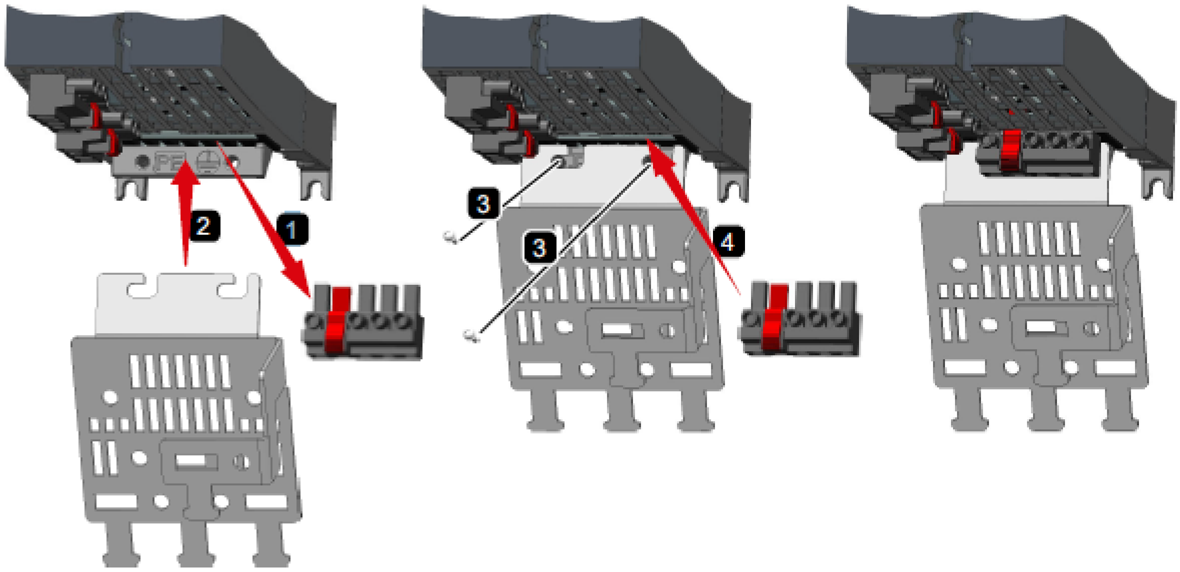


Figure 2: Mounting the shielding kit on PM240-2

The dimensions of the power module are given in table 1 and illustrated in figure 3.

Frame size	Width [mm]	Height without shield plate [mm]	Height with shield plate [mm]	Depth without D410-2 [mm]	Depth with D410-2 [mm]
FSA	73	196	276	165	240
FSB	100	292	370	165	240
FSC	140	355	432	165	240

Table 1: PM240-2 dimensions

The drilling dimensions of the power module are given in table 2 and illustrated in figure 3.

Frame size	Drilling dimensions [mm]		
	h	b	c
FSA	186	62.3	6
FSB	281	80	6
FSC	343	120	6

Table 2: Drilling dimension

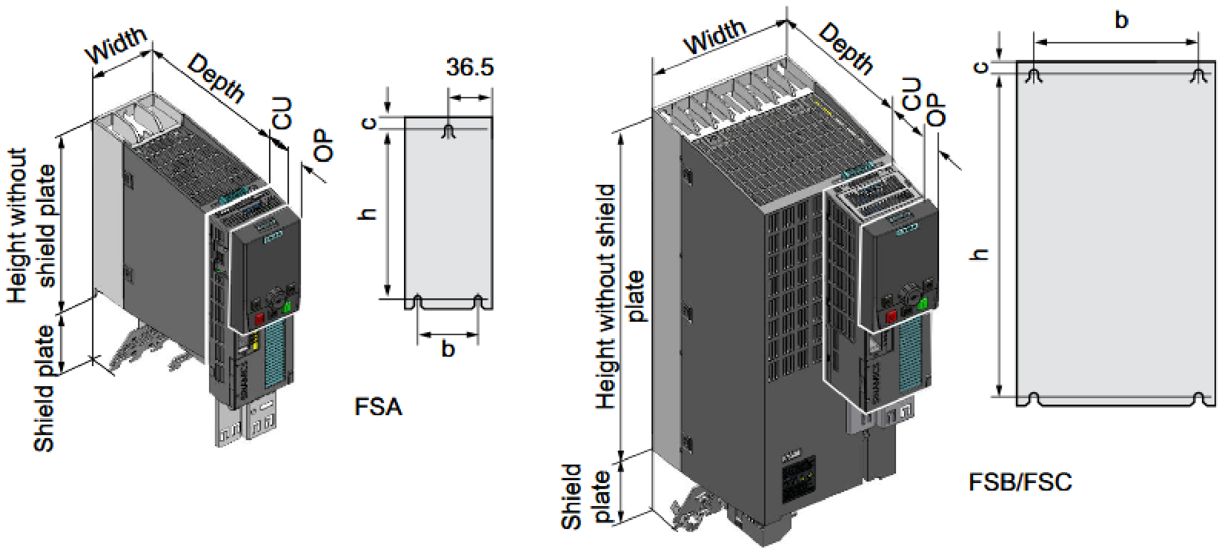


Figure 3: Power module dimensions and drilling dimensions

Be careful, the power module should only be installed in the vertical position with the motor connectors at the bottom.

For the cooling of the power module to be carried out correctly, it is necessary to provide air clearance at the top, bottom and front.

Frame size	Cooling air clearances [mm]		
	Top	Bottom	Front
FSA	80	100	100
FSB	80	100	100
FSC	80	100	100

Table 3: Cooling air clearances

Electrical wiring

For wiring the PM240-2 power module and the D410-2 control unit, refer to the document "Universal presse block diagram"

Creating a Profinet link

As explained in the topic "Overview of the universal press system", the press can be controlled by a PLC, using a PROFINET connection.

Below is a brief list of existing PROFINET commands:

- Selection of the operating mode (Manual or Automatic)
- Request for go up or go down in JOG (run on sight)
- Manual movement (position, abs/rel. mode, speed, acc./dec. and start manual positioning)
- Sending the part program number to be activated
- Executing the active part-program
- ...

For more information on the different PROFINET commands, see the help topic "Control the press by a PLC".

Wiring a PROFINET connection between the press and a PLC

We will take as an example a Siemens PLC from the S7-1500 range.

To be able to communicate with the press via PROFINET, you must make the connection shown in Figure 1.

The connection to the PC on which the MecaMotion software is located is shown as a broken line and the PROFINET connection to the PLC is shown as a continuous line.

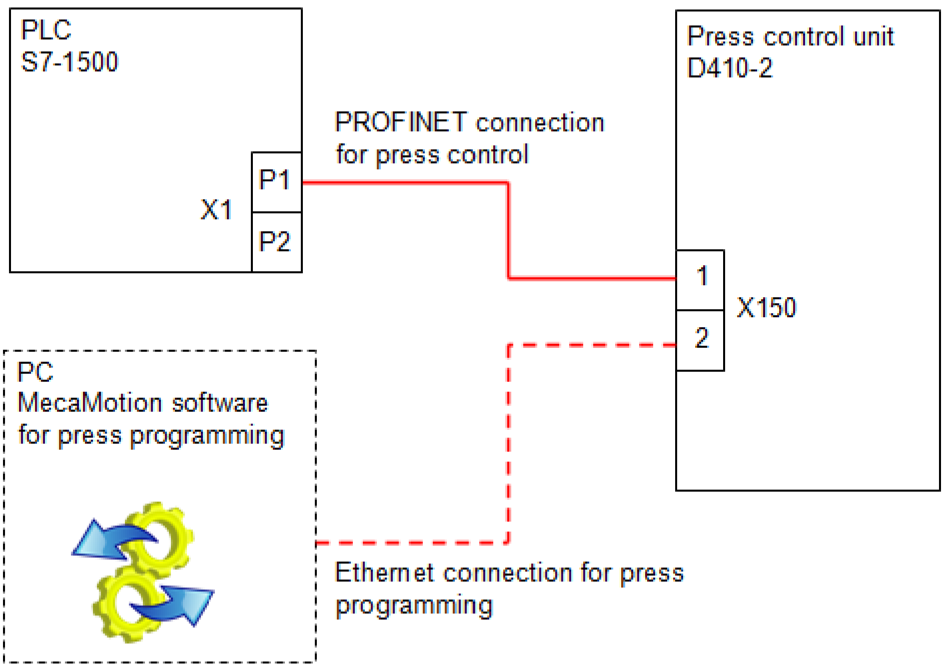


Figure 1: Diagram of the connections with the PLC and PC

Creation of the PROFINET connection in the hardware configuration of the PLC

The example below is made with Siemens TIA PORTAL software.

First, in the programming software, you must create a new project and add the PLC you are going to use (example S7-1500).

Then, install the GSD file provided with the press, to do so, click on the "Options" tab and then "manage general station description files".

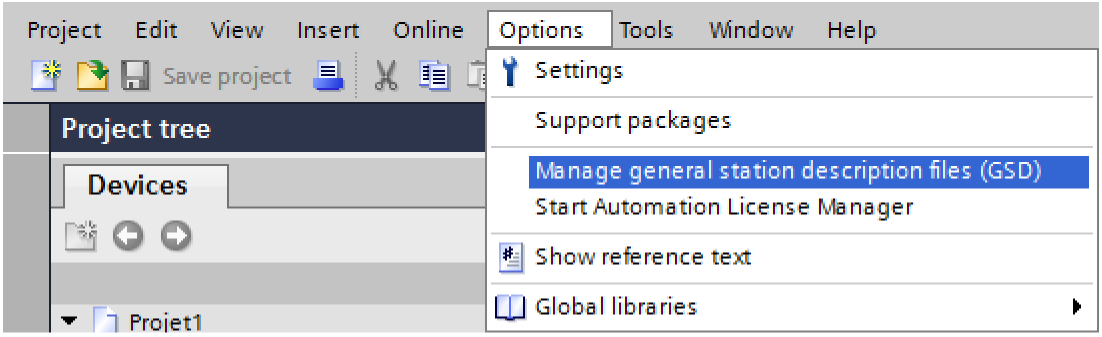


Figure 2: Opening the GSD file manager

Choose the source path and install the GSD file.

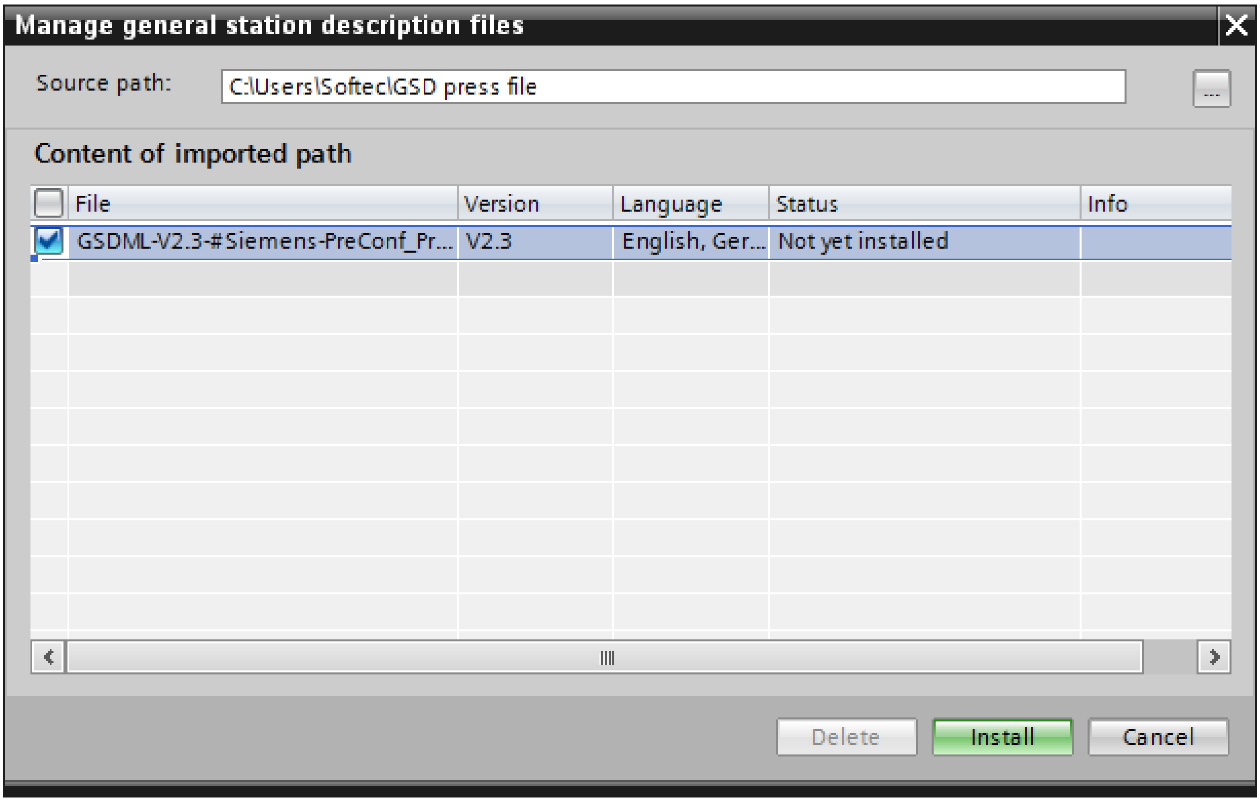


Figure 3: Installing the GSD press file

Once the GSD file is installed, you will find the press element in the hardware catalogue on the right side of the screen. The path is as follows: Other field devices -> PROFINET IO -> PLCs & Cps -> SIEMENS AG -> D410 -> Presse.

Drag and drop the press element into the network view.

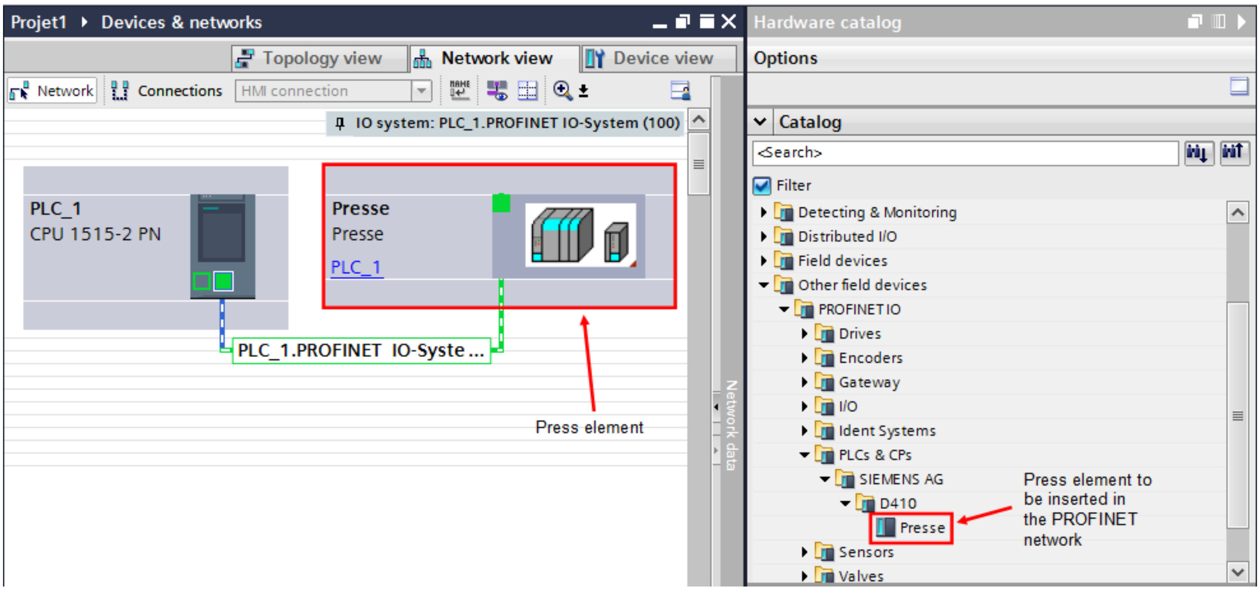


Figure 4: Inserting the press element into the network view

On the press element, click on "not assigned" and choose the PLC with which you want to create the PROFINET link.

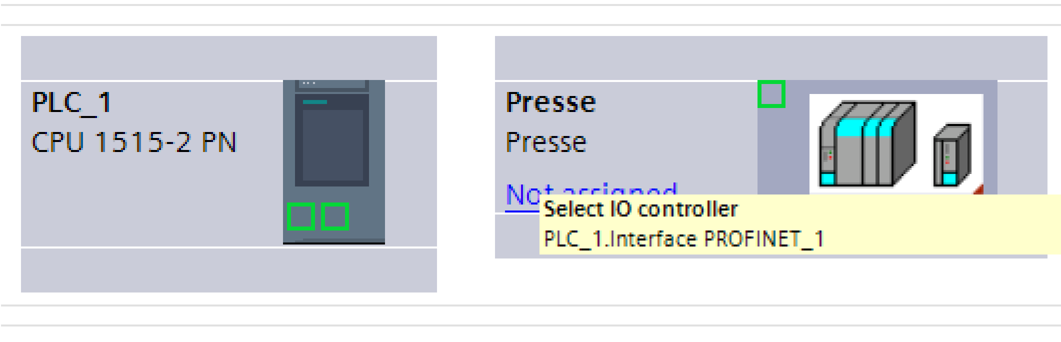


Figure 5: Assignment to the PROFINET network

The PROFINET link is now created, all you have to do is enter the IP addresses and subnetwork masks of the devices. You will also need to define the address ranges of the Profinet inputs/outputs of the press.

Encoder recognition

If the press motor is a **third party motor (not Siemens)** and the **encoder is of the EnDat type**, to be able to use the press, you must make a recognition of the encoder.

Procedure to follow for recognizing the encoder

1. **First of all, the emergency stop must be activated before the encoder can be recognized.** Then, in the project tree, right-click on the "Press" tab and, in the drop-down menu that appears, select "Expert Control". Once this menu is selected, the window in Figure 1 opens.

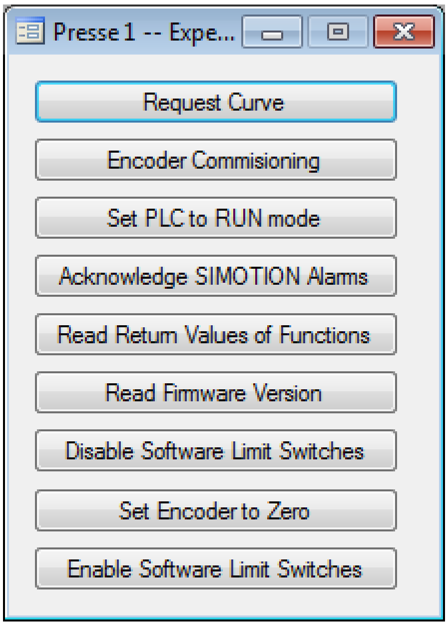


Figure 1: "Expert" control menu

2. Before proceeding, make sure that the IP address entered in the platform is correct. (See "Establishing the Ethernet connection").
3. When the IP address of the press is correct, click on the "**Encoder Commissioning**" button.

Thenceforth, the encoder recognition will be done automatically. Wait until the text "**Successful ROM writing - Learning completed**" appears in the event window.

Zeroing of the encoder

Before making any movement, you must set the encoder to zero. Without this, as the value of the encoder is not known, there is a **risk of collision**.

To perform this zeroing, if you are not in standalone mode, the press must be connected to a PLC in order to be able to use run on sight mode (JOG mode).

Procedure to follow for zeroing the encoder

1. In the project tree of the MecaMotion software, right-click on the "Press" tab, in the drop-down menu that appears, select "Expert control". The window in Figure 1 opens.

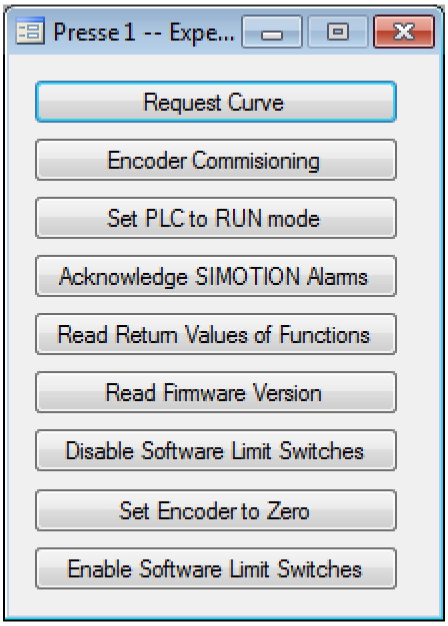


Figure 1: "Expert" control menu

2. Once this menu is open, the first step to perform to zero the encoder is to disable the software limit switches. To do this, click on the "**Disable software limit switches**" button.
3. As soon as the software limit switches have been deactivated, it is possible to manipulate the axis using the "JOG+" and "JOG-" commands. These commands are sent to the press by the PLC, via the PROFINET bus or from the manual mode of the control page, if you work in standalone mode.
4. Using the commands JOG+ and JOG-, position the axis to "0 mechanical".
5. When the axis is at the position you have decided as the "Zero" position, you must learn this position. To do this, in the "**Expert control**" window, click on the "**set encoder to zero**" button.
6. When the zeroing is done, all that remains is to reactivate the software limit switches. To do this, in the "**Expert control**" window, click on the "**Enable software limit switches**" button.

Note: If part programs have been created with another "0 mechanical", the position values will no longer be valid.

/!Beware of the risk of breakage /!

Press safety

To stop the press when the user requests an emergency stop, we use the safe stop 1 (SS1) function, which brakes the motor before turning off the power (STO function), to prevent the press from continuing its freewheeling travel.

When the press receives a stop request on the DI16 input, the motor is braked autonomously following a fast stop ramp and when the motor is stopped, the power supplied to the motor is switched off. It is then necessary to switch off the pulses using the DI17 input and switch off the contactor that supplies the motor for a safe stop of the motor.

The emergency press stop cycle is explained below. (figure 1)

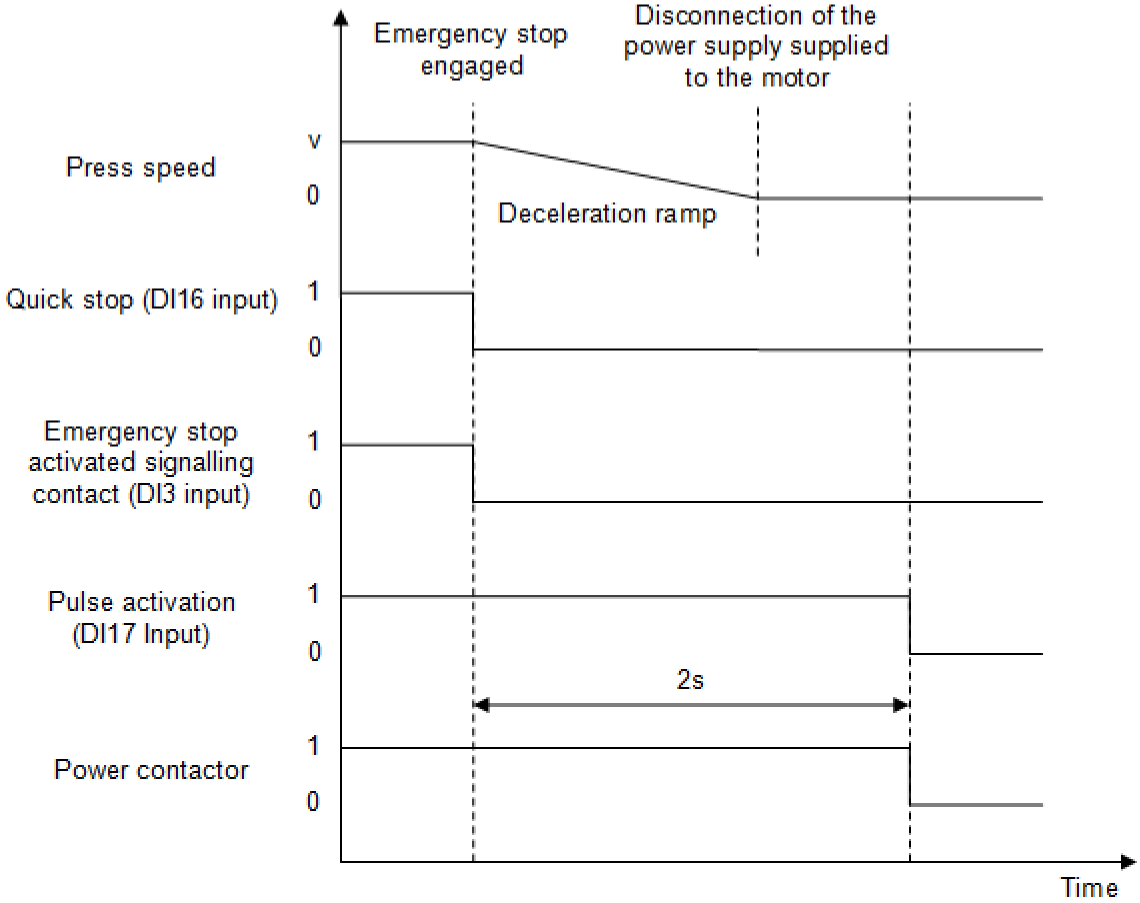


Figure 1: Chronogram of the press emergency stop cycle

Overview of the PROFINET dialogue

The press is connected to a programmable logic controller (PLC) via a PROFINET connection. From this channel, it is possible to send data to the press and receive data from the press.

On this channel, there are 254 input bytes and 254 output bytes available.

For the following examples, we will take the starting address 0, for inputs and outputs.

- PROFINET input addresses: 0...253
- PROFINET output addresses: 0...253

These addresses are relative to the starting address that the integrator will have given to the press. In the case of a system with several presses, the addresses must be different for each press.

PROFINET inputs

The 254 input bytes are distributed as follows, on the PROFINET dialog:

- Bytes 0...199; used as DWORD variables (4bytes = 1 variable => 50 DWORD variables). These 50 variables can be used to transfer "REAL", "DINT" or "LREAL" type data from the PLC to the press, via the association of user variables. (Variables of type LREAL must be sent in REAL format from the PLC)
- Bytes 200...203; used as BOOL variables (4 bytes = 32 bits). These 32 Boolean variables can be used to transfer flag from the PLC to the press via the association of variables.
- Bytes 204...223; 5 variables of type DWORD (1 variable = 4 bytes) reserved for the connection of a sensor or other instrument connected by PROFINET.
- Bytes 224... 253; These bytes are reserved for press commands (For more details on these different commands, see the topic "Control the press by a PLC"). In these commands are included among other things:
  - Start part program
  - Mode change (manual, automatic)
  - JOG + and -
  - Change of programs
  - Change of envelopes
  - Acknowledgement of errors
  - ...

Addresses from 0	Designation
Bytes 0...199	50 variables of type DWORD used to transfer REAL, DINT or LREAL type data from the PLC to the press, via the association of user variables.
Bytes 200...203	32 BOOL variables used to transfer data from the PLC to the press via the association of user variables.
Bytes 204...223	5 variables of type DWORD reserved for the connection of sensors or other instruments connected by PROFINET.
Bytes 224...253	Reserved addresses to control the press

Table 1: PROFINET inputs structure

PROFINET outputs

The 254 bytes of outputs are distributed as follows, on the PROFINET channel:

- Bytes 0...199; used as DWORD variables (4bytes = 1 variable => 50 DWORD variables). These 50 variables can be used to transfer "REAL", "DINT" or "LREAL" type data from the press to the PLC via the association of user variables. (LREAL variables are received in REAL format in the PLC)
- Bytes 200...203; used as BOOL variables (4 bytes = 32 bits). These 32 Boolean variables can be used to transfer flag from the press to the PLC via the association of variables.
- Bytes 204...223; 5 variables of type DWORD (1 variable = 4 bytes), reserved for the connection of a sensor or other instrument connected by PROFINET.
- Bytes 224...245 and 251...253; These bytes are reserved for the return of the press commands (For more details on these different returns, see the topic "Control the press by a PLC"). In these datas are included among other things:
    - Actual value of the axis, force
    - Actual mode
    - Active program number
    - Active envelope number
    - Error present
    - ...
- Bytes 246...250; 5 bytes reserved for all errors that the press can return. Each bit of these 5 bytes corresponds to a specific error => 40 possible errors.

Addresses from 0	Designation
Bytes 0...199	50 variables of type DWORD used to transfer REAL, DINT or LREAL type data from the press to the PLC, via the association of user variables.
Bytes 200...203	32 BOOL variables used to transfer data from the press to the PLC via the association of user variables.

Addresses from 0	Designation
Bytes 204…223	5 variables of type DWORD reserved for the connection of instruments connected by PROFINET.
Bytes 224…245	Reserved addresses for the return of press controls
Bytes 246…250	5 bytes reserved for all errors that the press can return
Bytes 251…253	Reserved addresses for the return of press controls

Table 2: PROFINET outputs structure

List of PROFINET variables

DWORD variables

As explained in the document "Overview of the PROFINET dialogue",bytes 0 to 199 are used as DWORD variables.

Since a DWORD type variable consists of 4 bytes, these first 200 bytes of the Profinet dialog actually correspond to 50 DWORD type variables. These variables are numbered from "0" to "49".

The address of each of these variables corresponds to the first byte of the latter.

In table 1 below, you will find a list of the press input variables and their addresses:

Variable number	PROFINET address	Byte number of the variable	Direction (from the press)	Format
0	0	0, 1, 2, 3	IN	DWORD
1	4	4, 5, 6, 7	IN	DWORD
2	8	8, 9, 10, 11	IN	DWORD
3	12	12, 13, 14, 15	IN	DWORD
4	16	16, 17, 18, 19	IN	DWORD
5	20	20, 21, 22, 23	IN	DWORD
6	24	24, 25, 26, 27	IN	DWORD
7	28	28, 29, 30, 31	IN	DWORD
8	32	32, 33, 34, 35	IN	DWORD
9	36	36, 37, 38, 39	IN	DWORD
10	40	40, 41, 42, 43	IN	DWORD
11	44	44, 45, 46, 47	IN	DWORD
12	48	48, 49, 50, 51	IN	DWORD
13	52	52, 53, 54, 55	IN	DWORD
14	56	56, 57, 58, 59	IN	DWORD
15	60	60, 61, 62, 63	IN	DWORD
16	64	64, 65, 66, 67	IN	DWORD
17	68	68, 69, 70, 71	IN	DWORD
18	72	72, 73, 74, 75	IN	DWORD
19	76	76, 77, 78, 79	IN	DWORD
20	80	80, 81, 82, 83	IN	DWORD
21	84	84, 85, 86, 87	IN	DWORD
22	88	88, 89, 90, 91	IN	DWORD
23	92	92, 93, 94, 95	IN	DWORD
24	96	96, 97, 98, 99	IN	DWORD

Variable number	PROFINET address	Byte number of the variable	Direction (from the press)	Format
25	100	100, 101, 102, 103	IN	DWORD
26	104	104, 105, 106, 107	IN	DWORD
27	108	108, 109, 110, 111	IN	DWORD
28	112	112, 113,114, 115	IN	DWORD
29	116	116, 117, 118, 119	IN	DWORD
30	120	120, 121, 122, 123	IN	DWORD
31	124	124, 125, 126, 127	IN	DWORD
32	128	128, 129, 130, 131	IN	DWORD
33	132	132, 133, 134, 135	IN	DWORD
34	136	136, 137, 138, 139	IN	DWORD
35	140	140, 141, 142, 143	IN	DWORD
36	144	144, 145, 146, 147	IN	DWORD
37	148	148, 149, 150, 151	IN	DWORD
38	152	152, 153, 154, 155	IN	DWORD
39	156	156, 157, 158, 159	IN	DWORD
40	160	160, 161, 162, 163	IN	DWORD
41	164	164, 165, 166, 167	IN	DWORD
42	168	168, 169, 170, 171	IN	DWORD
43	172	172, 173, 174, 175	IN	DWORD
44	176	176, 177, 178, 179	IN	DWORD
45	180	180, 181, 182, 183	IN	DWORD
46	184	184, 185, 186, 187	IN	DWORD
47	188	188, 189, 190, 191	IN	DWORD
48	192	192, 193, 194, 195	IN	DWORD
49	196	196, 197, 198, 199	IN	DWORD

Table 1: Address list of press PROFINET input variables

In table 2 below, you will find a list of the press output variables and their addresses:

Variable number	PROFINET address	Byte number of the variable	Direction (from the press)	Format
0	0	0, 1, 2, 3	OUT	DWORD
1	4	4, 5, 6, 7	OUT	DWORD
2	8	8, 9, 10, 11	OUT	DWORD
3	12	12, 13, 14, 15	OUT	DWORD
4	16	16, 17, 18, 19	OUT	DWORD

Variable number	PROFINET address	Byte number of the variable	Direction (from the press)	Format
5	20	20, 21, 22, 23	OUT	DWORD
6	24	24, 25, 26, 27	OUT	DWORD
7	28	28, 29, 30, 31	OUT	DWORD
8	32	32, 33, 34, 35	OUT	DWORD
9	36	36, 37, 38, 39	OUT	DWORD
10	40	40, 41, 42, 43	OUT	DWORD
11	44	44, 45, 46, 47	OUT	DWORD
12	48	48, 49, 50, 51	OUT	DWORD
13	52	52, 53, 54, 55	OUT	DWORD
14	56	56, 57, 58, 59	OUT	DWORD
15	60	60, 61, 62, 63	OUT	DWORD
16	64	64, 65, 66, 67	OUT	DWORD
17	68	68, 69, 70, 71	OUT	DWORD
18	72	72, 73, 74, 75	OUT	DWORD
19	76	76, 77, 78, 79	OUT	DWORD
20	80	80, 81, 82, 83	OUT	DWORD
21	84	84, 85, 86, 87	OUT	DWORD
22	88	88, 89, 90, 91	OUT	DWORD
23	92	92, 93, 94, 95	OUT	DWORD
24	96	96, 97, 98, 99	OUT	DWORD
25	100	100, 101, 102, 103	OUT	DWORD
26	104	104, 105, 106, 107	OUT	DWORD
27	108	108, 109, 110, 111	OUT	DWORD
28	112	112, 113,114, 115	OUT	DWORD
29	116	116, 117, 118, 119	OUT	DWORD
30	120	120, 121, 122, 123	OUT	DWORD
31	124	124, 125, 126, 127	OUT	DWORD
32	128	128, 129, 130, 131	OUT	DWORD
33	132	132, 133, 134, 135	OUT	DWORD
34	136	136, 137, 138, 139	OUT	DWORD
35	140	140, 141, 142, 143	OUT	DWORD
36	144	144, 145, 146, 147	OUT	DWORD
37	148	148, 149, 150, 151	OUT	DWORD
38	152	152, 153, 154, 155	OUT	DWORD
39	156	156, 157, 158, 159	OUT	DWORD

Variable number	PROFINET address	Byte number of the variable	Direction (from the press)	Format
40	160	160, 161, 162, 163	OUT	DWORD
41	164	164, 165, 166, 167	OUT	DWORD
42	168	168, 169, 170, 171	OUT	DWORD
43	172	172, 173, 174, 175	OUT	DWORD
44	176	176, 177, 178, 179	OUT	DWORD
45	180	180, 181, 182, 183	OUT	DWORD
46	184	184, 185, 186, 187	OUT	DWORD
47	188	188, 189, 190, 191	OUT	DWORD
48	192	192, 193, 194, 195	OUT	DWORD
49	196	196, 197, 198, 199	OUT	DWORD

Table 2: Address list of press PROFINET output variables

BOOL variables

The bytes 200 to 203 of the PROFINET dialog, are used as a Boolean variable. Indeed, since each byte is composed of 8 bits, there are a total of **32 input Boolean variables and 32 output Boolean variables** that can be used.

These 32 variables are numbered from "0" to "31".

In table 3 below, you will find a list of the Boolean input variables and their respective addresses:

Variable number	PROFINET address	Direction (from the press)	Format
0	200.0	IN	BOOL
1	200.1	IN	BOOL
2	200.2	IN	BOOL
3	200.3	IN	BOOL
4	200.4	IN	BOOL
5	200.5	IN	BOOL
6	200.6	IN	BOOL
7	200.7	IN	BOOL
8	201.0	IN	BOOL
9	201.1	IN	BOOL
10	201.2	IN	BOOL
11	201.3	IN	BOOL
12	201.4	IN	BOOL
13	201.5	IN	BOOL

Variable number	PROFINET address	Direction (from the press)	Format
14	201.6	IN	BOOL
15	201.7	IN	BOOL
16	202.0	IN	BOOL
17	202.1	IN	BOOL
18	202.2	IN	BOOL
19	202.3	IN	BOOL
20	202.4	IN	BOOL
21	202.5	IN	BOOL
22	202.6	IN	BOOL
23	202.7	IN	BOOL
24	203.0	IN	BOOL
25	203.1	IN	BOOL
26	203.2	IN	BOOL
27	200.3	IN	BOOL
28	203.4	IN	BOOL
29	203.5	IN	BOOL
30	203.6	IN	BOOL
31	203.7	IN	BOOL

Table 3: Address list of Boolean press input variables

In table 4 below, you will find a list of the output Boolean variables and their respective addresses:

Variable number	PROFINET address	Sens (depuis la presse)	Format
0	200.0	OUT	BOOL
1	200.1	OUT	BOOL
2	200.2	OUT	BOOL
3	200.3	OUT	BOOL
4	200.4	OUT	BOOL
5	200.5	OUT	BOOL
6	200.6	OUT	BOOL
7	200.7	OUT	BOOL
8	201.0	OUT	BOOL
9	201.1	OUT	BOOL
10	201.2	OUT	BOOL
11	201.3	OUT	BOOL

Variable number	PROFINET address	Sens (depuis la presse)	Format
12	201.4	OUT	BOOL
13	201.5	OUT	BOOL
14	201.6	OUT	BOOL
15	201.7	OUT	BOOL
16	202.0	OUT	BOOL
17	202.1	OUT	BOOL
18	202.2	OUT	BOOL
19	202.3	OUT	BOOL
20	202.4	OUT	BOOL
21	202.5	OUT	BOOL
22	202.6	OUT	BOOL
23	202.7	OUT	BOOL
24	203.0	OUT	BOOL
25	203.1	OUT	BOOL
26	203.2	OUT	BOOL
27	200.3	OUT	BOOL
28	203.4	OUT	BOOL
29	203.5	OUT	BOOL
30	203.6	OUT	BOOL
31	203.7	OUT	BOOL

Table 4: Address list of Boolean press output variables

Control the press by a PLC

The press can be controlled by a PLC via a PROFINET connection.

Be careful, you must first set the "communication" parameter to Profinet in MecaMotion.

Press PROFINET inputs

In the table below, you will find all the commands that the press can receive via the PROFINET link.

N°	Direction (from the press)	Description	PROFINET address	Format
1	IN	Program number to be activated	224	BYTE
2	IN	Operating mode to be activated	225	BYTE
3	IN	Position setpoint for manual positioning[mm]	226	REAL
4	IN	Speed setpoint for manual positioning [mm/s]	230	REAL
5	IN	Acc./Dec. setpoint for manual positioning [mm/s2]	234	REAL
6	IN	Range of the force sensor to be activated	238	BYTE
7	IN	Start the actual program	239.0	BIT
8	IN	Going down in JOG (run on sight)	239.1	BIT
9	IN	Going up in JOG (run on sight)	239.2	BIT
10	IN	Manual positioning mode, absolute or relative (absolute = 0)	239.3	BIT
11	IN	Start manual positioning	239.4	BIT
12	IN	Reset the force sensor	239.5	BIT
13	IN	Acknowledge present errors	239.6	BIT
14	IN	Change the operating mode	239.7	BIT
15	IN		240.0	BIT
16	IN	Validate the part program number	240.1	BIT
17	IN	Enable breakpoints in automatic mode (if = 0, breakpoints not used)	240.2	BIT
18	IN	Continue the program execution after breakpoint	240.3	BIT
19	IN	Go to release position	240.4	BIT
20	IN	Go to initial position	240.5	BIT
21	IN	Validate envelope (number + decoding)	240.6	BIT
22	IN	Enable envelope (active =1 inactive =0)	240.7	BIT
23	IN	Envelope number to be activated (0 = no envelope)	241	BYTE
24	IN	Stop axis movement	242.0	BIT

N°	Direction (from the press)	Description	PROFINET address	Format
25	IN	Continue axis movement	242.1	BIT

Table 1: PROFINET commands

PROFINET press outputs

In the table below, you will find all the information that the press can send back to the PLC via the PROFINET link

N°	Direction (from the press)	Description	PROFINET address	Format
1	OUT	Actual program number	224	BYTE
2	OUT	Actual press mode	225	BYTE
3	OUT	Actual range of the force sensor	226	BYTE
4	OUT		227	BYTE
5	OUT	Actual position[mm]	228	REAL
6	OUT	Actual speed [mm/s]	232	REAL
7	OUT	Actual measuring sensor value (option) [mm]	236	REAL
8	OUT	Actual value of the force sensor [N]	240	REAL
9	OUT	Actual program running	244.0	BIT
10	OUT	Program finished	244.1	BIT
11	OUT		244.2	BIT
12	OUT	Actual manual positioning mode, absolute or relative (absolute = 0)	244.3	BIT
13	OUT	Manual position reached	244.4	BIT
14	OUT	Error present	244.5	BIT
15	OUT	Initial position reached	244.6	BIT
16	OUT	Program stopped on breakpoint	244.7	BIT
17	OUT	Axis enabled (unlocked)	245.0	BIT
18	OUT	Presse cpu started (power up)	245.1	BIT
19	OUT	Stopwatch 1 in progress	245.2	BIT
20	OUT	Stopwatch 2 in progress	245.3	BIT
21	OUT	Stopwatch 3 in progress	245.4	BIT
22	OUT	Stopwatch 4 in progress	245.5	BIT
23	OUT	Stopwatch 5 in progress	245.6	BIT
24	OUT		245.7	BIT
25	OUT	Errors see "PROFINET error list",	246.0 à 250.7	BIT

N°	Direction (from the press)	Description	PROFINET address	Format
26	OUT	Actual envelope number	251	BYTE
27	OUT	Release position reached	252.0	BIT

Table 2: Informations returned by PROFINET

Explanation and examples to use the different PROFINET input/output signals with the press

Power on press

When the press is switched on, you must wait until output bit n°245.1 (press cpu started) is at "1" to send commands or read the information.

Acknowledgement of errors

To acknowledge the errors, you must set input bit n°239.6 for 50 ms.

Then, you can check that no errors are present using output bit n°244.5. (if they are an error present, the bit is "1")

Changing the operating mode

There are 2 different operating modes. To change mode, byte n°225 ("**Operating mode to be activated**") must take one of the values below:

- Byte n°225 = 16#01:

  Value to past the press in "manual mode". In this mode, it is possible to control the press in run on sight (JOG +/-).It is also possible to perform manual positioning by giving a relative or absolute position setpoint.

- Byte N°225 = 16#02:

  Value to past the press into "automatic mode". This mode allows you to execute the various part-programs.

To change the press mode, you must send one of the values described above in input byte 225, wait 50 ms, then set input bit 239.7 to "1" for 50 ms to confirm the mode change. You can then see if the actuel press mode has been changed using the output byte n°225. (see figure 1 below)

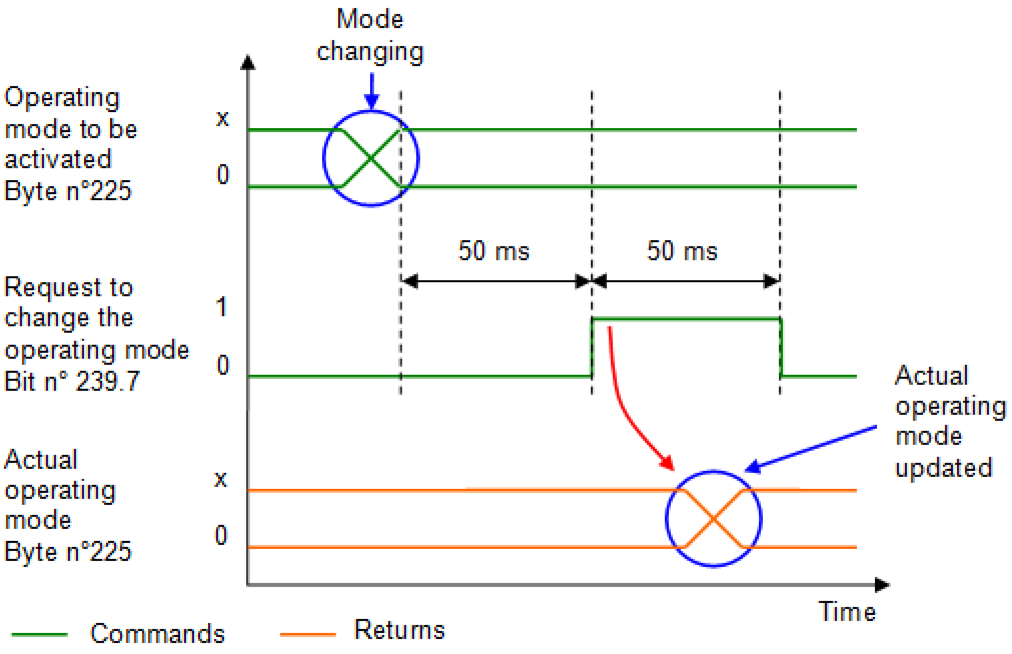


Figure 1: Chronogram of mode changing

Manual positioning

As explained above, in order to be able to perform manual positioning, the press must be in manual mode.

First, you must update the "positioning" data above:

- Position [mm] (REAL input n°226)
- Velocity [mm/s] (REAL input n°230)
- Acceleration / deceleration [mm/s2] (REAL input n°234)
- Positioning mode "Absolute" (bit input n°239.3 = "0") or "Relative" (bit input n°239.3 = "1").

These parameters will be taken into account by the press when it receives the **" Start manual positioning"** signal (Bit n°239.4).

As soon as the positioning is completed, output bit n°244.4 (manual position reached) will be set to "1". You can then reset the bit "start manual positioning"

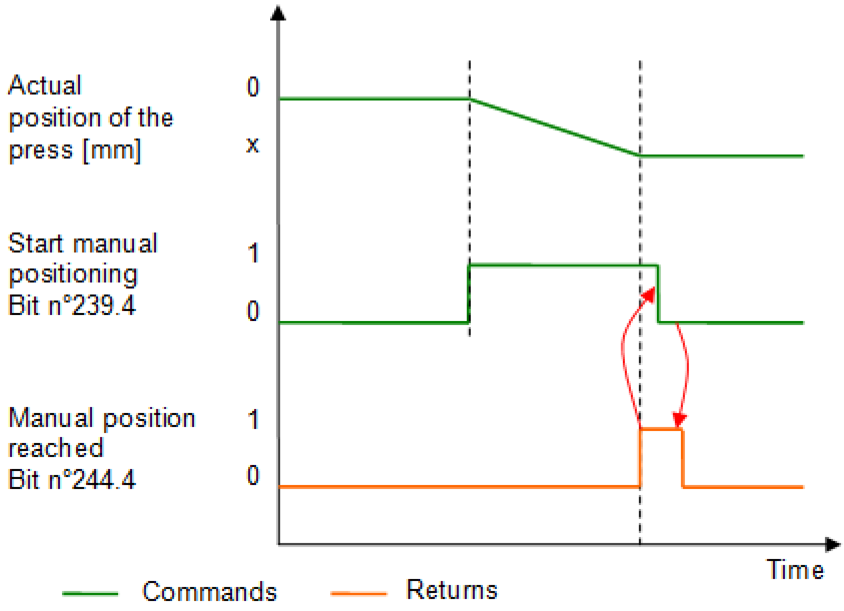


Figure 2: Starting manual positioning

Run on sight (JOG)

To control the press in run on sight (JOG mode), first, it is necessary to activate the manual mode and enter a speed in [mm/s] in output Real n°230.

Bit n°239.1 (JOG+) allows to go down and bit n°239.2 (JOG-) allows to go up.

These bits must be held at "1" for movement to occur, when they past to "0" the press stops.

Go to the initial position

To go to the initial position, you must set bit n°240.5. The press will then move and when the initial position is reached, the output bit n°244.6 will set to "1", you can then reset the control bit n°240.5. (see figure 3).

The initial position of the press is a default parameter to be entered in MécaMotion.

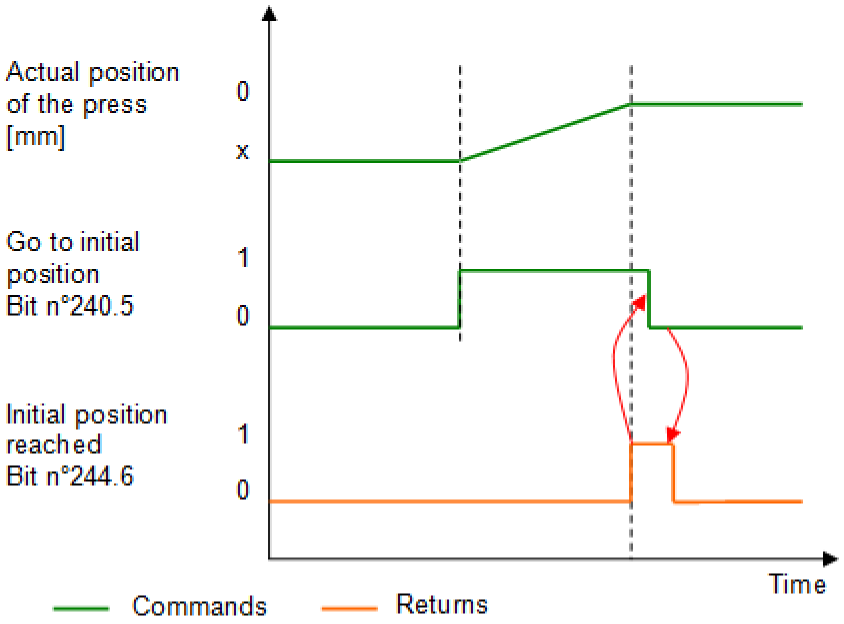


Figure 3: Chronogram to go to the initial position

Go to the release position

To go to the release position, you must set bit n°240.4. The press will then move and when the release position is reached the output bit n°252.0 will set to "1", you can then reset the control bit n°240.4 to "0". (see figure 4).

The press release position is a default parameter to be entered in MecaMotion.

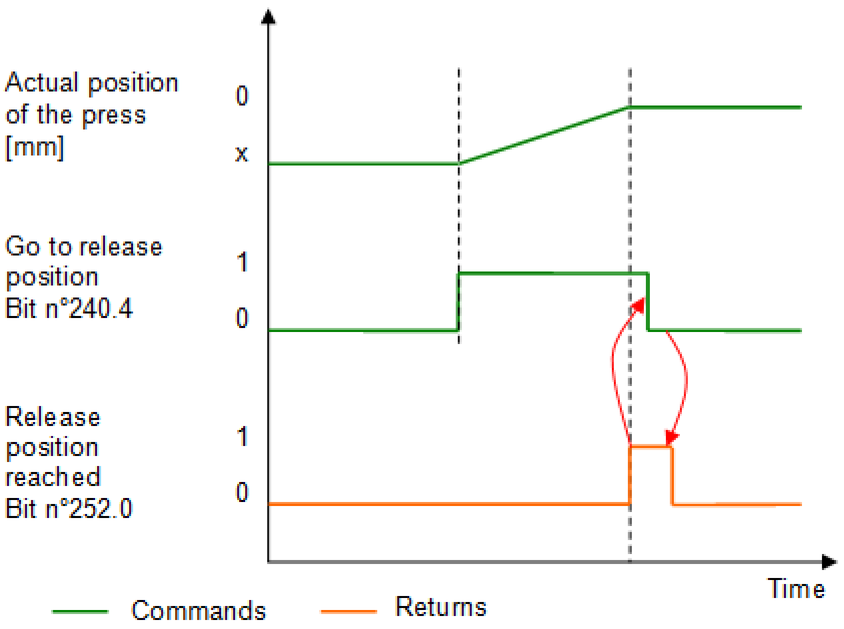


Figure 4: Chronogram to go to the release position

Activating a program number

To be able to activate a program number, there must be no program running (output bit n°244.0 must be at "0").

If this is the case, you must send the number of the program to be activated to input byte n°224, wait 50 ms, then validate this number by setting input bit n°240.1 for 50 ms.

When the active part program number is updated in output byte n°224, it means that the change has been made, you can then start the actual program using input bit n°239.0.

If the return of the active program number (output byte n°224) does not update, check that no errors are present.

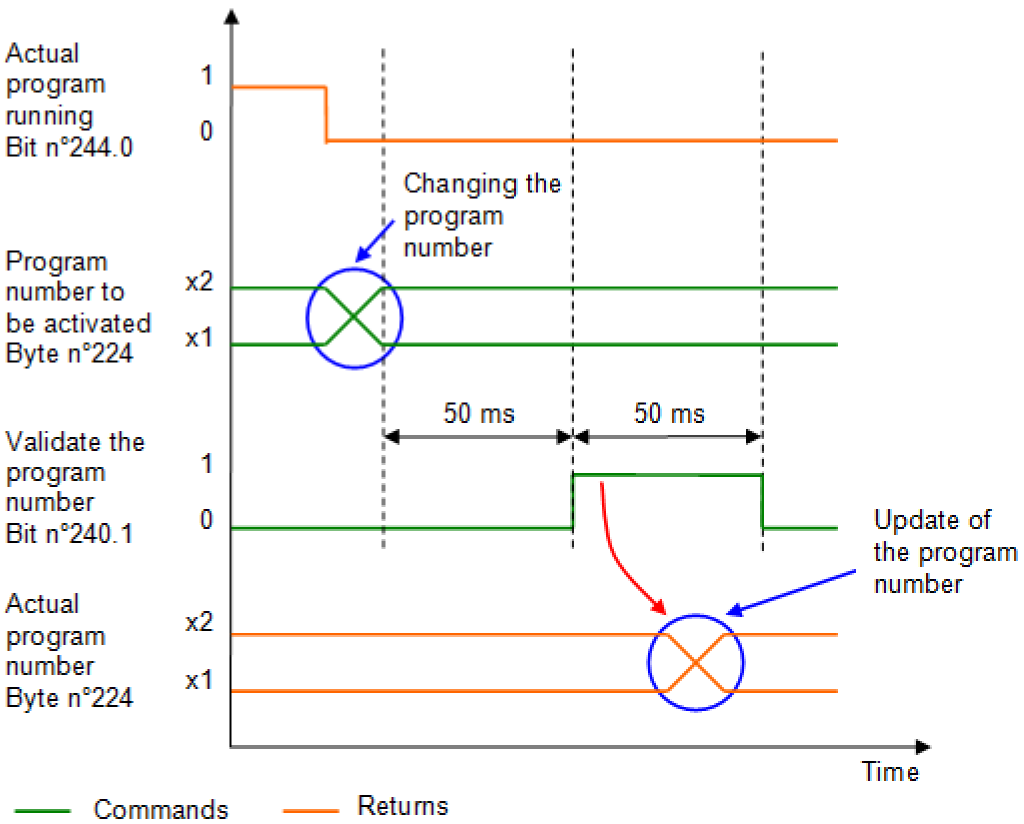


Figure 5: Chronogram activation of a program number

Activating an envelope number

To activate an envelope, you must send its number to input byte n°241, wait 50 ms, then validate this number by activating input bit n°240.6 for 50 ms. When the active envelope number is updated in output byte n°251, it means that the change has been made.

If the number of the active envelope does not update, check that no errors are present (output bit n° 244.5).

Important, you can choose at any time to work with or without the envelope using bit n°240.7. This bit must be at "1" to work with the envelope.

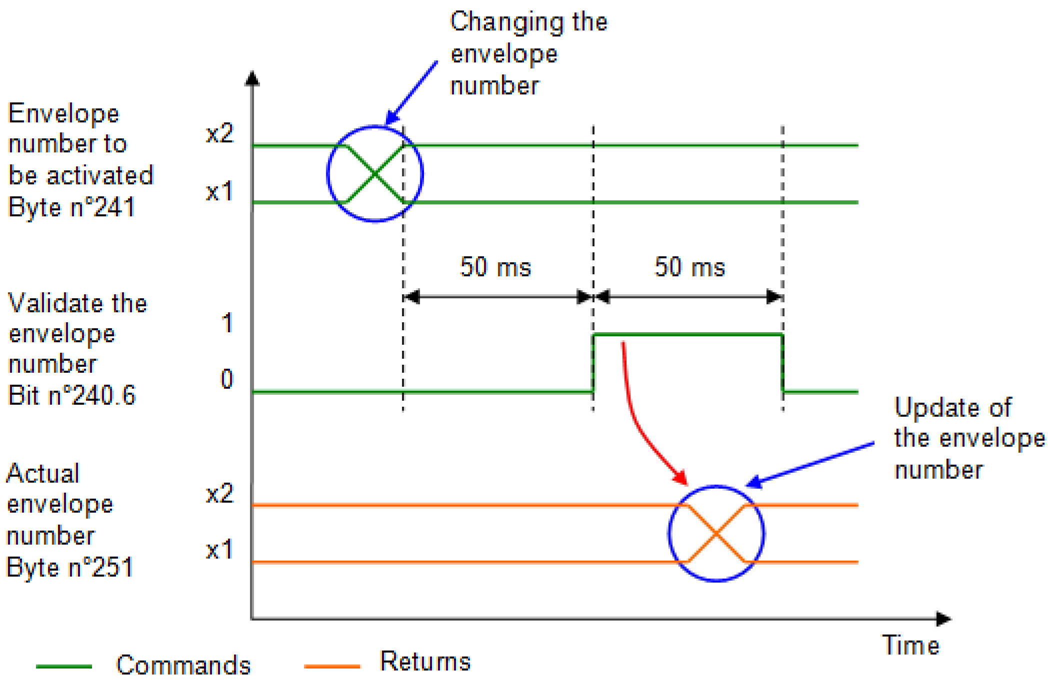


Figure 6: Chronogram activation of an envelope number

Starting a program

Before starting a program, you must make sure that no errors are present (output bit n°244.5 at "0"), that the program is not running (output bit n°244.0 at "0") and that the press is in automatic mode (output byte n°225 = 16#02).

If the above conditions are met, you can start the actual program by setting input bit n°239.0, this bit must be maintained at "1" until the program execution is completed (output bit n°244.1 switches to "1"). When you have the information that the program is finished or that an error is present, you can set the program start command to "0" (input bit n°239.0).

If an error is present, you must set the error acknowledgement bit n°239.6.

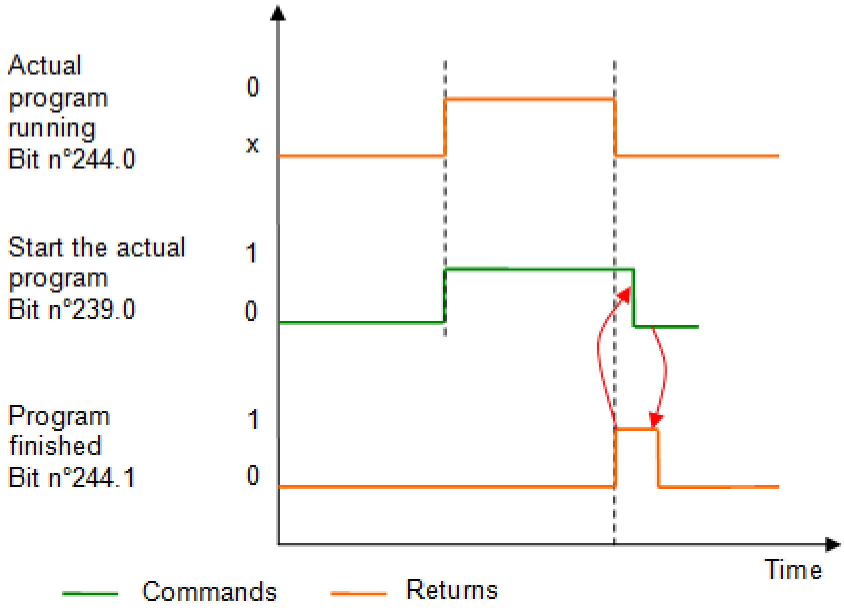


Figure 7: Chronogram starting the actual program

Using the force sensor in manual mode

In manual mode, if you want to reset the force sensor value to zero, you must set input bit n°239.5 for 200 ms.

To select the range of the force sensor, you must set bit n°239.5 to "1" (force sensor in reset mode), wait 50 ms, send the number of the range chosen in input byte n°238, wait 50 ms and then set bit n° 239.5 to "0" to switch the force sensor to measurement mode.

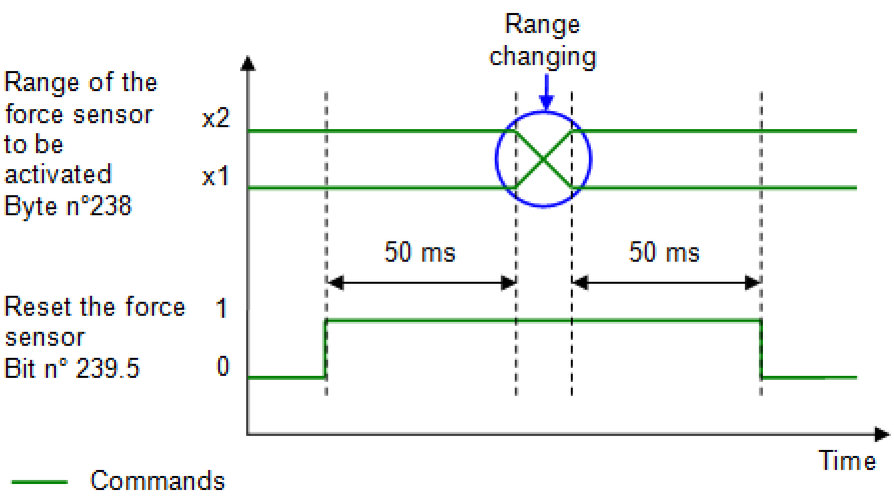


Figure 8: Chronogram change of force sensor range

Currently, there are 2 possible ranges:

- Range 1 (small range), the value to be transferred to the byte is "1".
- Range 2 (large range), the value to be transferred to the byte is "2".

You can view the active range of the force sensor, using output byte 226.

Stopping the axis movement

At any time, and independently of the active mode, you can stop the axis movement using the input bit n°242.0. You can then resume the movement using input bit n°242.1. If a part-program was running at the time of stopping, it is paused and if you restart the movement of the axis, program execution resumes.

Program stop with instruction break point

When a break point instruction is present in the active program, you have the choice to do the break or not. This choice is made with input bit n°240.2 to be set to "1" if you want to make the breaks.

When the program execution is paused, you must set input bit 240.3 for 50 ms to continue the execution.

Stopwatchs

When you use the stopwatch instruction in a program, you can view the stopwatches that are currently being scrolled via PROFINET.

Up to 5 stopwatches can be programmed.

Below is the list of bit addresses to view the stopwatches currently running:

- Stopwatch 1: Output bit n°245.2
- Stopwatch 2: Output bit n°245.3
- Stopwatch 3: Output bit n°245.4
- Stopwatch 4: Output bit n°245.5
- Stopwatch 5: Output bit n°245.6

Viewing press data

It is possible to visualize the position, speed and force of the press in real time.

Actual position of the press [mm]: REAL output n°228

Actual speed of the press [mm/s]: REAL output n°232

Actual measuring sensor value [mm]: REAL output n°236 (option)

Actual value of the force sensor [N]: REAL output n°240

PROFINET error list

Press errors are given to the PLC via the PROFINET link.

As explained in the chapter "Overview of the PROFINET dialogue", these errors are given by the 5 bytes located at addresses 245... 249.

Each byte is composed of 8 bits and each bit corresponds to a specific error => 40 errors maximum.

List of possible errors:

Error number	Type	Description erreur	Direction (from the press)	Address	Format
1	Error	Internal error, contact Softeca	OUT	246.0	BIT
2	Error	The program number to be activated doesn't exist	OUT	246.1	BIT
3	Error	Force overload with range 1 (small range)	OUT	246.2	BIT
4	Error	Force overload with range 2 (large range)	OUT	246.3	BIT
5	Error	Emergency stop detected	OUT	246.4	BIT
6	Error	Axis in technological error	OUT	246.5	BIT
7	Error	External encoder error (measuring sensor)	OUT	246.6	BIT
8	Error	Program execution timeout (Time set in the MecaMotion parameters, the function is disabled if the parameter is at 0)	OUT	246.7	BIT
9	Error	Force limits of a positioning exceeds maximal force on selected range	OUT	247.0	BIT
10	Error	Force limit exceeded in a positioning	OUT	247.1	BIT
11	Error	Telegram too large (part program sent by Ethernet (Mecamotion))	OUT	247.2	BIT
12	Error	"Force Measurement" instruction without "Curve Recording" instruction before	OUT	247.3	BIT
13	Error	Incorrect Firmware License	OUT	247.4	BIT
14	Warning	Curve recording: Too many points (Velocity too low?)	OUT	247.5	BIT
15	Error	Division by 0	OUT	247.6	BIT
16	Error	Arithmetic overflow of an integer	OUT	247.7	BIT
17	Error	Positive software limit switch reached	OUT	248.0	BIT
18	Error	Measuring sensor failed to initialize	OUT	248.1	BIT
19	Error	Absolute encoder reference needed	OUT	248.2	BIT

Error number	Type	Description erreur	Direction (from the press)	Address	Format
20	Error	Execution error of the part program	OUT	248.3	BIT
21	Warning	Force/position out of envelope tolerance	OUT	248.4	BIT
22	Error	Envelope number to be activated doesn't exist	OUT	248.5	BIT
23	Warning	Envelope object offset too large	OUT	248.6	BIT
24	Error	Force regulator function: Position not OK	OUT	248.7	BIT
25	Error	Force regulator function: Force out of tolerance	OUT	249.0	BIT
26	Error	Negative software limit switch reached	OUT	249.1	BIT
27	Error	Maximal position limit exceeded in a positioning	OUT	249.2	BIT
28	Error	Minimal position limit exceeded in a positioning	OUT	249.3	BIT
29	Error	Speed limit exceeded in a positioning	OUT	249.4	BIT
30	Error	Acceleration/deceleration limit exceeded in a positioning	OUT	249.5	BIT
31	Error	Position limit of a positioning instruction exceeds the value of the control parameter	OUT	249.6	BIT
32	Error	Speed limit of a positioning instruction exceeds the value of the control parameter	OUT	249.7	BIT
33	Error	Acceleration/deceleration limit of a positioning instruction exceeds the value of the control parameter	OUT	250.0	BIT
34	Error		OUT	250.1	BIT
35	Error		OUT	250.2	BIT
36	Error		OUT	250.3	BIT
37	Error		OUT	250.4	BIT
38	Error	Unknown message number sending by the HMI (MecaMotion)	OUT	250.5	BIT
39	Error	Length of the message sending by the HMI (MecaMotion), not ok	OUT	250.6	BIT
40	Error	Automatic mode missing	OUT	250.7	BIT

Table 1: Errors list

PROFINET press control FB

To facilitate the control of the press from a PLC, we have developed a function block (FB_MecaMotion_press).

This one includes all PROFINET inputs and outputs required to control the press.

List of block inputs and outputs

Variable name	Declaration	Type of data	Description
Hardware_identifier_press_inputs	INPUT	HW_IO	Hardware identifier of the PROFINET inputs
Hardware_identifier_press_outputs	INPUT	HW_IO	Hardware identifier of the PROFINET outputs
Ack_error	INPUT	BOOL	Acknowledge present errors (Rising edge detection)
Operating_mode	INPUT	BYTE	Operating mode to be activated (Manual = 1, Automatic = 2)
Manual_positioning_relative_mode	INPUT	BOOL	Manual positioning mode (Absolute = 0, Relative = 1)
Manual_position	INPUT	REAL	Position setpoint for manual positioning[mm]
Manual_speed	INPUT	REAL	Speed setpoint for manual positioning [mm/s]
Manual_acceleration_deceleration	INPUT	REAL	Acceleration/deceleration setpoint for manual positioning[mm/s²]
Start_manual_positioning	INPUT	BOOL	Start manual positioning (This input must be kept at "1" until the "Manual_position_reached" output is active)
Jog_downwards	INPUT	BOOL	Run on sight, the press goes down as long as the input is active (Manual Mode)
Jog_upwards	INPUT	BOOL	Run on sight, the press goes up as long as the input is active (Manual Mode)
Go_to_initial_position	INPUT	BOOL	Go to the initial position entered in the press parameters from MecaMotion
Go_to_release_position	INPUT	BOOL	Go to the release position entered in the press parameters from MecaMotion
Stop_movement	INPUT	BOOL	Stop the axis movement (manual or automatic mode) and pause the execution of the part program (Rising edge detection)
Continue_movement	INPUT	BOOL	Resume the axis movement (manual or automatic mode) and continue executing the part-program (Rising edge detection)
Reset_force_sensor	INPUT	BOOL	Reset the force sensor, the input must be "1" for the reset to take place (Rising edge detection) (Manual mode)

Variable name	Declaration	Type of data	Description
Force_sensor_range	INPUT	BYTE	Choice of force sensor range (1 = small range, 2 = large range) The force sensor automatically switches to "reset" mode each time the range is changed (Manual Mode)
Program_number	INPUT	BYTE	Program number to be activated (from 1 to 253) (Program activation is done as soon as the value of the input is modified)
Envelope_number	INPUT	BYTE	Envelope number to be activated (from 1 to 253) (The decoding of the envelope is done as soon as the value of the input is modified)
Envelope_monitoring	INPUT	BOOL	Enable or disable envelope control (The input must be set to "1" for the control to be active)
Start_program	INPUT	BOOL	Start the execution of the program. (This input must be kept at "1" until the "Program_finished" or "Error" output is active)
Enable_break_points	INPUT	BOOL	Enable breakpoints (The input must remain active to make breaks at the breakpoints)
Continue_after_break_point	INPUT	BOOL	Continue after breakpoint (Rising edge detection)
DB_setpoints_results	INPUT	DB_ANY	Data block containing the user variables that allow setpoints to be sent and results to be received
Error	OUTPUT	BOOL	At least one error is present if this output is "true".
Message	OUTPUT	BOOL	At least one message is present if this output is "true".
Axis_enabled	OUTPUT	BOOL	When this output is at "1" the press is ready to work
Press_cpu_started	OUTPUT	BOOL	When you turn on the press, wait until this output changes to "1" before sending a command.
Actual_operating_mode	OUTPUT	BYTE	Actual operating mode (Manual = 1, Automatic = 2)
Manual_positioning_relative_mode_active	OUTPUT	BOOL	Actual manual positioning mode (Absolute = 0, Relative = 1)
Manual_position_reached	OUTPUT	BOOL	This output is activated when the manual position is reached
Initial_position_reached	OUTPUT	BOOL	This output is activated when the axis has reached the initial position
Release_position_reached	OUTPUT	BOOL	This output is activated when the axis has reached the release position
Force_sensor_actual_range	OUTPUT	BYTE	Actual force sensor range (1 = small range, 2 = large range)
Actual_program_number	OUTPUT	BYTE	Actual program number (from 1 to 253)
Program_running	OUTPUT	BOOL	This output is activated when the program is running

Variable name	Declaration	Type of data	Description
Program_finished	OUTPUT	BOOL	This output is activated when the program execution is finished.
Actual_envelope_number	OUTPUT	BYTE	Actual envelope number (from 1 to 253)
Stopwatch_1_running	OUTPUT	BOOL	Stopwatch 1 is running when the output is at "1".
Stopwatch_2_running	OUTPUT	BOOL	Stopwatch 2 is running when the output is at "1".
Stopwatch_3_running	OUTPUT	BOOL	Stopwatch 3 is running when the output is at "1".
Stopwatch_4_running	OUTPUT	BOOL	Stopwatch 4 is running when the output is at "1".
Stopwatch_5_running	OUTPUT	BOOL	Stopwatch 5 is running when the output is at "1".
Program_on_break_point	OUTPUT	BOOL	The program is stopped at a breakpoint when this output is at "1".
Actual_position	OUTPUT	REAL	Actual axis position in[mm]
Actual_speed	OUTPUT	REAL	Actual axis speed setpoint in[mm/s]
Actual_measuring_sensor_value	OUTPUT	REAL	Actual measuring sensor value in[mm]
Actual_force	OUTPUT	REAL	Actual force value[N]
Errors_messages	OUTPUT	Struct of 120 Bool	Errors and messages

Table 1: Designation of the inputs/outputs of the block

Siemens PLC

On the "Hardware_identifier_presse_inputs" and "Hardware_identifier_presse_outputs" inputs, you must give the hardware identifier of the Profinet input and output range in decimal format.

When using a Siemens PLC, you must use a data block to write/read user variables. This block must have the following structure and must be transferred to the "DB_setpoints_results" input.

Structure of the setpoints / results DB:

Type	Start address	Size and format	Description
Setpoint	0.0	50 DWORD	User input variables of type DINT or REAL
Setpoint	200.0	32 BOOL	User input variables of type BOOL
Setpoint	204.0	5 DWORD	Reserve
Result	224.0	50 DWORD	User output variables of type DINT or REAL
Result	424.0	32 BOOL	User output variables of type BOOL
Result	428.0	5 DWORD	Reserve

Table 2: Structure of the setpoint / result data block under Siemens

Beckhoff PLC

In the FB "MecaMotion Press" for Beckhoff PLCs, the declaration of the Profinet input and output ranges is done via 2 Bytes tables of length 254.

When using a Beckhoff PLC, user variables are written/read via tables.

Type	Size and format of the table	Description
Setpoint	50 DWORD	User input variables of type DINT or REAL
Setpoint	32 BOOL	User input variables of type BOOL
Result	50 DWORD	User output variables of type DINT or REAL
Result	32 BOOL	User output variables of type BOOL

Table 3: Setpoints/results tables with Beckhoff PLC

FB errors

In addition to the Profinet press errors listed in the "PROFINET error list" topic, the errors below may appear, these errors are part of the Boolean structure of the "Errors_messages" output. If you use a Siemens PLC, the structure must have 120 Booleans.

Error n°	Type	Error description	Format
80	Error	Press not started (The press is starting)	BOOL
81	Error	Manual mode not active (manual controls cannot be used)	BOOL
82	Error	Program number or envelope number not active at program start (setpoint different from return)	BOOL

Table 4: Internal errors in the functional block

Error acknowledgement

To acknowledge the errors, you must set the "Ack_error" input to "1". (Rising edge detection)

Operating mode changing

There are 2 possible operating modes, manual = 1 or automatic = 2.

To activate one of these modes, you must write one of the values listed above in the "Operating_mode" input byte. You can then check that the mode has been activated, using the output byte "Actual_operating_mode".

Positionnement manuel

To perform manual positioning, you must enter the following input parameters:

- Manual_positioning_relative_mode (absolute=0, relative=1)
- Manual_position (position setpoint [mm])
- Manual_speed (speed setpoint [mm/s])
- Manual_acceleration_deceleration (acceleration and deceleration setpoint [mm/s2])

Once these parameters have been entered, you can start positioning using the "Start_manual_positioning" input, this input must be maintained to "1" until the "Manual_position_reached" output becomes true. Then, when you reset the "start" input, the output "Manual_position_reached" becomes false too. You can then perform a new positioning.

Run on sight (JOG)

The "Jog_downwards" and "Jog_upwards" inputs are used to move axis in run on sight. As long as the input is true, the axis moves, otherwise it stops.

Be careful, you must enter a speed in the "Manual_speed" input parameter for the movement to take place.

Position initiale et position de dégagement

Go to the initial position (manual or automatic mode): The "Go_to_initial_position" input allow to move axis to the initial position entered in the parameters from MecaMotion. This input must be maintained to "1" until the "Initial_position_reached" output becomes true. It means that the positioning is finished, you can then reset the input and the output will also reset.

Go to the release position (manual or automatic mode): The "Go_to_release_position" input allow to move axis to the release position entered in the parameters from MecaMotion. This input must be maintained to "1" until the "Release_position_reached" output becomes true. It means that the positioning is finished, you can then reset the input and the output will also reset.

Stop movement

In manual or automatic mode, when a rising edge is detected on the "Stop_movement" input, the axis stops and the program execution is paused.

Movement resumption

The resumption of axis movement and program execution takes place when a rising edge is detected on the "Continue_movement" input. (Manual or automatic mode)

Force sensor management

To reset the force sensor, you must set the "Reset_force_sensor" input to "1". The reset is performed for 200[ms] after detecting a rising edge on the input.

To change the range of the force sensor, you must enter the number of the range you want to activate in the "Force_sensor_range" input byte. When you change the range of the force sensor, it is reset for 500[ms].

Program changing

To activate a program, simply enter its number in the "Program_number" input byte. You can then check whether the program has been activated using the "Actual_program_number" output.

Starting the actual program

To start the actual program, the "Program_running" and "Error" outputs must be "false". Then you must activate the "Start_program" input and keep it active until the "Program_finished" output becomes true. When the program is finished, you can reset the "Start_program" input and the "Program_finished" output will also be reset. You can then restart the program.

Envelope changing

To select and activate an envelope, you must enter its number in the "Envelope_number" input byte. You can check that the selected envelope has been activated using the output byte "Actual_envelope_number".

You can activate or deactivate the envelope control at any time using the "Envelope_monitoring" input. This input must be set to "1" for the control to be active.

Break points

If you use the breakpoint instruction in your program, you have the choice to make or not the break during program execution.

To activate the breakpoints you must set the "Enable_breakpoints" input to "1". When the program execution is stopped at a breakpoint, the output "Program_on_breakpoint" changes to "1". You can continue running the program by activating the "Continue_after_breakpoint" input (rising edge detection).

Stopwatches

If you use the stopwatch instruction in your program, you can view the running stopwatches at any time using the "Stopwatch_…_running" outputs.

Press technology data

You can view at any time the position and speed of the axis, the measuring sensor value and the force value using the following outputs:

- "Actual_position" [mm] REAL Format
- "Actual_speed" [mm/s] REAL Format
- "Actual_measuring_sensor_value" [mm] (option) REAL Format
- "Actual_force" [N] REAL Format

Example of use of the press control FB:

We would like to insert a rivet.

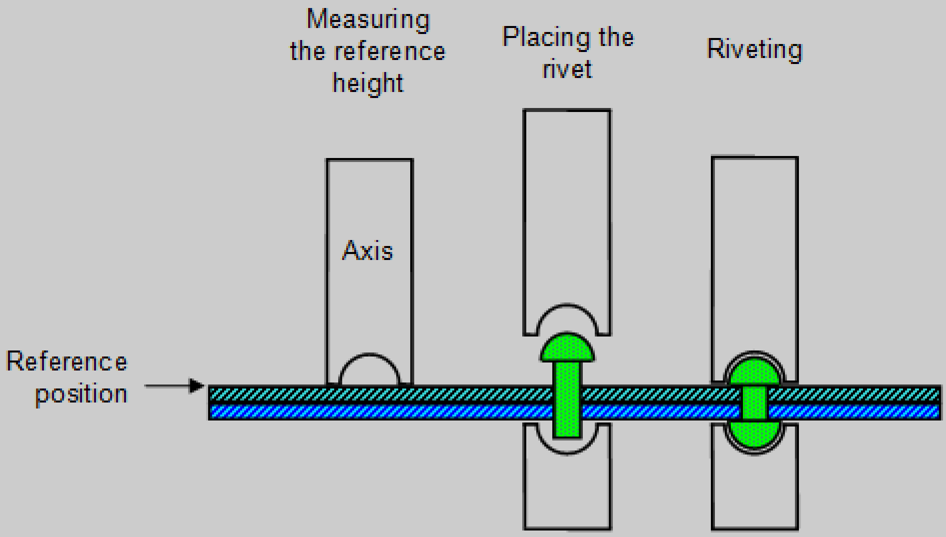


Figure 1: Insertion of a rivet

To do this, we use two programs created in MecaMotion (Programming and press configuration software).

The first program measures the reference height that will be used to determine the final insertion position.

In this one (Figure 2), the reference height is measured using the "position measurement" function. This function can be used when the press is equipped with a precision switch. The measured position will allow you to determine the insertion position of the rivet.

The position measurement function must always be located before a positioning, it is during this one that the measurement is performed. In this positioning, you must set a position that cannot be reached (lower than the reference), the axis will be stopped when the precision switch is switched by the "position measurement" function.

Before executing this program, you must send the setpoints "reference pre-position" and "unreachable position" to the press via PROFINET. When the cycle is finished, the press returns by PROFINET, the maximum force measured during detection and the reference position.

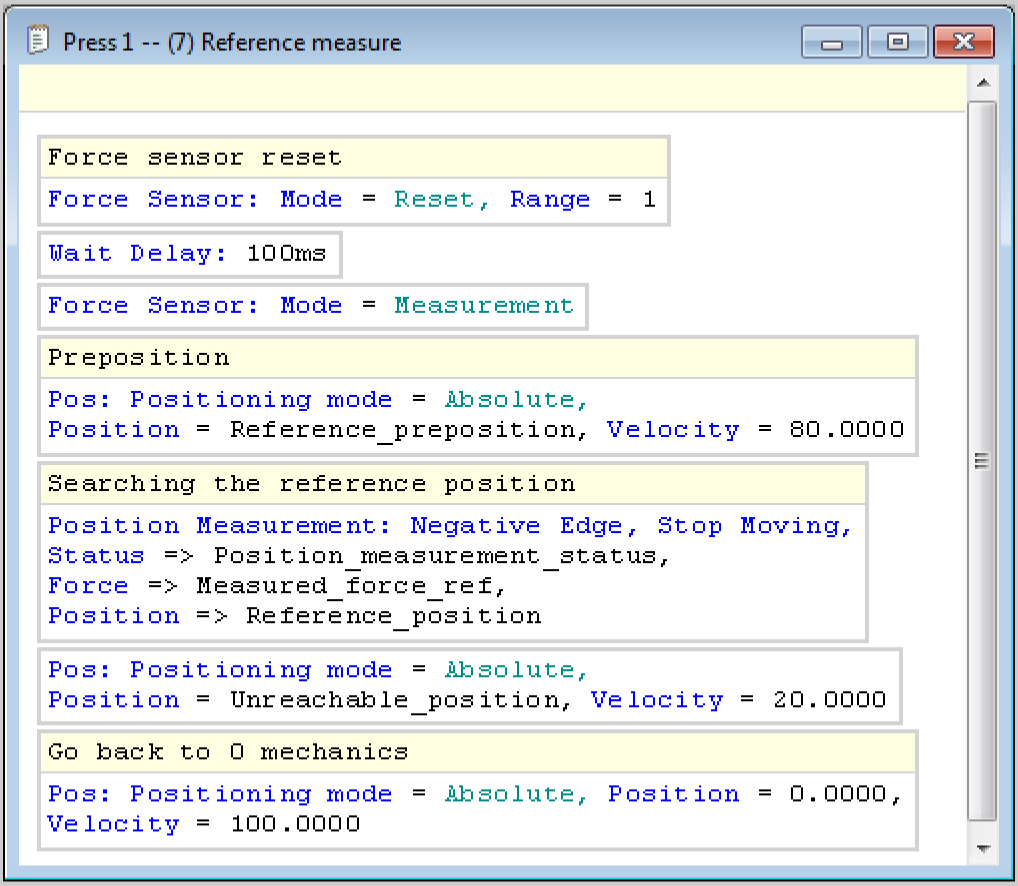


Figure 2: MecaMotion Program - Reference Measurement

The second program allows the rivet to be inserted, this insertion is done in position. The actual insertion position is calculated as follows: "actual insertion position" = "reference position" + "precision switch stroke" - "insertion height"

Before starting the program, you must give the following setpoints by PROFINET:

- *Preposition (Position up to which the press descends at high speed)*
- *Switch stroke (Precision switch stroke)*
- *Insertion height (desired insertion height compared to the reference)*
- *Insertion speed (descent speed during insertion)*

When insertion is complete, the maximum force measured during the cycle is returned by PROFINET.

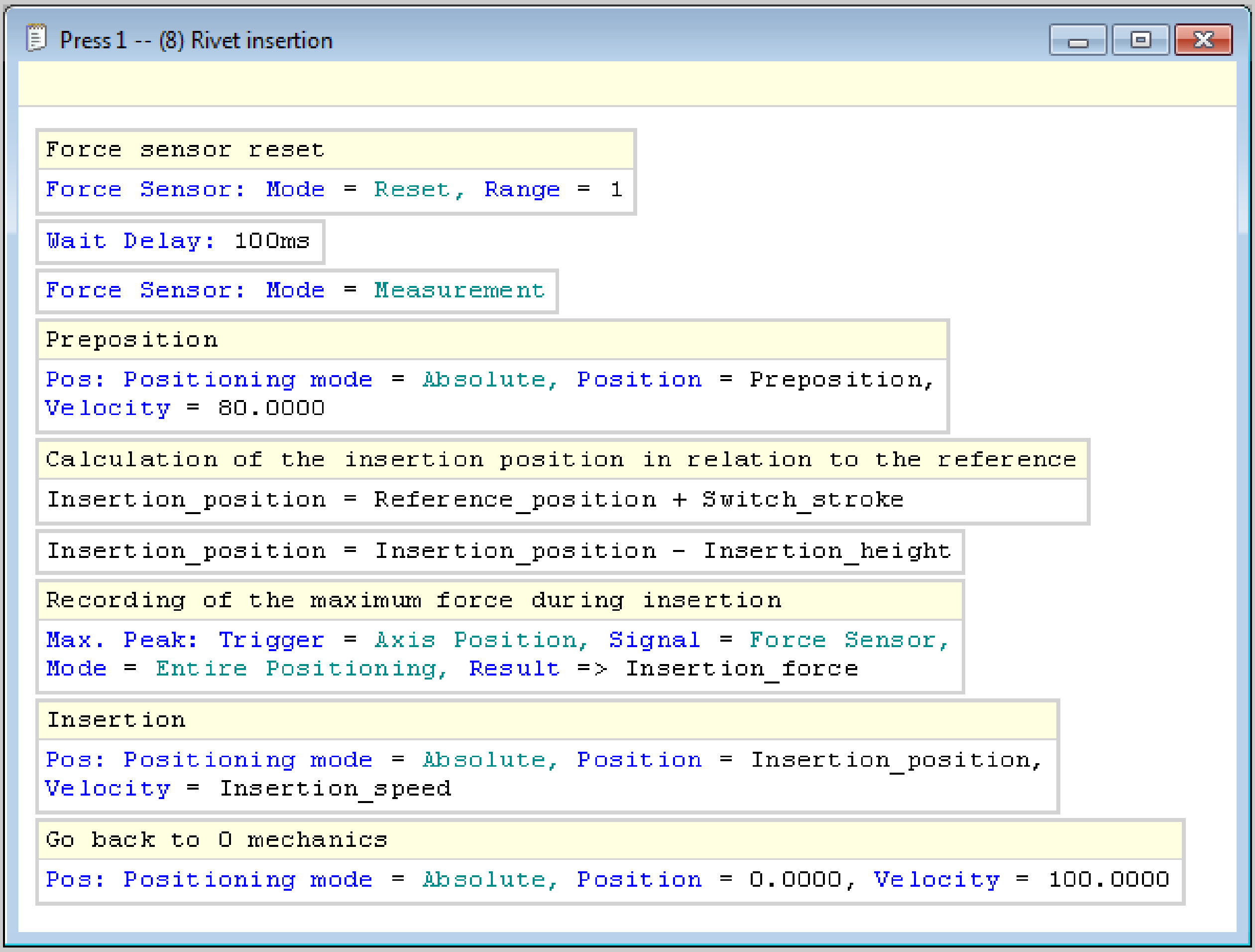


Figure 3: MecaMotion program - Rivet insertion

After having carried out the two programs, you must associate in the PLC and MecaMotion the PROFINET variables used to give the setpoints and receive the results.

In Figure 4 below, you must declare all the setpoints that will have to be sent to the PLC.

It is essential to enter the start address of the Profinet input/output variables (they must be the same as those configured in the PLC).

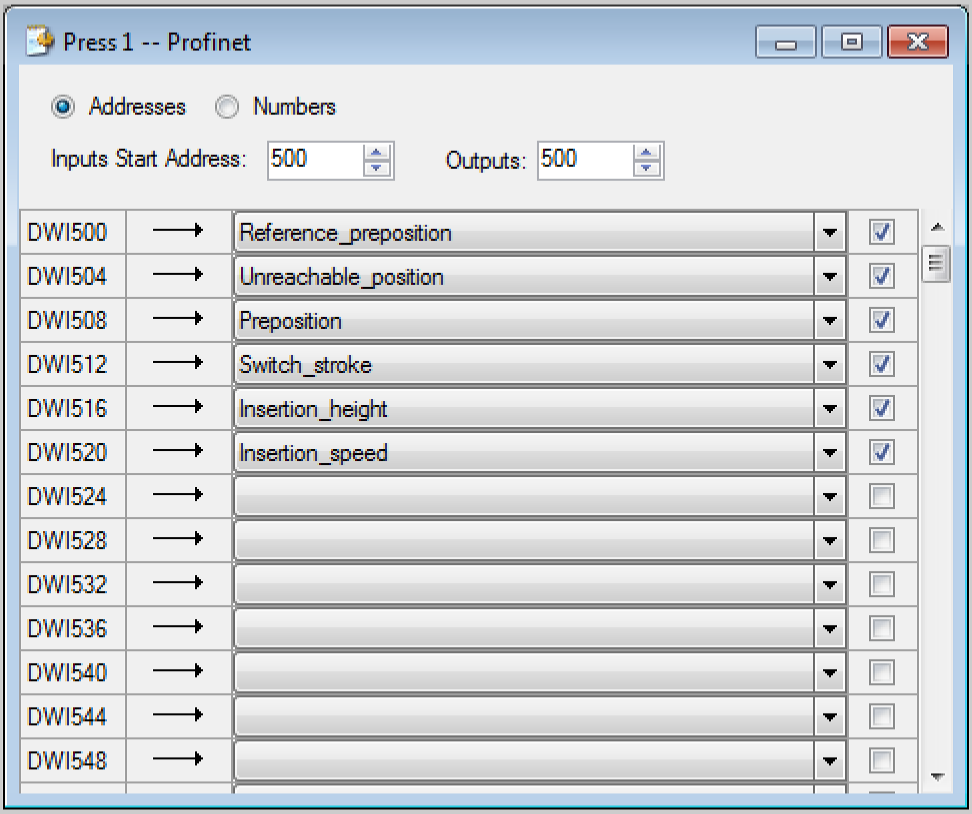


Figure 4: Association of setpoint variables to PROFINET variables from Mecamotion

In Figure 5 below, you associate all the result variables that will be returned to the PLC.

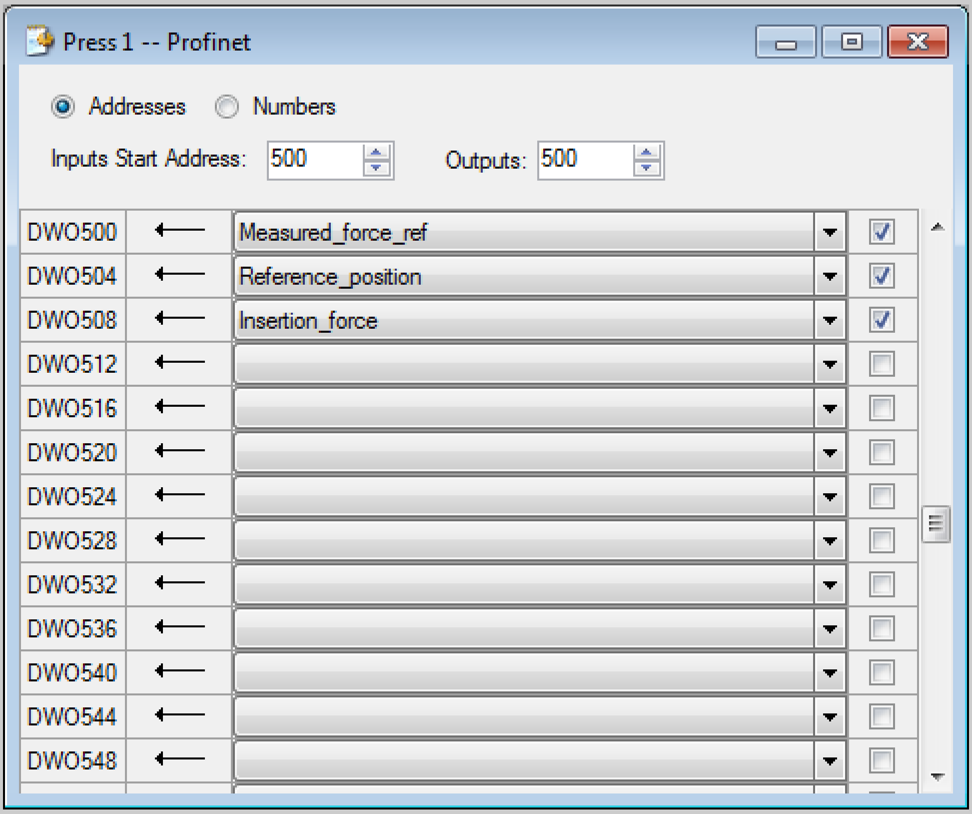


Figure 5: Association of result variables to PROFINET variables from MecaMotion

When the programs are created and the input/output variables are declared in MecaMotion, you must, in the PLC programming platform, call the FB and assign all its inputs/outputs. These are all detailed in the block commentary.

Then, you must perform the programming below:

- *Send the setpoints and read the results to the PROFINET addresses configured in MecaMotion.*
- *Enable automatic mode: To activate automatic mode, you must check that the "Press_cpu_started" output is active and then you must set the value 2 in the "Mode" input to activate automatic mode. You can then check that the mode has been activated using the "Actual_operating_mode" output.*
- *Select the program number and start it.*

  Before starting a program, you must check that the "Axis_enabled" and "Press_cpu_started" outputs are active and that the outputs "Error" and "Program_running" are not active. If an error is present, you can acknowledge it using the "Ack_error" input.

  You must then enter the number of the program you want to activate in "Program_Number" input.

  Finally, you can activate the "Start_program" input to start the execution of the program. This input must be maintained at "1" until one of the "Program_finished" or "Error" outputs is active.

Once you have completed all the above steps, all you have to do is run the "reference_measurement" program and then the "rivet_insertion" program.

Creating a new project

To create a new project, you must proceed as follows:

1. In the menu bar, select "File > New project" or click on the "New" button located in the general toolbar (see figure 1).
2. By making one or other of these selections, a new tree structure is created under the folder "Project". For the moment the name of this project is "New project" and appears in the software header (see figure 1).

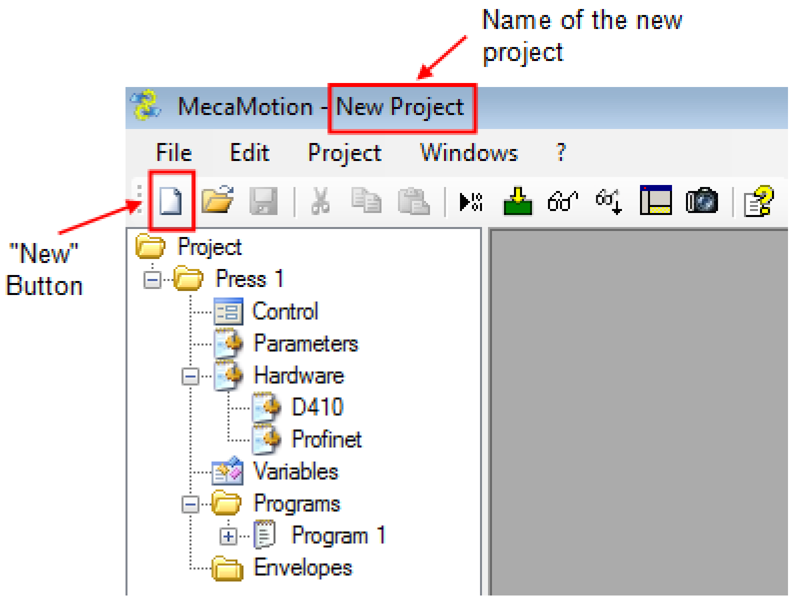


Figure 1: Creating a new project

3. When the new project is created, the first thing to do is to save it. To do this, select "**File > Save as...**". A window opens in which, under "**File name**", give the name of the project, then select the location where you want to save it using the explorer.
4. The project name is displayed in the header of the press software (Figure 1).

Establishing the Ethernet connection

The connection between the computer where the MecaMotion programming software is located and the press is made via an Ethernet bus.

This Ethernet connection is used to send the various part programs, envelopes, parameters and user variable association to the press. The reception of the force/position curves is also done through this connection. In addition, it allows the press to be controlled in standalone mode.

How to connect the computer to the press

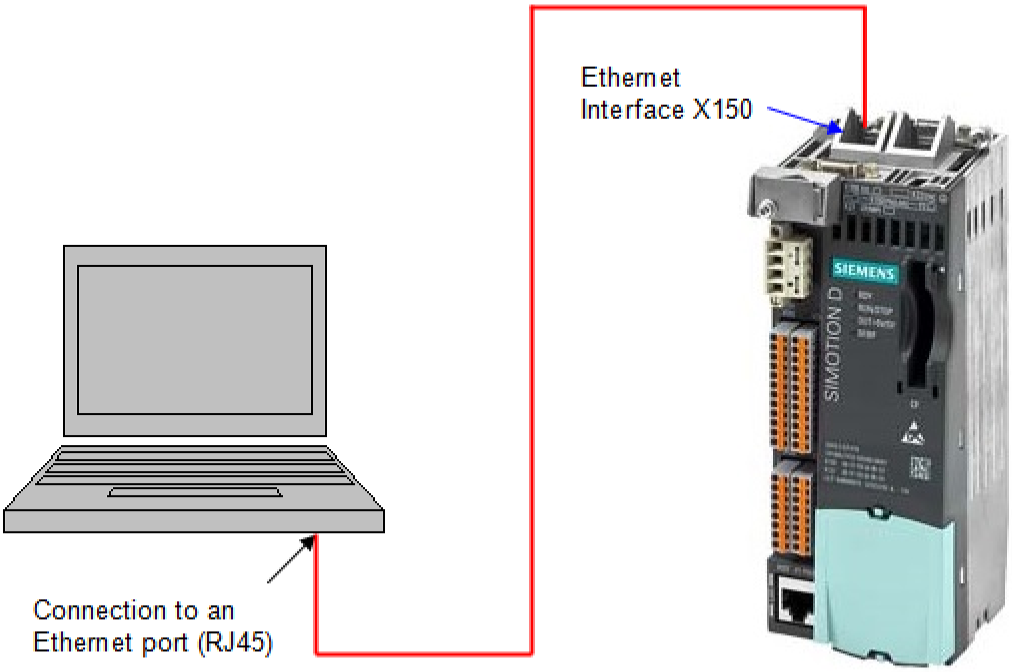


Figure 1: Connection of the Ethernet link

The IP address of the press must be saved in the project. This address is written on the press control unit (D410-2) (see Figure 2).

Figure 2: Where to find the IP address on the press hardware

To enter this address in MecaMotion, you must, in the project tree, double-click on the "Hardware" tab and when the window in figure 3 is open, enter the IP address of the press in the field provided.

To be able to connect to the press, the programming platform uses the port n°1025.

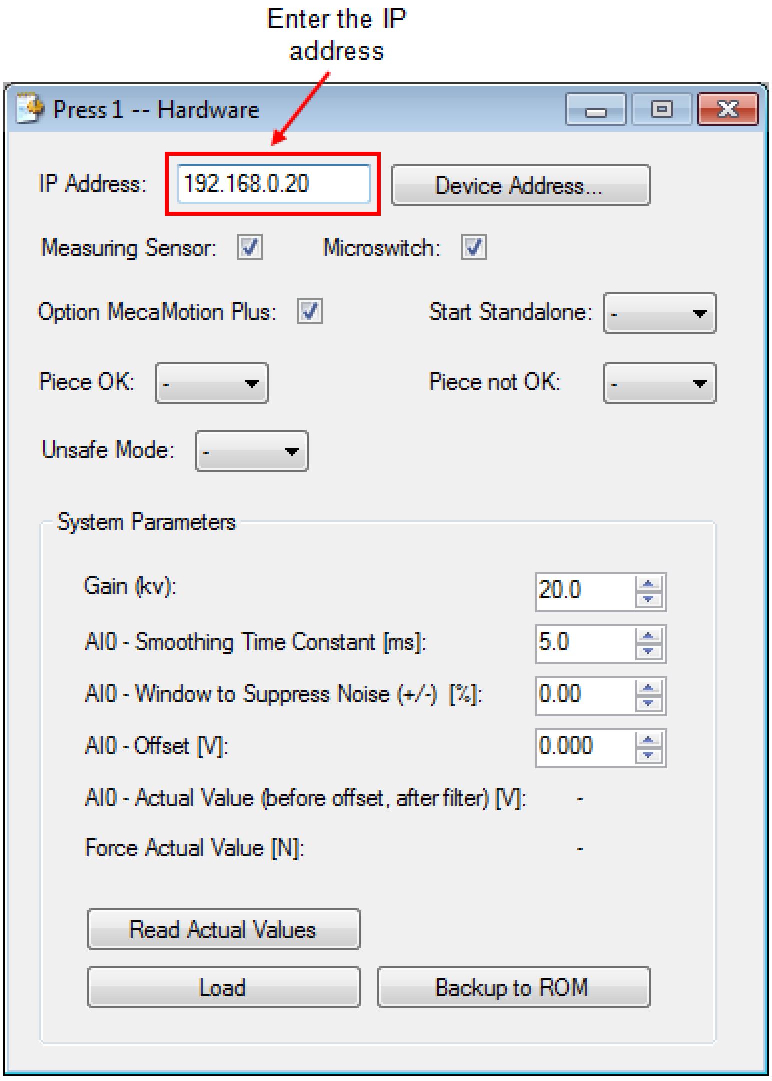


Figure 3: Enter the IP address in the project

Changing the IP address and subnet mask

If necessary, you can change the IP address and subnet mask of the press as well as the default gateway address.

To access the address change window, click on the "device address" button in the "Hardware" window (figure 3), then in the window that opens (figure 4) you can read the current addresses and edit them.

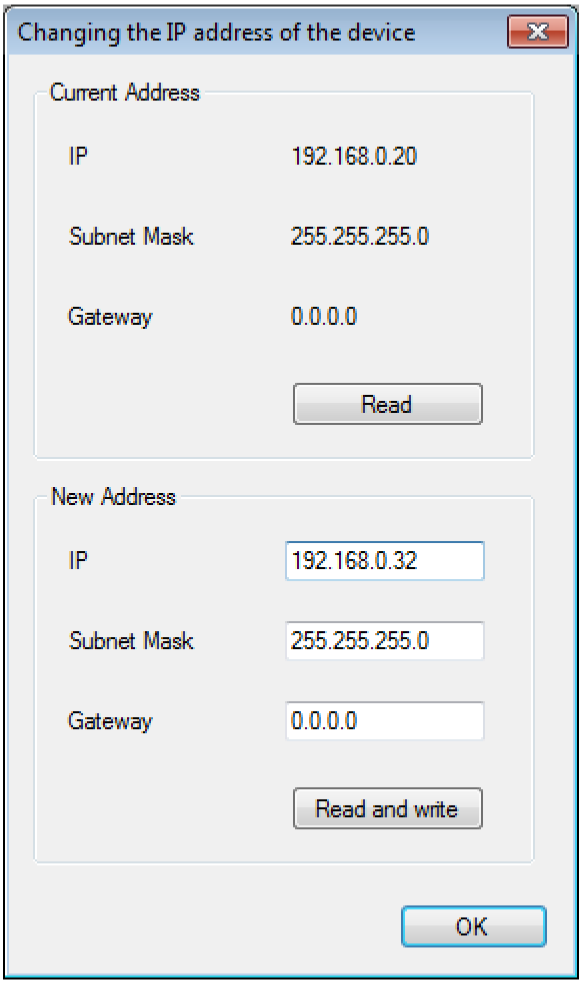


Figure 4: Change IP address

The "Read" button displays the current addresses. (reading may take some time)

You can change the IP address of the press, the subnet mask and the gateway address by entering them in the "new address" section, then click on the "Read and write" button to make the change. (this may take some time)

When you change the IP address of the press, the "IP Address" field in the "Hardware" window is replaced by the new address, your computer will then be automatically reconnected to the press after the address change.

Be careful, if you make changes and the address of your computer is no longer part of the same network as the press, the window below will appear (figure 5).

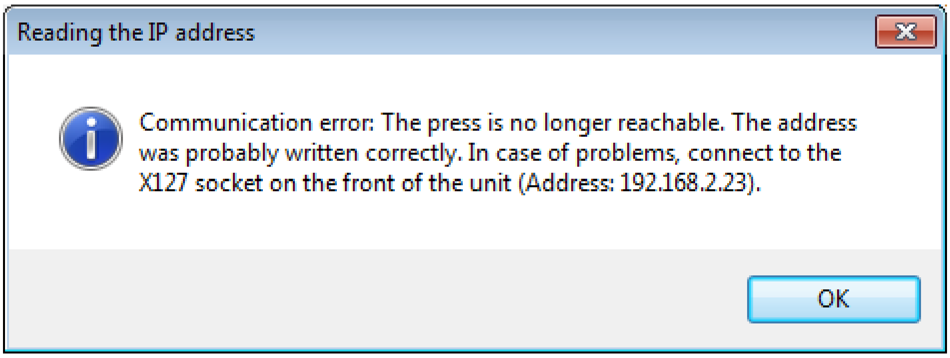


Figure 5: Communication error

In order for your computer to reconnect to the press, you must adjust the addresses on your Ethernet card.

To do this, open the start menu of your computer and click on "Control Panel", in it, click on "Network and Sharing Center", then "Change adapter settings" (Figure 6).

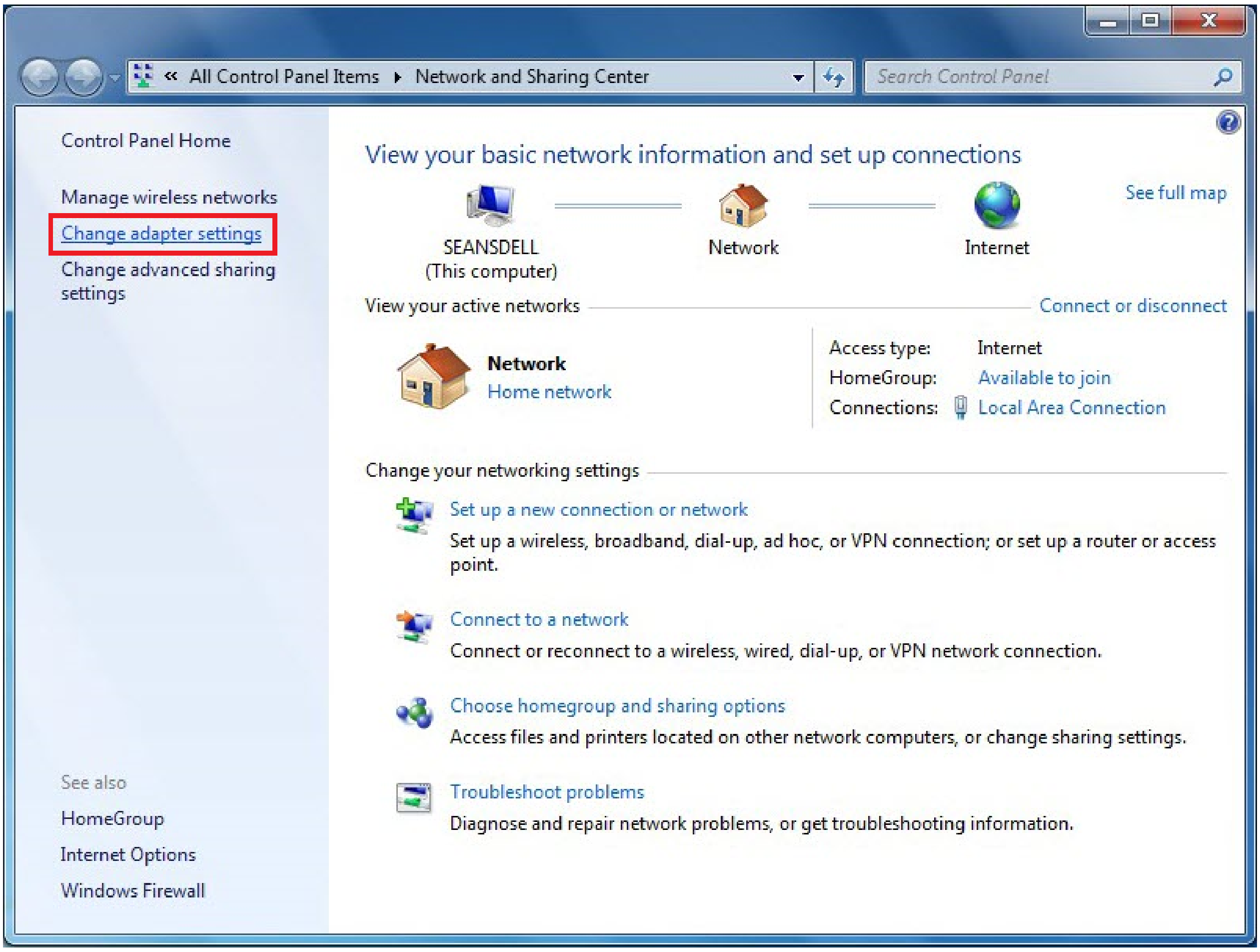


Figure 6: Network center - Change adapter settings

Right-click on the Ethernet card you are using and click on "Properties".

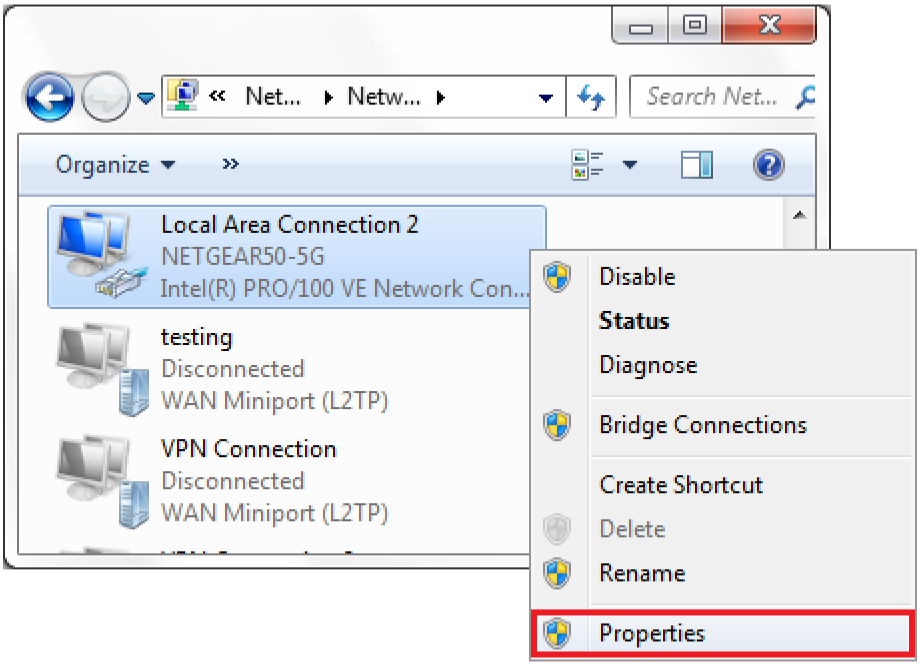


Figure 7: Properties of the Ethernet card

In the "Networking" tab of the property window, double-click on "Internet Protocol version 4 (TCP/IPv4)". Make sure that this protocol is active, the box to the left of the line must be selected.

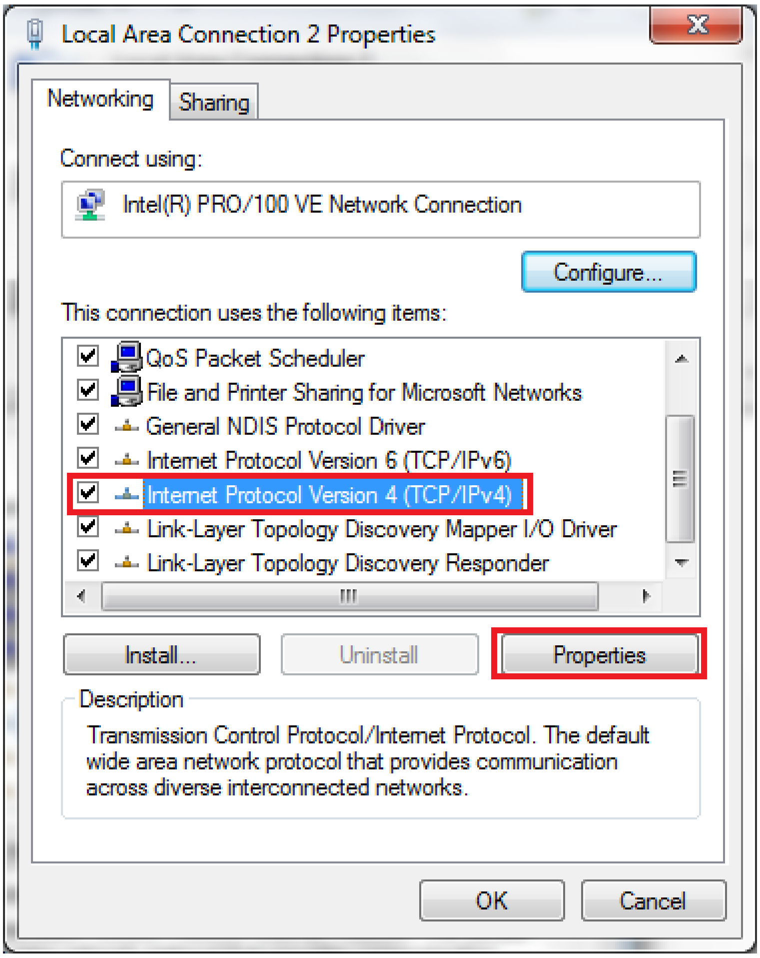


Figure 8: IPv4 protocol

Select "Use the following IP address" to enter an address manually.

Enter the subnet mask you have chosen for the press, an IP address that is part of the same network as the press but different from the press and the default gateway address if you use one.

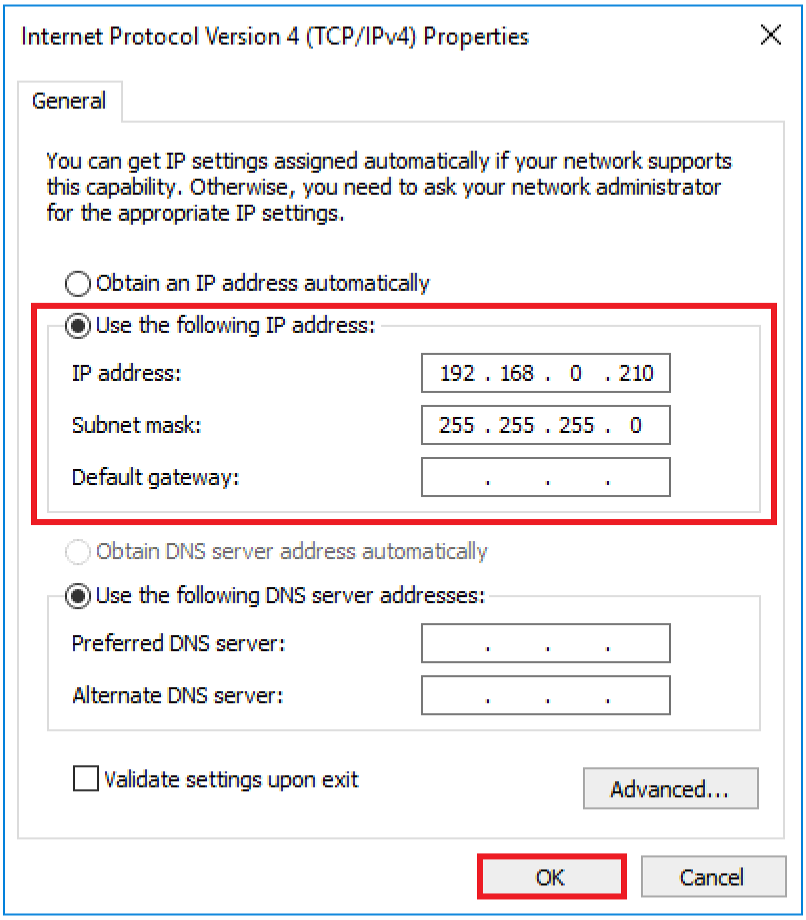


Figure 9: Change of addresses

Once you have entered the addresses of your computer's Ethernet card, you can return to MecaMotion and enter the new press address in the "IP Address" field of the hardware window".

Saving and loading a project in the press

When you load a project into the press, MecaMotion software performs the following tasks:

- Compile and validate programs and envelopes to send them to the press
- Load part programs and envelopes
- Load variable association
- Load press parameters

If before loading the project into the press, the project has not been saved, the programs, envelopes, parameters and variable association will not be saved in the press flash memory.

To load the project in the press, in the menu bar, select "**Project > Load**" or click on the "**Load**" button located in the general toolbar (see figure 1).

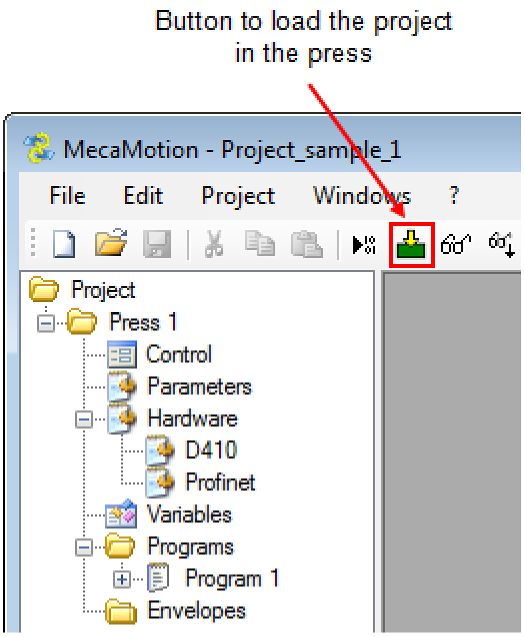


Figure 1: Load button

When a change is made to the project, the save icon will be automatically be enabled. As soon as the save has been performed, this same icon will be disabled (it becomes greyed out).

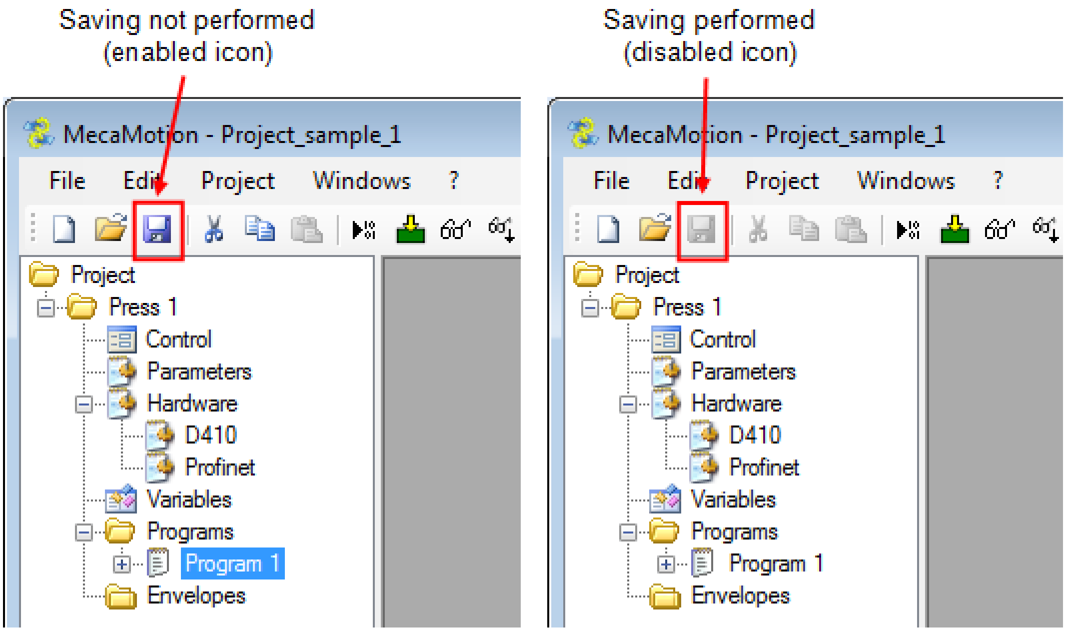


Figure 2: Save icon

Parameters

The "parameters" window accessible from the project tree includes:

- The default parameters used by some part-program instructions

- The press configuration parameters (operating mode of the force sensor, various limits, type of communication...)

The list below lists these different parameters:

Group	N°	Name	Description	Default value	Unit
Positioning	0	Acceleration	If you don't give an acceleration setpoint in a positioning instruction, this value is taken into account.	6000	[mm/s2]
	1	Deceleration	If you don't give an deceleration setpoint in a positioning instruction, this value is taken into account.	6000	[mm/s2]
	2	Acceleration start jerk	If you don't give an acceleration start jerk setpoint in a positioning instruction, this value is taken into account.	50000	[mm/s3]
	3	Acceleration end jerk	If you don't give an acceleration end jerk setpoint in a positioning instruction, this value is taken into account.	50000	[mm/s3]
	4	Deceleration start jerk	If you don't give an deceleration start jerk setpoint in a positioning instruction, this value is taken into account.	50000	[mm/s3]
	5	Deceleration end jerk	If you don't give an deceleration end jerk setpoint in a positioning instruction, this value is taken into account.	50000	[mm/s3]
Force	14	Positive force overload with range 1	When force range 1 is active, if the force exceeds this value (>), stop the axis and the part-program	120	[N]
	15	Positive force overload with range 2	When force range 2 is active, if the force exceeds this value (>), stop the axis and the part-program	1200	[N]
	31	Negative force overload with range 1	When force range 1 is active, if the force exceeds this value (<), stop the axis and the part-program	-120	[N]
	32	Negative force overload with range 2	When force range 2 is active, if the force exceeds this value (<), stop the axis and the part-program	-1200	[N]
Manual movements	9	JOG acceleration	When you make a movement in JOG or a manual positioning, this acceleration value is used	200	[mm/s2]

Group	N°	Name	Description	Default value	Unit
	10	JOG deceleration	When you make a movement in JOG mode or a manual positioning, this deceleration value is used	200	[mm/s2]
	11	Velocity to initial/release position	Axis speed when moving back to the initial or release position (not for part-program)	10	[mm/s]
	12	Acceleration to initial/release position	Axis acceleration when moving back to the initial or release position (not for part-program)	50	[mm/s2]
	13	Deceleration to initial/release position	Axis deceleration when moving back to the initial or release position (not for part-program)	50	[mm/s2]
	29	Release position	Position to release the press from the part	0	[mm]
	30	Initiale position	Cycle start position	0	[mm]
Miscellaneous	19	Part program execution timeout	When a part program is started, if this time is finished before the end of the part program => Timeout error If this value is equal to 0, the time control is disabled	0	[ms]
	27	Maximal positon offset of envelope objects (+/-)	Maximum offset that the objects in the envelope can have. (+/-)	300.00	[mm]
	28	Communication (Profinet or Standalone)	Controlled by a PLC via Profinet or directly from MecaMotion (standalone)	Profinet	
	33	Positive position limit positioning instruction	Maximum value that the positive position limit can have in the positioning instruction	90	[mm]
	34	Negative position limit positioning instruction	Minimum value that the negative position limit can have in the positioning instruction	-5	[mm]
	35	Velocity limit positioning instruction	Maximum value that the velocity limit can have in the positioning instruction	200	[mm/s]
	36	Acceleration deceleration limit positioning instruction	Maximum value that the acceleration and deceleration limit can have in the positioning instruction	7000	[mm/s2]
Measuring sensor	20	Measuring sensor resting position tolerance (+/-)	Position window in which the measuring sensor value must be located at the time of the check	0.05	[mm]
	21	Measuring sensor resting position	Value that the measuring sensor must have at rest (When the press is in measuring sensor control position) If the measuring sensor value (+ or - tolerance) is not equal to this parameter, => Error	0.10	[mm]

Group	N°	Name	Description	Default value	Unit
	22	Measuring sensor check position	Position of the axis at which the measuring sensor value is checked with respect to parameters 20 and 21.	0.00	[mm]
Expert	6	Force sensor sensitivity (range 1)	Data provided by the force sensor manufacturer	4.1	[pC/N]
	7	Force sensor sensitivity (range 2)	Data provided by the force sensor manufacturer	4.1	[pC/N]
	23	Force sensor - Load at full range (range 1)	Parameter given by the charge amplifier of the force sensor	1000.00	[pC]
	24	Force sensor - Load at full range (range 2)	Parameter given by the charge amplifier of the force sensor	5000.00	[pC]
Manufacturer	16	Monitoring display frequency	When the visualization is activated, this parameter gives the time between each refresh of the value of the user variable	200	[ms]
	17	Zero position	Do not modify	0.00	[mm]
	18	Zero position tolerance (+/-)	Do not modify	0.03	[mm]
	25	Output value to select range 1 of the force sensor	State in which the digital output of the D410-2 must be in order for range 1 of the force sensor to be active (0 or 1)	1	
	26	Output value to reset force sensor	State in which the digital output of the D410-2 must be in order for the force sensor reset to be active (0 or 1)	1	

Table 1: Press parameters

These parameters are loaded into the press at the same time as the part programs and envelopes.

To load them, you must first save the project. Then, in the "**Project**" menu, click on the "**Load**" button.

User management

Creating different users allows you to manage the rights of each user of the press.

You can access to the user management window from the menu bar by clicking on "?" then "User management".

If you have not yet created users, the user "None" will be Administrator, this means that if no user is connected, you can use all MecaMotion functions. (figure 1)

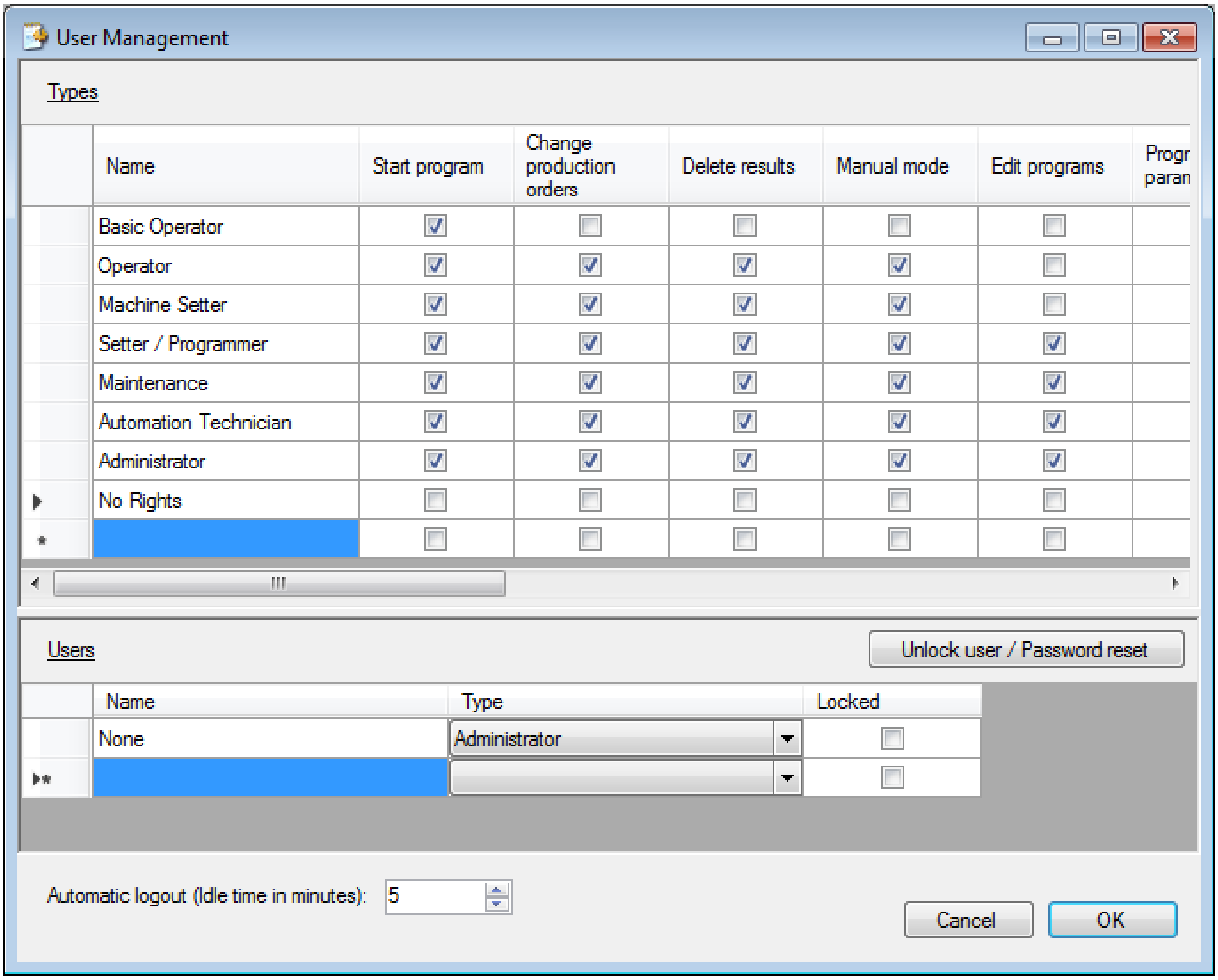


Figure 1: User management

If you want to add users, you must first create an administrator user. In the example below (figure 2), we have created an "administrator" user whose name is "Admin".

Once you have created an administrator user, you can choose that when no user is logged in, no MecaMotion functions are available, to do this, choose the type "No rights" for the user "None".

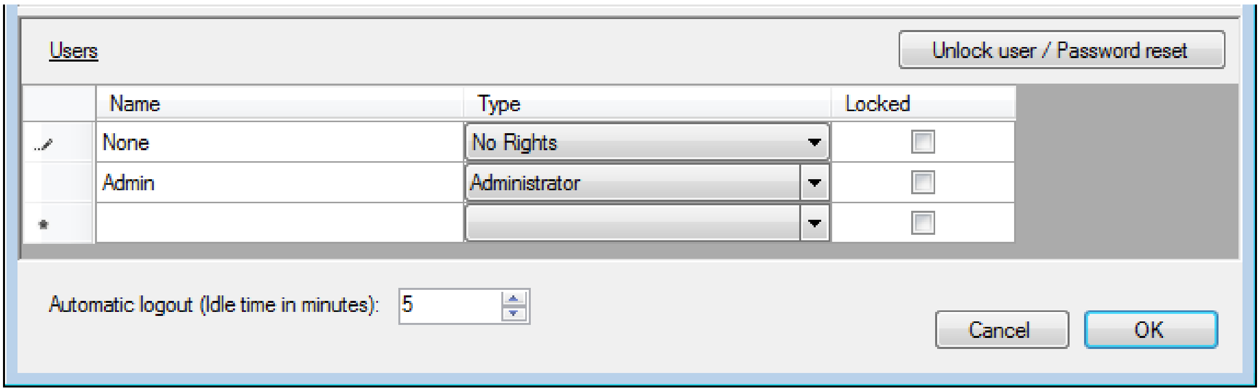


Figure 2: Creating an administrator user

To assign a password to the user you have just created, close the user management window by clicking on "OK", click on "?" in the menu bar then "Login". (figure 3)

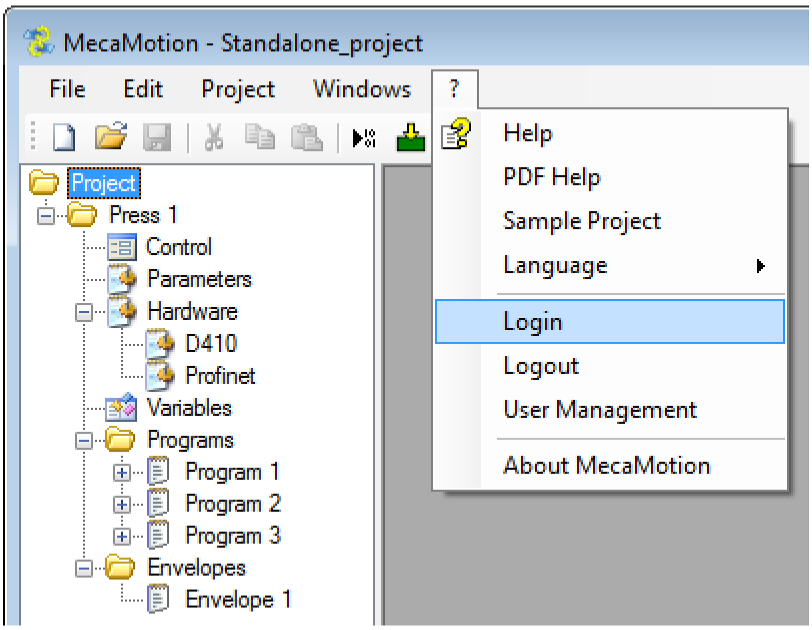


Figure 3: User login

Enter the name of the user you just created in the "User" field and click on "Login". You can leave the "password" field blank.

A new window opens (figure 4), in which, you must enter a password for this user. When you have entered the password, you will automatically be logged in as this user, to logout you must, click on "?" in the menu bar, then "Logout".

You can view the logged in user at any time at the bottom right in the main window.

Be careful: When you create an administrator user, make a note of the password you entered, because if you lose it, you will no longer be able to access the user management window.

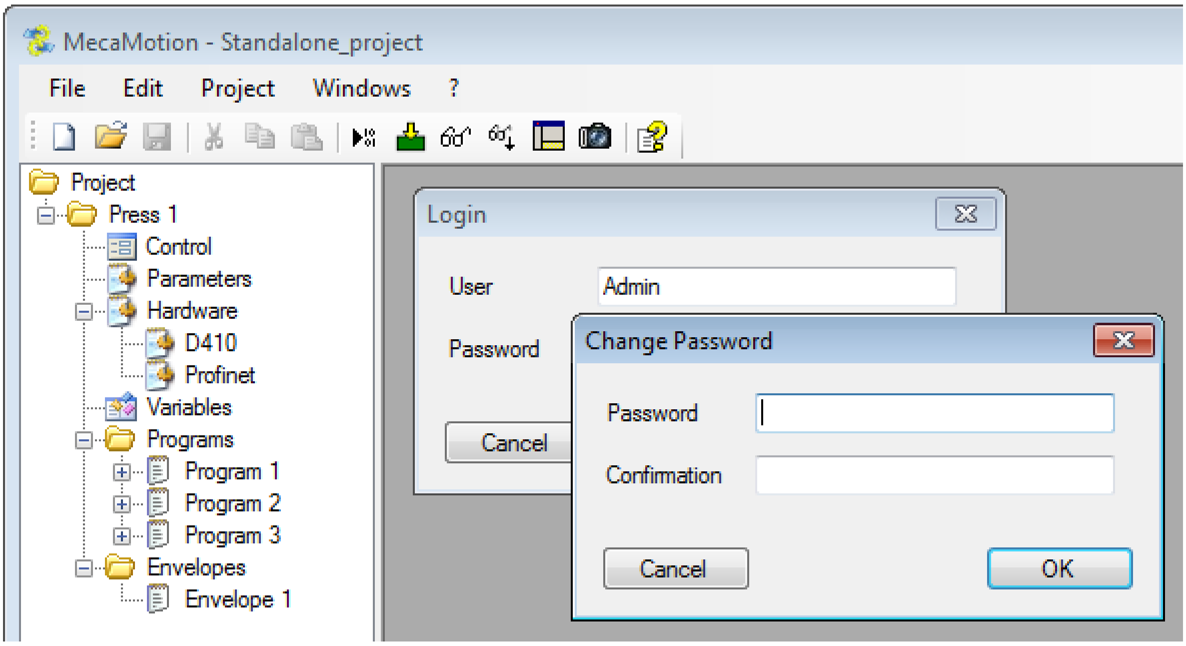


Figure 4: Creating a password

Several types of users have already been created, but you can modify them or create new ones.

In the figure 5 below, we have created the user type "Technician", we give all the rights to it except the modification of production orders and the deletion of results.

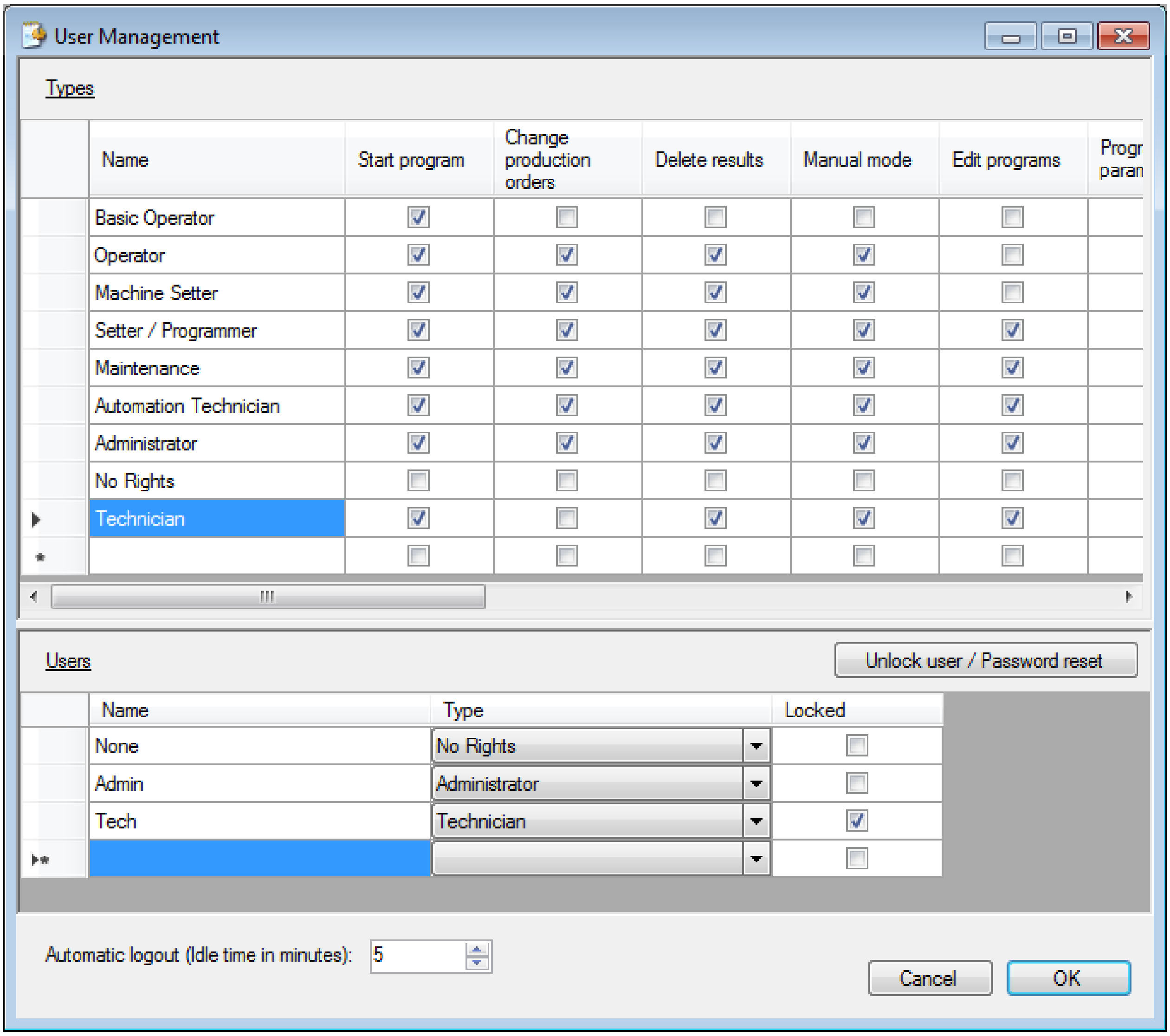


Figure 5: Creating a user type

You can choose the time after which the logged-in user is automatically logged out if he is inactive. Enter a time in minutes in the "Automatic Logout" field at the bottom of the user management window.

If you forget a user's password, you can unlock this user and reset his password from the user management window by logging in as an administrator.

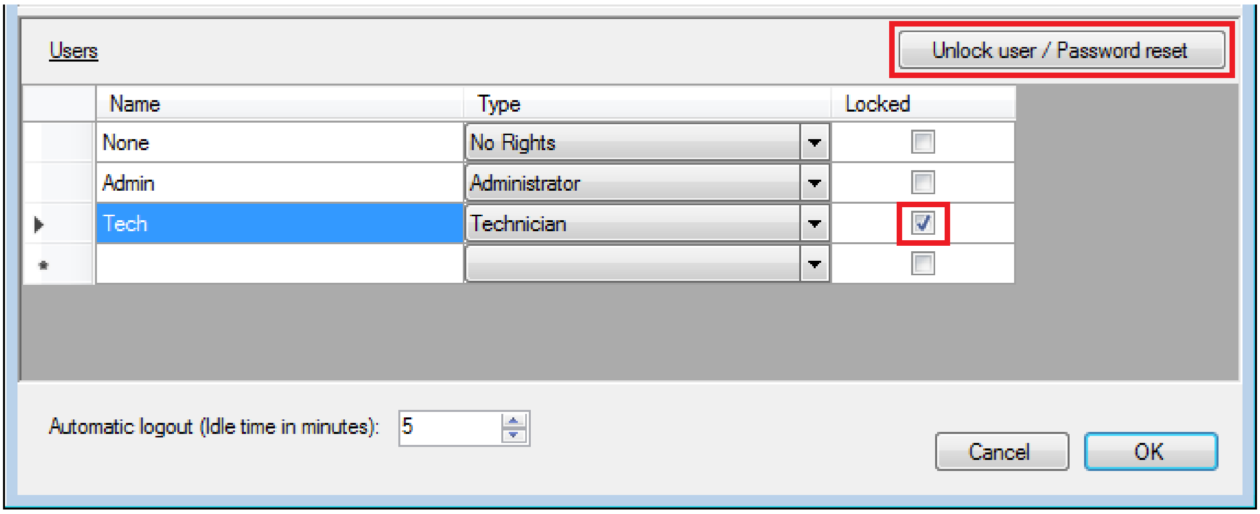


Figure 6: Password reset

Declaration of user variables

A user variable is, as its name suggests, a variable created by the user. The user variables can then be used in the various part-programs or linked to physical or PROFINET inputs/outputs.

All created variables are global to the entire platform. This means that each variable can be used in any part-program.

There are 6 variable formats, these formats are described below:

Name	Format	Coded on ... [bits]	Min and Max value
Floating point number type LONG	LREAL	64 [bits] floating	-1.797_693_134_862_315_8E+308 to -2.225_073_858_507_201_4E-308, 0.0, +2.225_073_858_507_201_4E-308 to +1.797_693_134_862_315_8E+308
Floating point number	REAL	32 [bits] floating	3.402_823_466E+38 to - 1.175_494_351E-38, 0.0, +1.175_494_351E-38 to +3.402_823_466E+38
Double integer	DINT	32 [bits]	-2147483648 to 2147483647
Integer	INT	16 [bits]	-32769 to 32767
Bit	BOOL	1 [bit]	0 or 1
Time	TIME	32 [bits]	-2147483648 to 2147483647

The value contained in a Time variable represents a time in [ms].

Example:

If a TIME variable contains the value 75, when passing from this variable to the press in a part-program, the press will understand 75[ms].

There are 15 other variable formats, these formats are specific to some instructions.

There are also tables of 20 variables of type REAL, LREAL and DINT used by the instruction "Recording values in a table".

Declaration of a variable

To declare a variable, in the project tree of the MecaMotion, double-click on the "Variables" tab. The window of figure 1 opens.

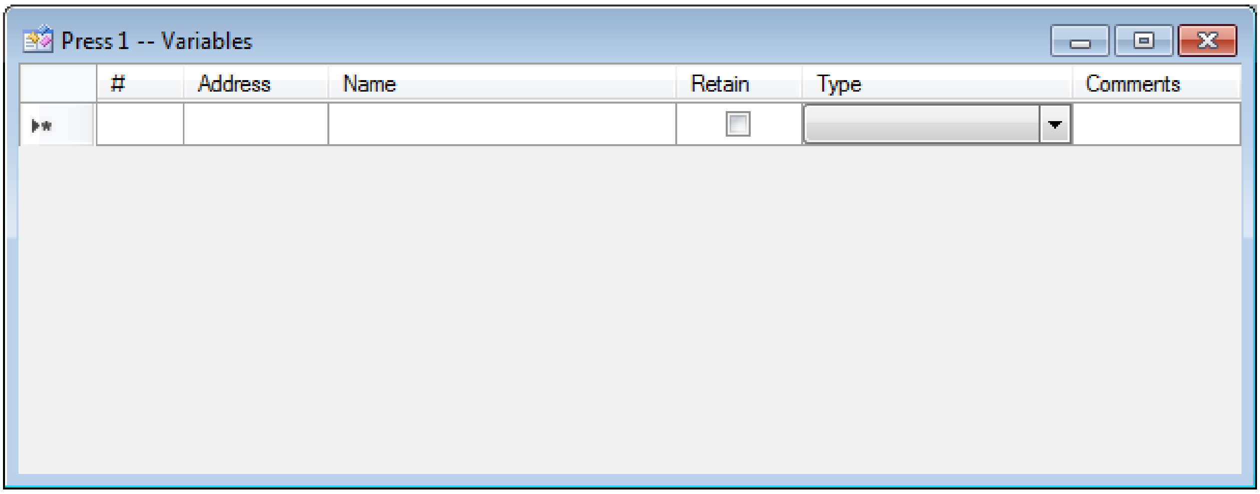


Figure 1: Variable declaration window

It is in this window that the different variables used in part-programs are declared.

To declare a variable:

- The first thing to do is to fill in the name of the variable, it is by this name that the variable will be called in the different part programs.
- When you have entered the name of the variable, you must choose the variable format by opening the "Types" drop-down menu.
- As soon as the variable format is defined, the "N°" and "Address" fields will be automatically filled in.

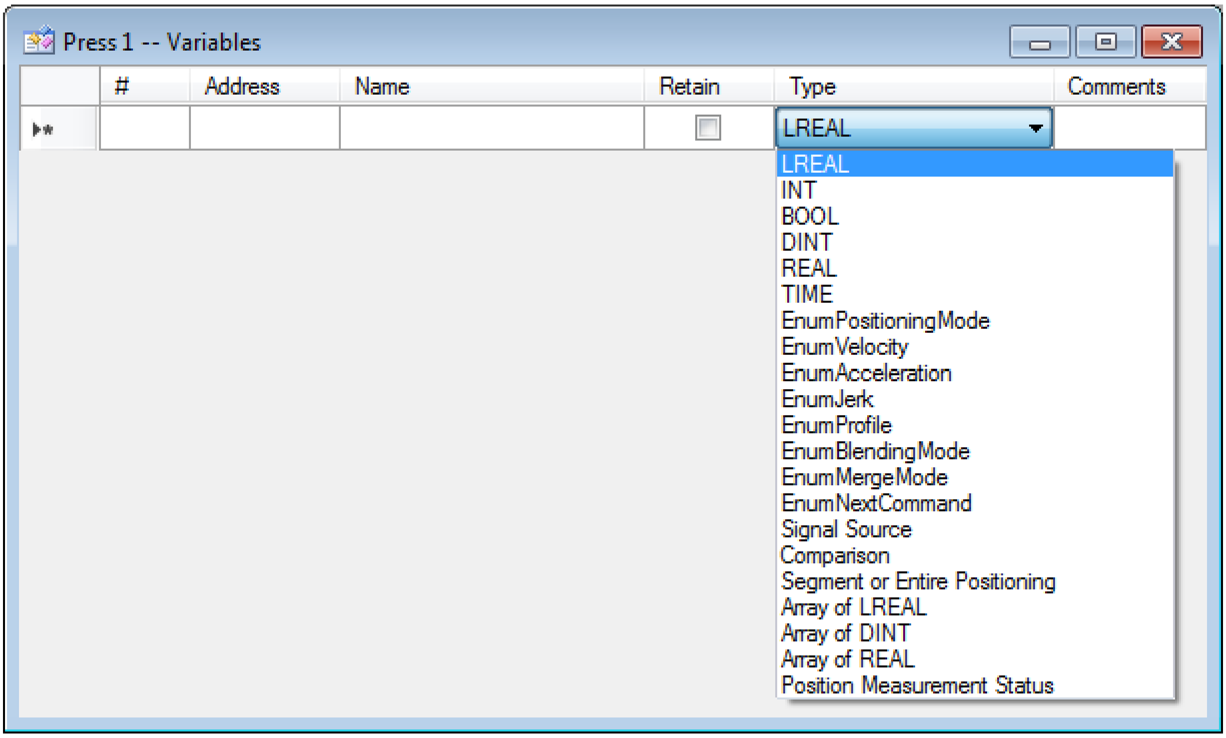
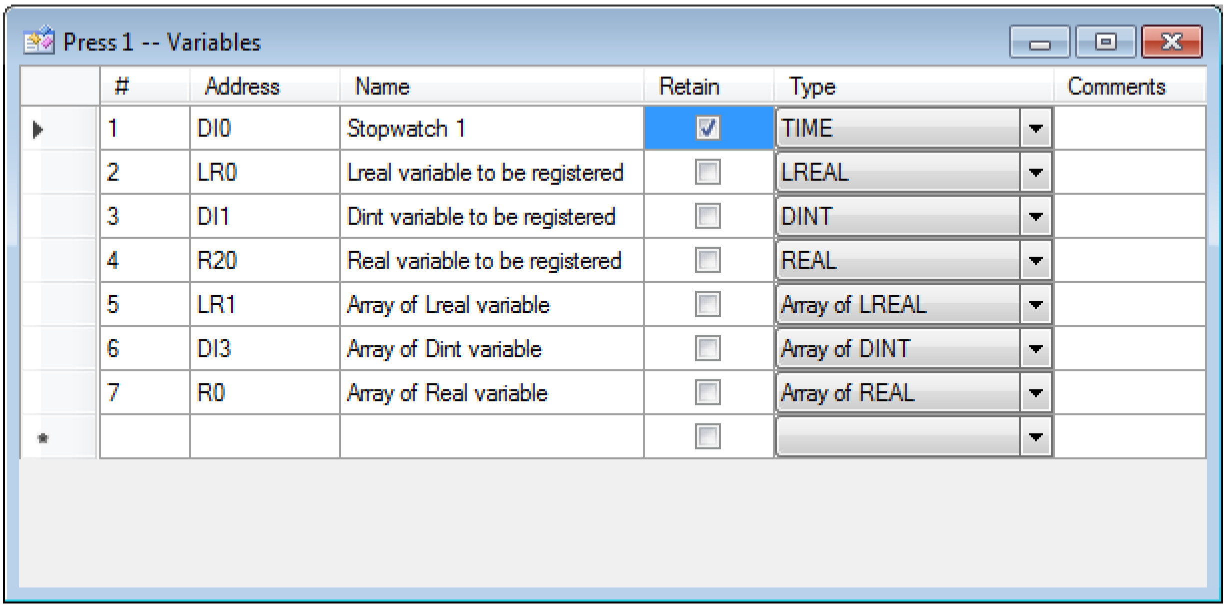


Figure 2: Declaration of a variable

Each variable can be declared as retain, this allows the value of the variable to be kept in memory when the press is switched off.



#	Address	Name	Retain	Type	Comments
1	DI0	Stopwatch 1	☑	TIME	
2	LR0	Lreal variable to be registered	☐	LREAL	
3	DI1	Dint variable to be registered	☐	DINT	
4	R20	Real variable to be registered	☐	REAL	
5	LR1	Array of Lreal variable	☐	Array of LREAL	
6	DI3	Array of Dint variable	☐	Array of DINT	
7	R0	Array of Real variable	☐	Array of REAL	

Figure 3: Declaration a retain variable

Association of user variables to physical inputs/outputs

It is possible to associate user variables (variables that are used in part programs) to physical inputs or outputs of the D410-2 control unit.

In Table 1, you find all the digital inputs and outputs of the D410-2 control unit that can be used to send or receive signals from outside.

Direction (from the press)	Input/Output number	Physical terminal	Format of the associated variable
IN	DI0	X121.1	BOOL
IN	DI1	X121.2	BOOL
IN	DI2	X121.3	BOOL
IN	DI8	X121.7	BOOL
IN	DI9	X121.8	BOOL
OUT	DO10	X121.10	BOOL
OUT	DO11	X121.11	BOOL
OUT	DO15	X131.5	BOOL
IN	DI18	X120.6	BOOL
IN	DI19	X120.7	BOOL
IN	DI20	X120.9	BOOL
IN	DI21	X120.10	BOOL
IN	DI22	X121.1	BOOL

Table 1: List of physical inputs/outputs that can be used on the D410-2

In addition to these Boolean inputs/outputs, there are three other inputs that can be associated to variables. These inputs are of LREAL type and must be associated to variables of the same type.

Direction (from the press)	Input/Output number	Physical terminal	Format of the associated variable
IN	Actual measuring sensor position value	--	LREAL
IN	Actual force value	--	LREAL
IN	Actual axis position	--	LREAL

Table 2: List of LREAL inputs

The inputs "Actual measuring sensor position" and "Actual force value" are inputs that are already connected to the measuring sensor and the force sensor respectively. The "Actual axis position" input is not a physical input, but an internal value of the control unit.

When associated to a variable, these three values can be read from the part-program.

Create the association of a variable to a physical input/output

Before a user variable can be associated to an input or output of the D410-2 control unit, this user variable must be created (see the chapter "Declaration of user variables").

As soon as the variable is created, please follow the steps below to associate this variable to a physical input/output of the D410-2.

1. In the project tree, open the "Hardware" tab and double-click on the "D410" tab. The window below opens.

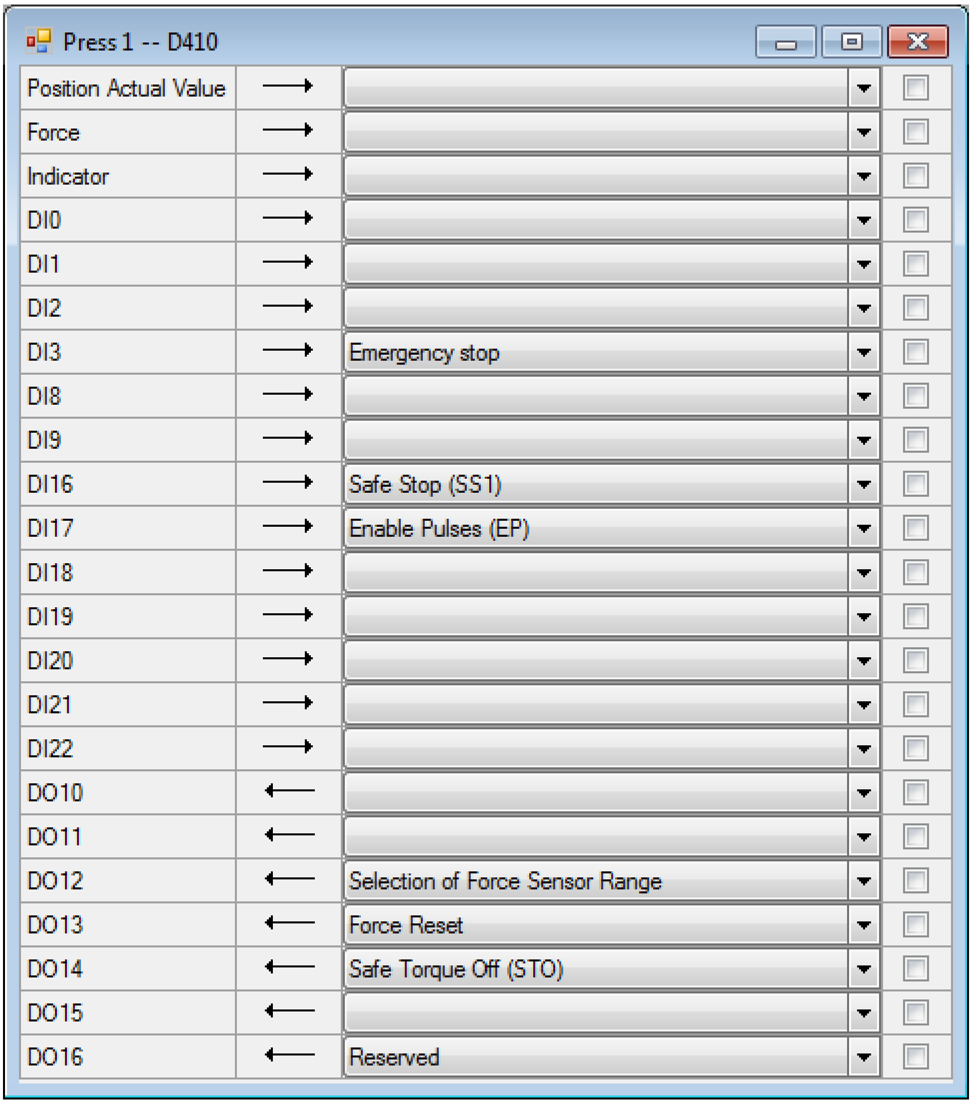


Figure 1: Window for associating variables to inputs/outputs of the D410-2

2. Select the input or output of the D410-2 to be associated to the user variable. To do this, click in the "CheckBox" of this physical input/output. (see figure 2 below).

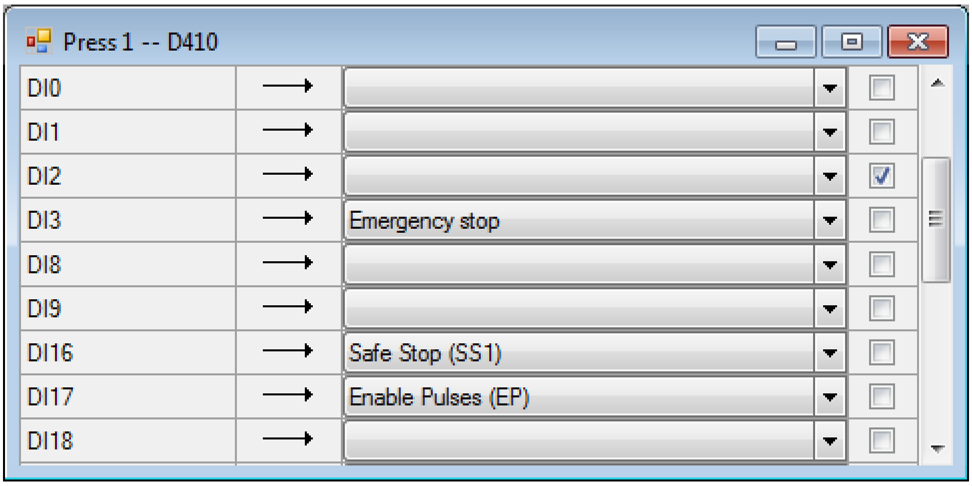


Figure 2: Selection of input 2 (DI2)

3. Use the drop-down list to select the variable to be associated to this input/output.

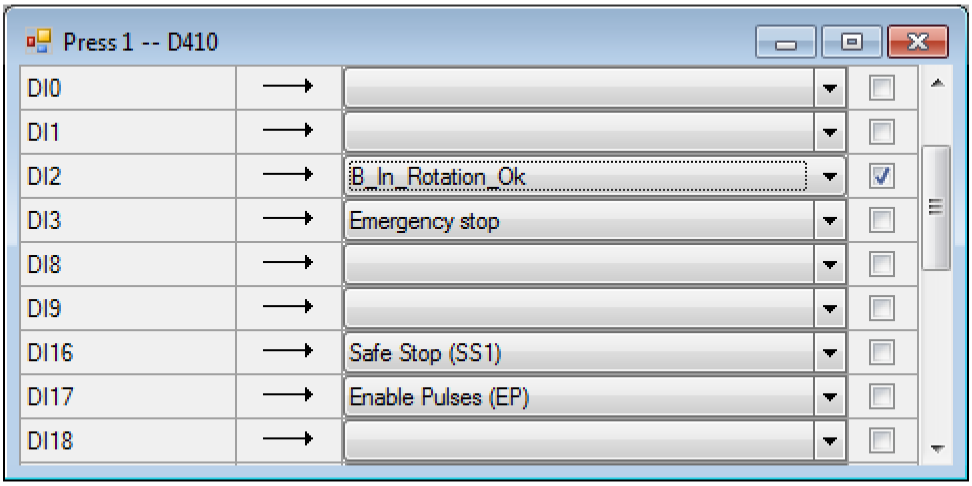


Figure 3: Selecting the variable to be associated

4. For the association to be effective, load the project into the D410-2 control unit .

Example of use:

We want to check that the final insertion force of a part is greater than 65[N].

- *If "Final Force" > 65[N] => a green lamp connected to the DO10 output of the D410-2 lights up.*
- *If "Final Force" < 65[N] => a red lamp connected to the DO11 output of the D410-2 lights up.*

To do this, you must first associate two Boolean variables to the physical outputs DO10 and DO11. (See Figure 4)

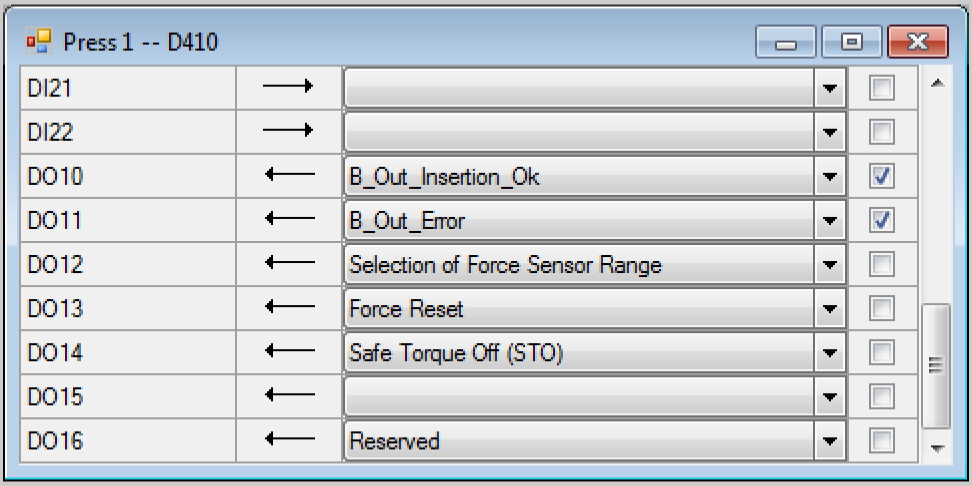


Figure 4: Association of variables to the outputs of the D410-2 CPU (lamps can be connected to these outputs)

Then, in a program (figure 5), you must test if the value of the final force is greater than 65[N], and depending on the result, set the "Insert OK" or "Error" variables to "1" to switch on the lamps.

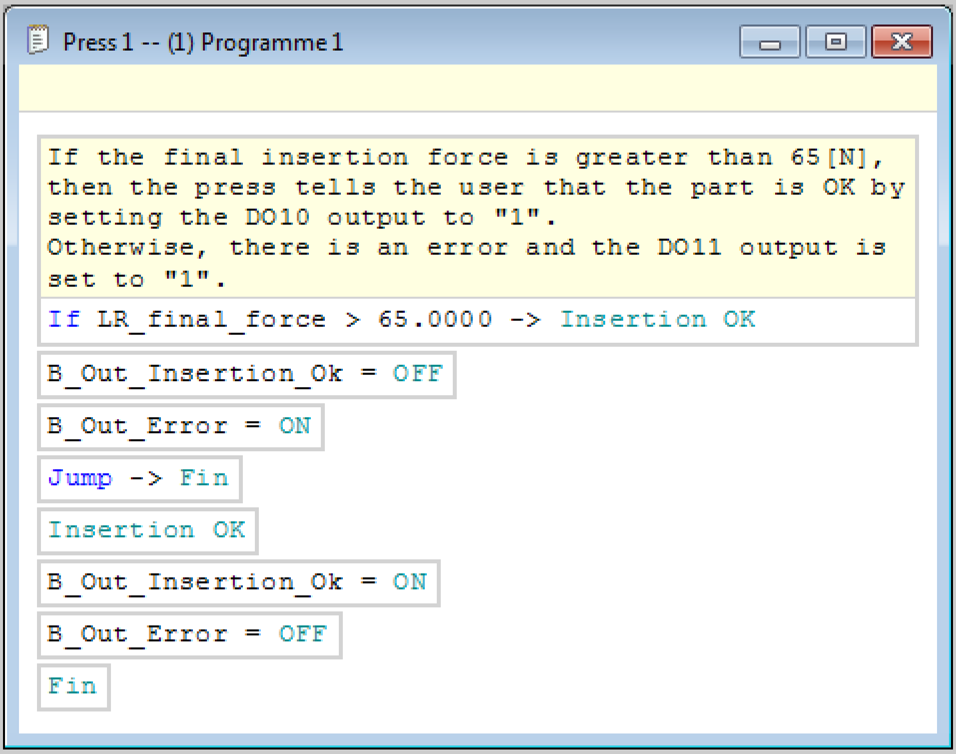


Figure 5: Example of a program testing whether the final force > 65[N]

Association of user variables to PROFINET inputs/outputs

The PROFINET link allows you to send setpoints and parameters to the press from a PLC (For example: Siemens S7-1500 PN/DP) and the press can return data and results to the PLC. The reception and sending of this data can be done by the HMI of your choice.

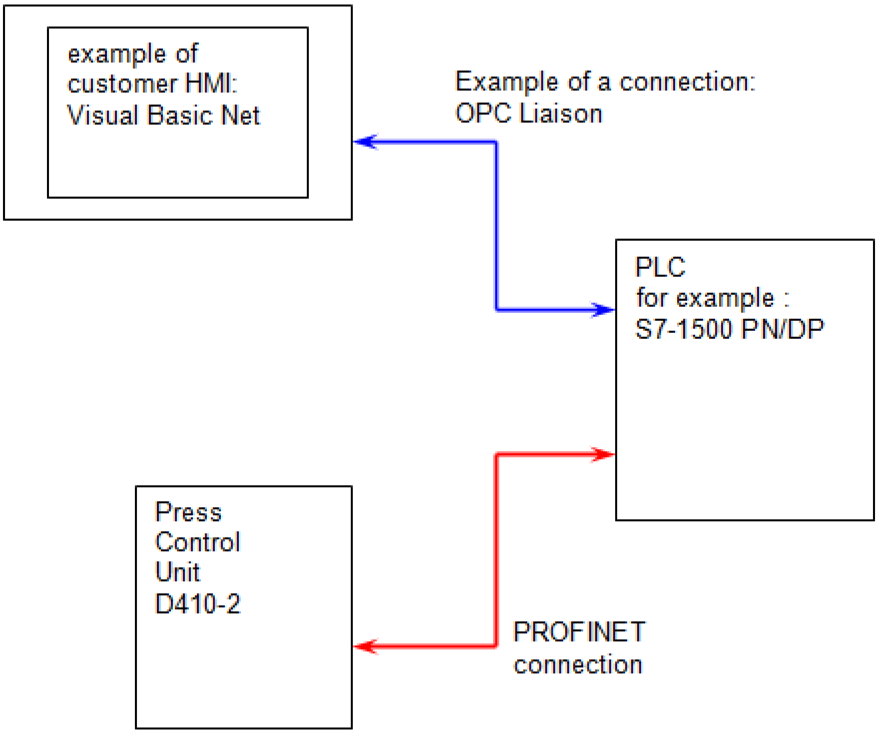


Figure 1: Customer HMI connection to the universal press

Example of using the association of variables to PROFINET inputs/outputs:

We would like to insert a component.

We can send to the press via the PROFINET connection, the insertion force set point, the insertion preposition etc...

At the end of the cycle, the press can return us, the final insertion force, the number of iterations and other results in LREAL format.

Note:

This association of user variables to PROFINET inputs/outputs allows you to modify and read the variable values used in the press part-programs from a PLC.

There are two types of PROFINET inputs/outputs:

- BOOL types: Transmission of values "0" (FALSE) or "1" (TRUE)

- Type DWORD (Double Word): Transmission of values of type LREAL, REAL, DINT

As explained in the "Overview of the PROFINET dialogue" topic, there are 32 possible *associations* for type BOOL and 50 possible *associations* for type DWORD.

Create the association of a variable to a PROFINET input/output

In order to associate a user variable to a PROFINET input or output, this user variable must first be created (see the topic "Declaration of user variables").

As soon as the user variable is created, please follow the steps below to associate the user variable to a PROFINET input/output.

1. In the project tree, open the "hardware" tab, then double-click on the "Profinet" tab, then the window for associating variables to the PROFINET bus will be opened (see figure 2). Note: It is possible to display the inputs/outputs by number or address, if you display them by address, enter the starting addresses in order to have the same structure as on the Profinet master.

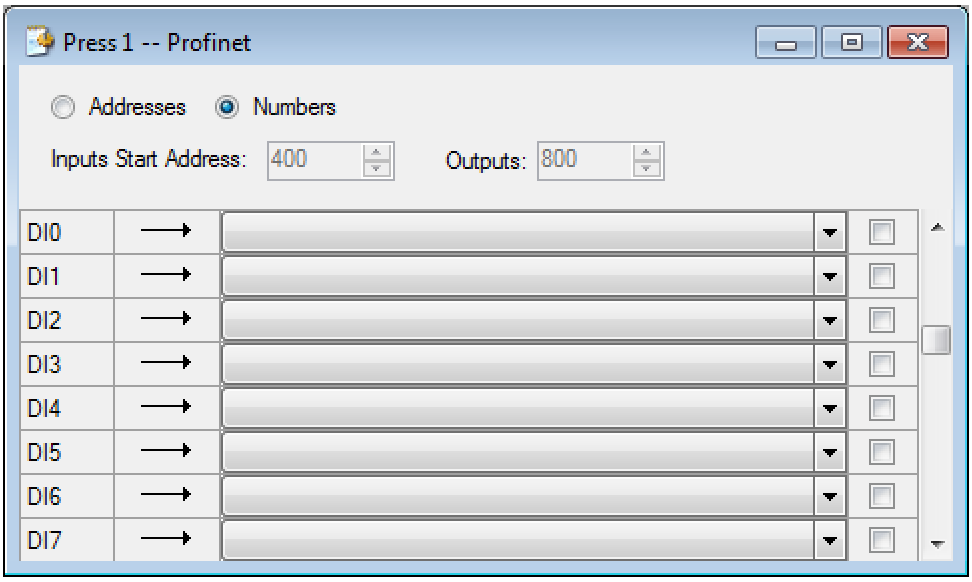


Figure 2: Window for associating variables to PROFINET inputs/outputs

2. Select the PROFINET input or output that will be associated to the user variable. To do this, click on the "CheckBox" of the input/output in question. Attention, the inputs (DWIx or DIx) and outputs (DWOx or DOx) are seen from the press side.

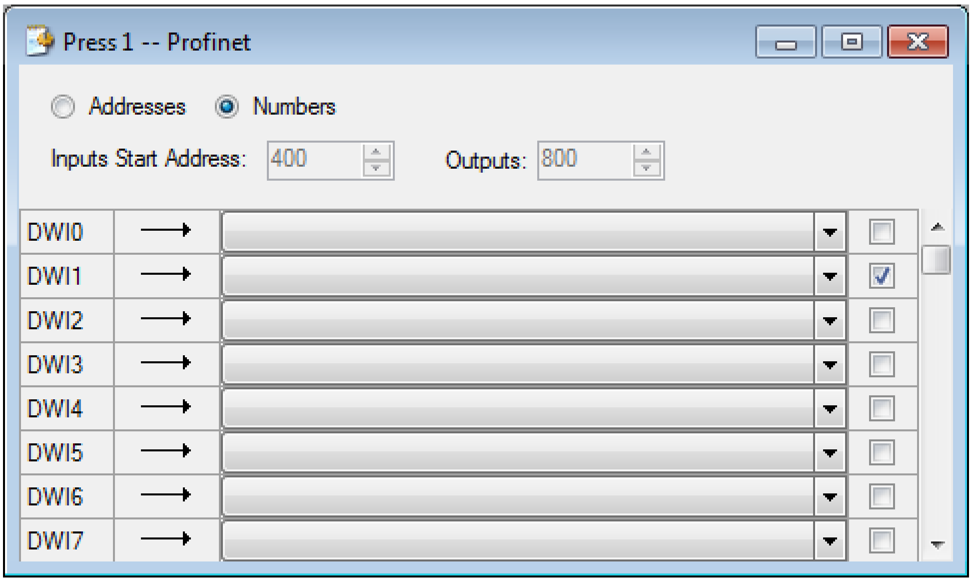


Figure 3: Selection of the PROFINET input/output to be associated

3. Select the user variable that should be associated with the PROFINET input/output in question.

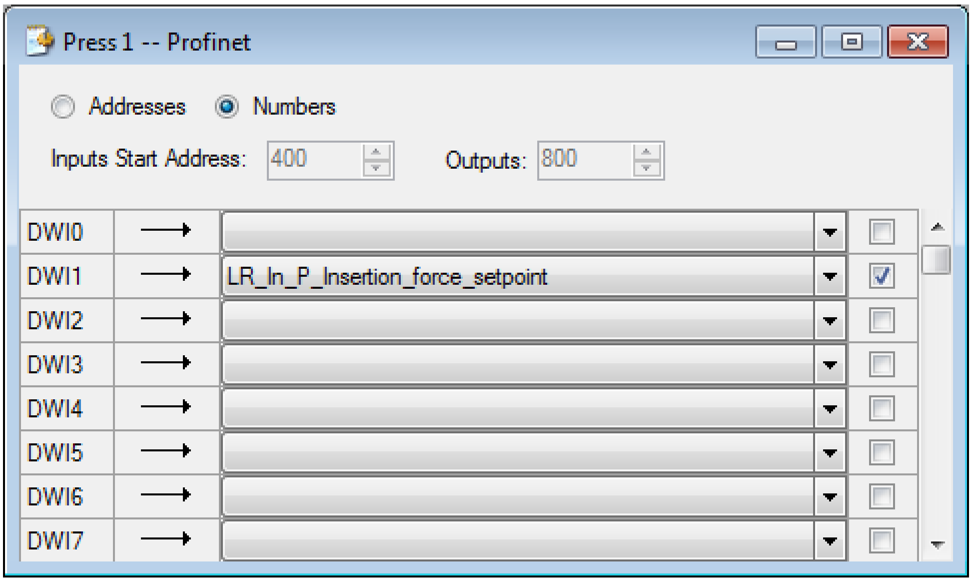


Figure 4: Selection of the variable to be associated

4. For the association to be effective, load the Project into the D410-2 control unit.

Example:

During a component insertion cycle, the force setpoint is a parameter that the operator must be able to change. This force command is sent to the press from the PLC, via a PROFINET connection.

In Figure 5, the value of the "force" of the "Stop on force" control comes from a variable called "LR_In_P_Insertion_force_setpoint". This variable is associated with a PROFINET input, so that the force setpoint can be sent from the PLC.

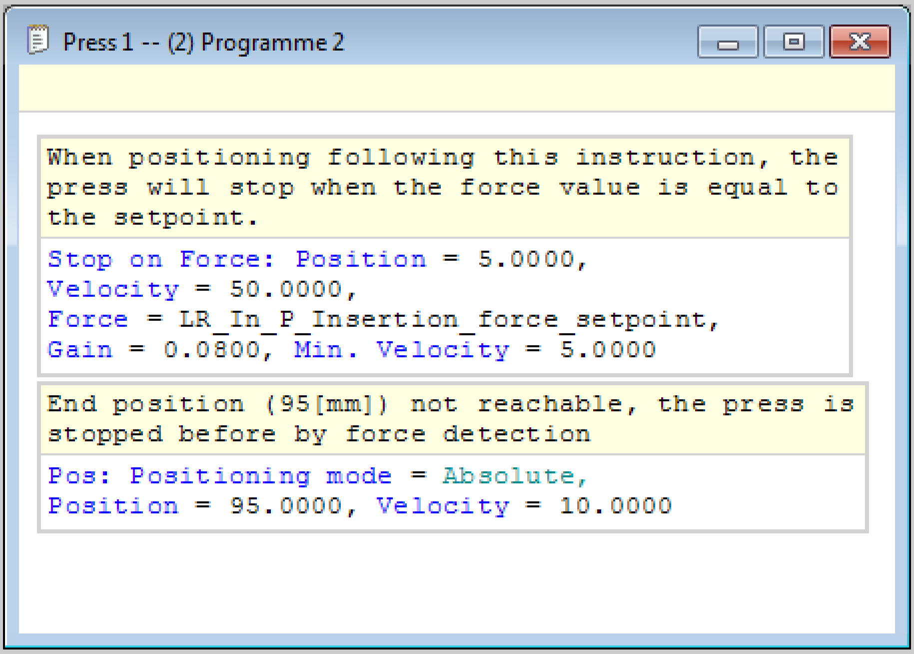


Figure 5: Exemple de programme pour insertion en force

Association of the variable "LR_In_P_Insertion_force_setpoint" with the Profinet inputs "DIW1". (figure 6)

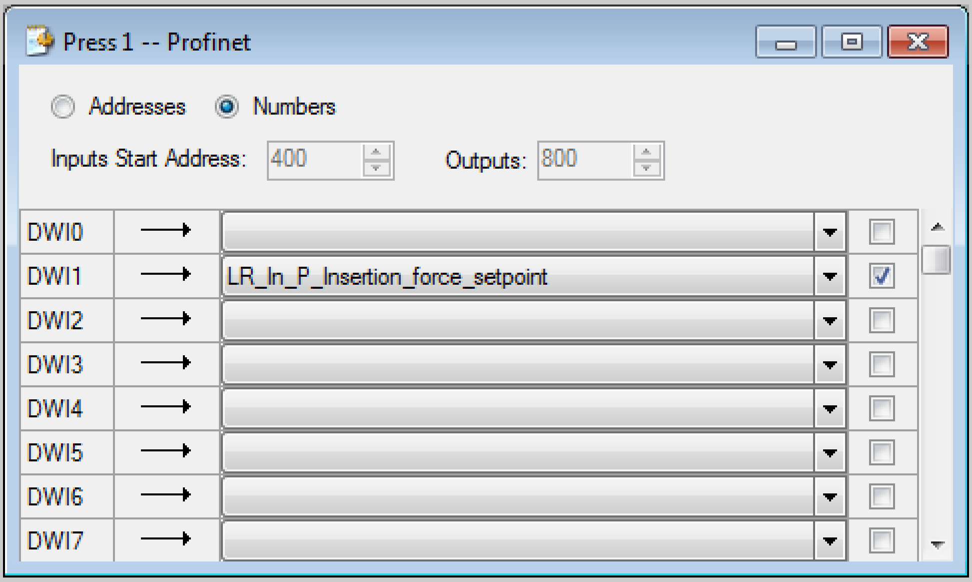


Figure 6: Association of the "Insertion force setpoint" variable with a PROFINET input

Envelope

The envelope allows the shape of the force curve to be controlled in relation to the position.

Adding a new envelope

Envelopes can be created in the project tree structure by right-clicking on the "Envelope" folder and then "Add an envelope". (see figure 1)

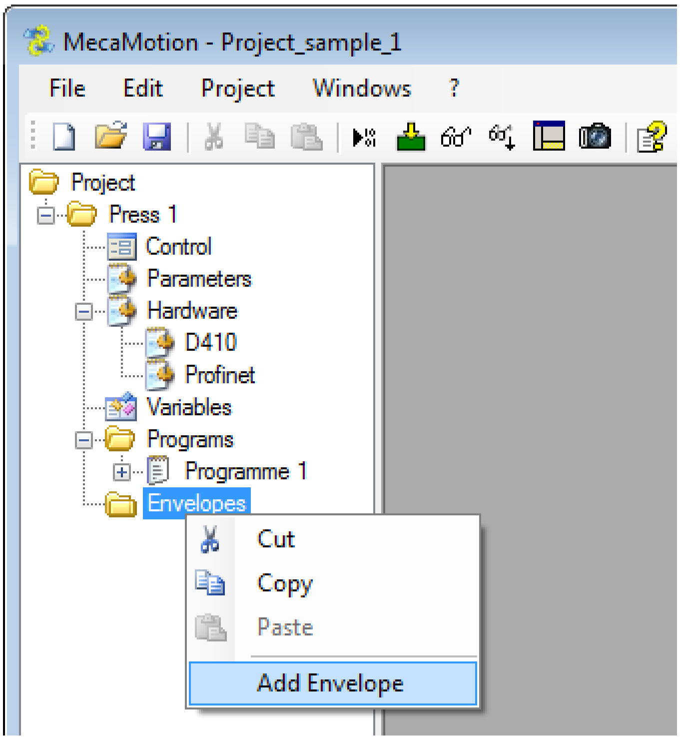


Figure 1: Adding a new envelope

To be able to place the objects of the envelope, it is essential to have one or more force/position curves reference.

To obtain them, you must perform cycles and save the curves.

See the instruction "Curve recording control" for more information on recording curves.

You can view the curves received in the "control" window. To view several curves on the screen, you must select them by clicking and dragging.

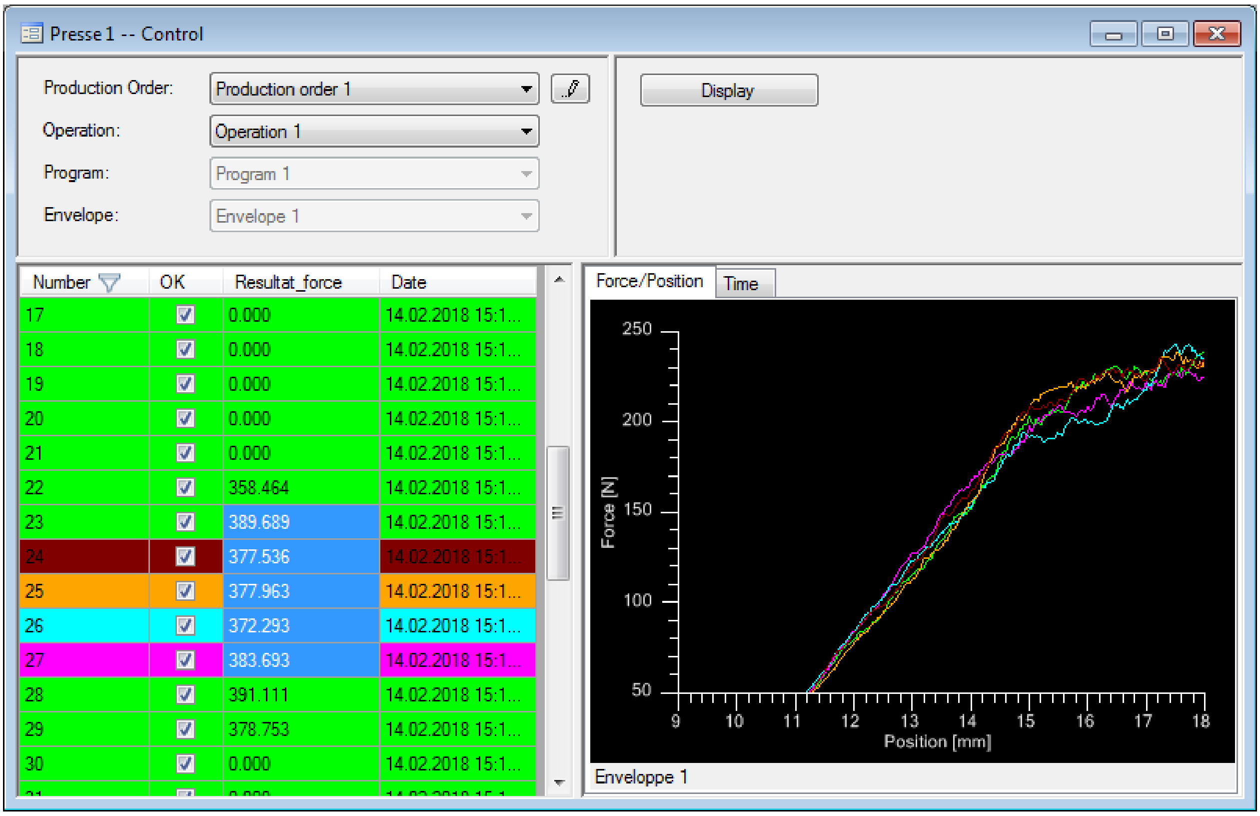


Figure 2: Retrieving reference curves

When you have enough curves to define the envelope, select them from the list on the left, using the "CTRL" or "SHIFT" key. The selected curves will then be displayed in different colors.

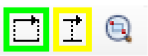
You can now right-click on the graph and then "send to envelope". You must then choose the envelope to which you want to send them.

Placing the objects in the envelope

Now open the envelope.

Select the object you want to add. Two types of objects are available:

Quadrilateral (green) / Control point (yellow)



You can place a maximum of 8 quadrilateral objects and 8 other objects. Currently the other objects are force control points and position control points.

Place the object on the graph and dimension it by moving its points, tolerances or manually entering the coordinates of its points. (see Figure 3)

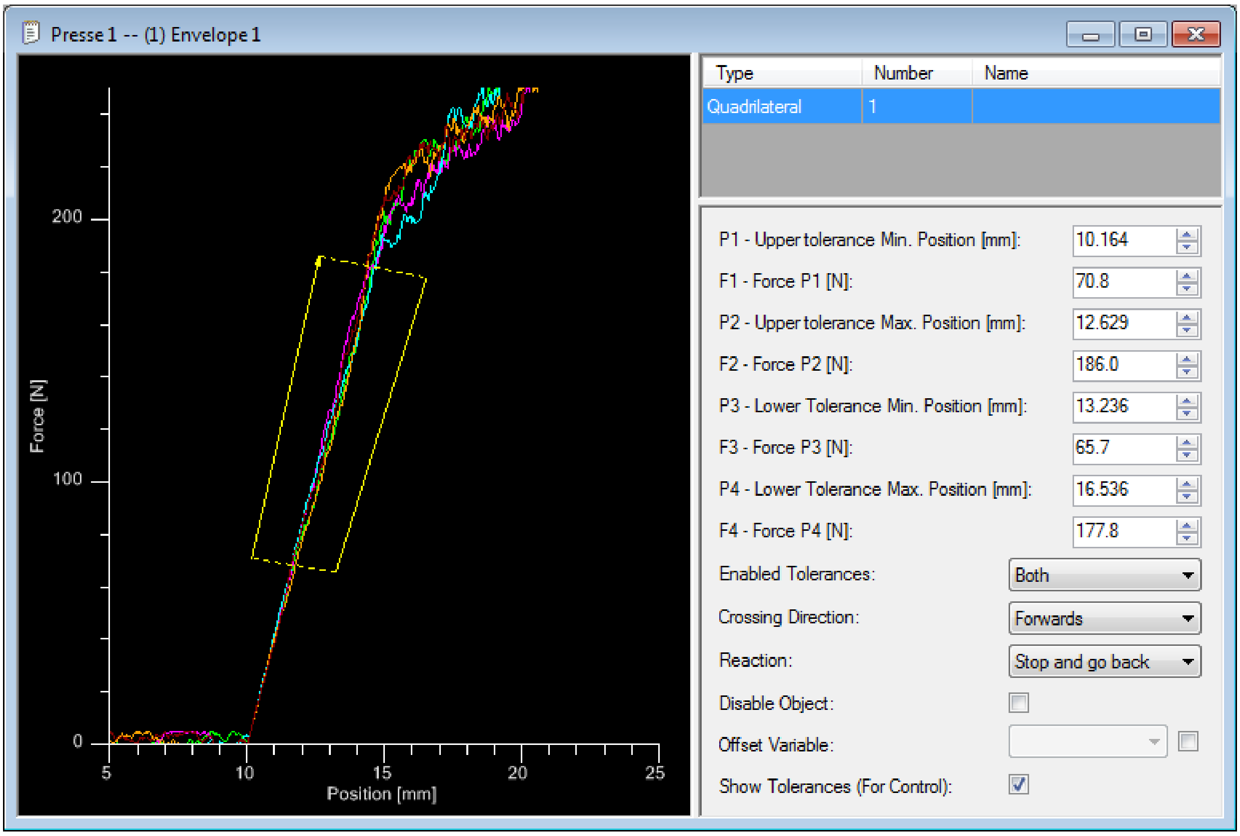


Figure 3: Placing an object in the envelope

Objects parameters

When an object is selected, you can change its settings in the right of the window (see figure 3).

Quadrilateral parameters:

- Enabled tolerances (lower, upper, or both)
- Direction of passage of the curve in the object.
- Reaction when the tolerance is exceeded (stopping the axis, stopping the axis and go back to the release position or continuing the cycle normally)
- Deactivation of the object, if the box is checked, the object will no longer be processed and it will be greyed out on the graph
- Added a position offset that allows to shift the objects of the envelope if the starting position is different
- Display the tolerances for control that allows you to return with the curve, the tolerance points that have been calculated on only one object of the envelope at a time

Recovery of curves and tolerances

To check that the curve is within the tolerance, go back to the "control" window.

Important, to receive the force/position curves, you must create a production order and an operation containing the program and envelope you are working with.

If you have requested the return of tolerances for an object, all calculated tolerance points will be displayed in red.

By right-clicking on the graph, you can show and hide the objects of the envelope.

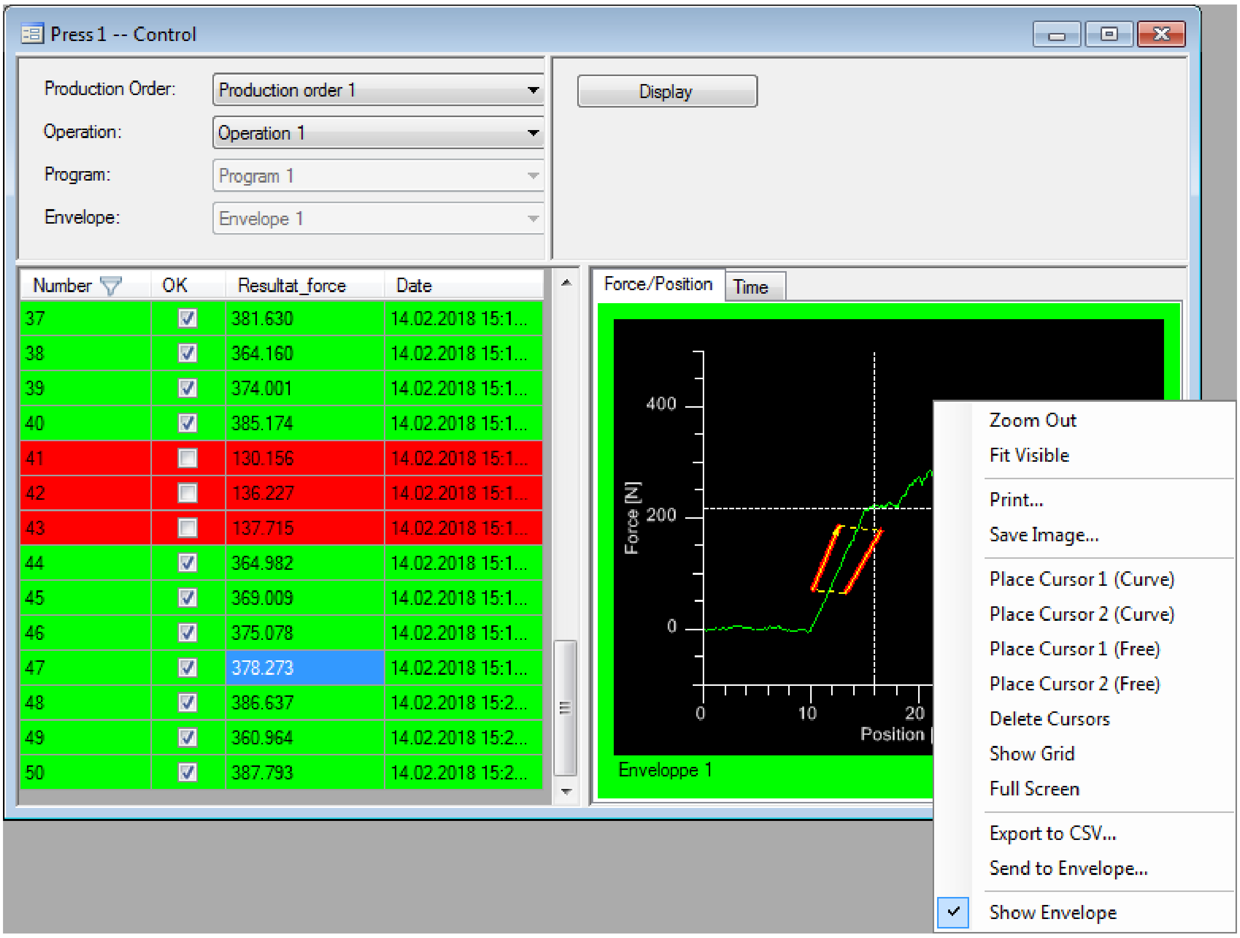


Figure 4: Visualization of the curve, envelope object and return of tolerance points

PROFINET inputs / outputs for the envelope

By PROFINET, the input byte n°241 allows you to indicate the envelope number you wish to activate.

The input bit n°240.6 is used to validate the envelope number to be activated.

If the envelope is valid, the press returns the number of the active envelope in the output byte n°251.

You can enable or disable the envelope control using input bit n°240.7 (at "1" for the control to be active).

When you validate the envelope to be activated, if it is not present in the part-program, error bit n°248.5 changes to "1".

During operation, if the curve exceeds an envelope tolerance, error bit n°248.4 changes to "1".

Online / offline project comparison

The online project comparison function allows you to visualize the differences between the active project in the press and the open project in MecaMotion.

The offline comparison allows you to visualize the differences between two projects saved on a data carrier.

The comparison is made on part programs, envelopes, input/output associations with user variables and parameters.

Using the online comparison

To carry out the comparison online, right-click on the "press" folder and then "compare online". (see figure 1)

A window opens, on which you can check the differences between the projects.

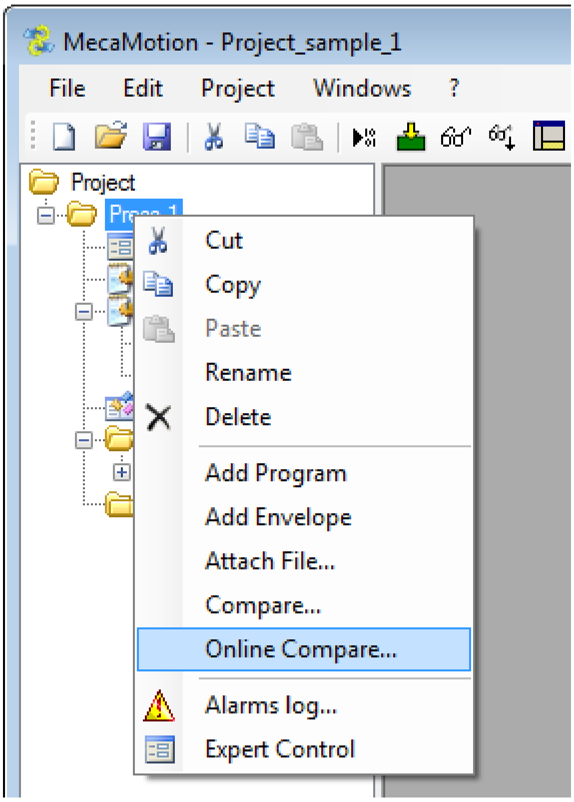


Figure 1: Compare online

Using the offline comparison

To perform the comparison offline, right-click on the "press" folder and then "compare". (see figure 2)

The windows explorer opens, in this one, choose the project to compare.

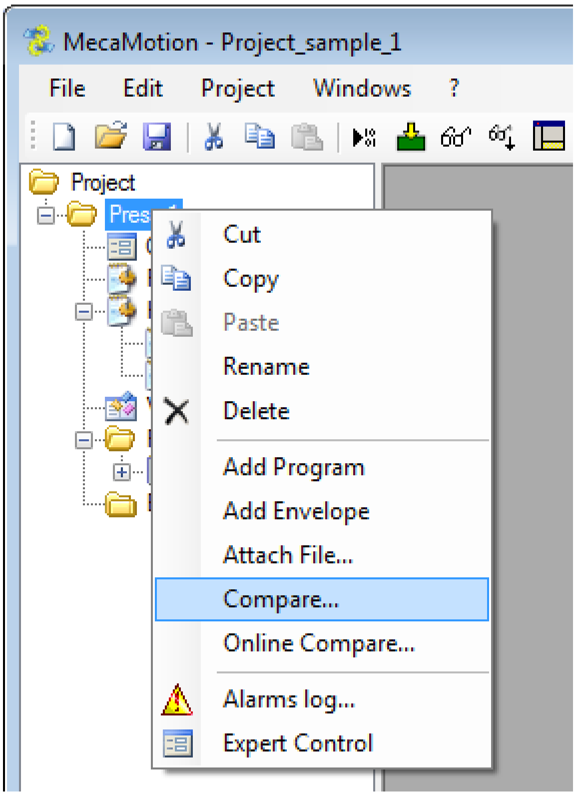


Figure 2 : Compare offline

For each part-program, instructions with different parameters are displayed in red and missing instructions in yellow. (see figure 3)

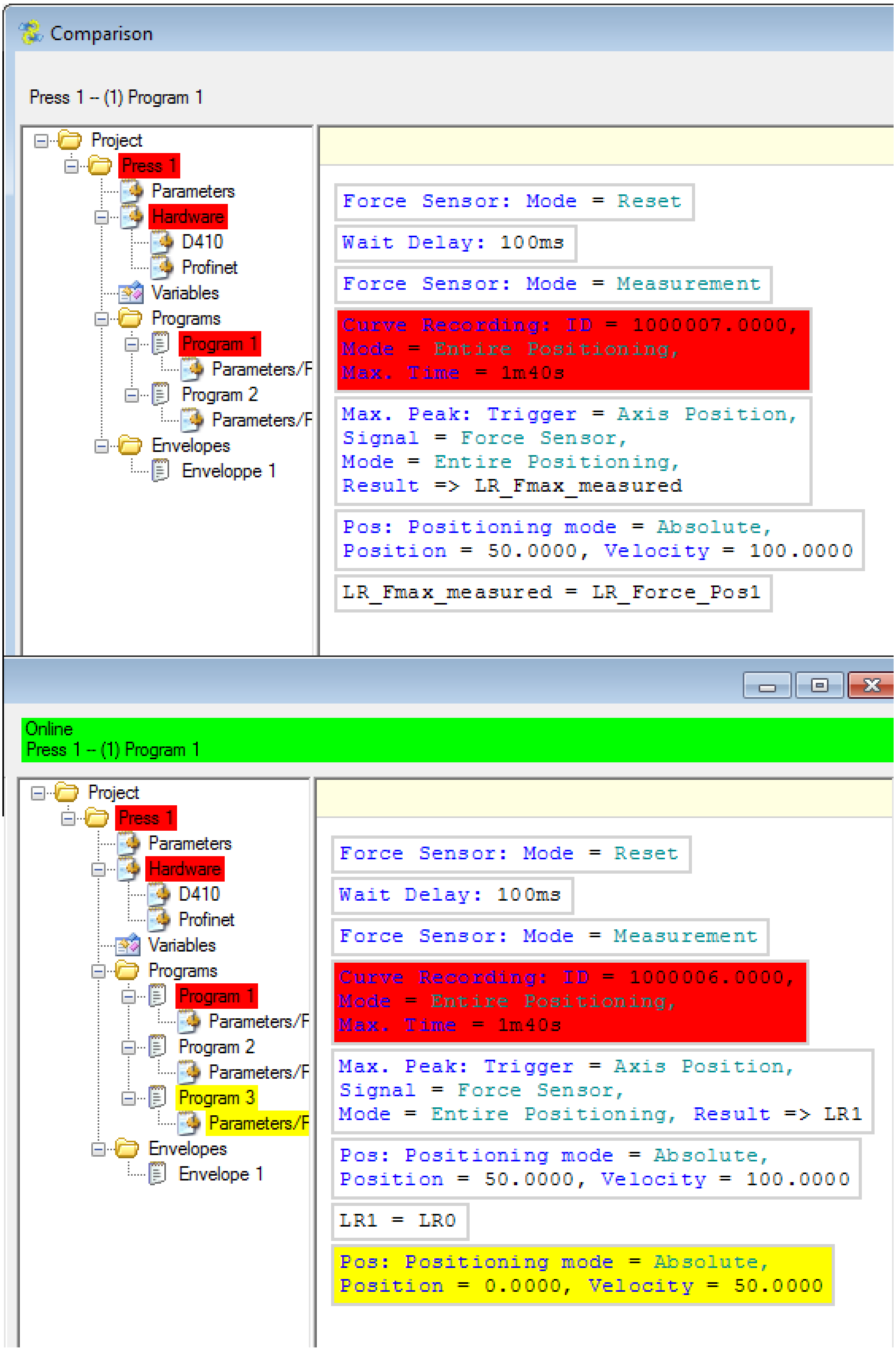


Figure 3: Comparison of part-programs

User variables that are not declared are displayed in yellow.

If you perform an online comparison, the variables not used in the part programs will not be visible in the online project and will then be returned in yellow in the offline project. (see figure 4)

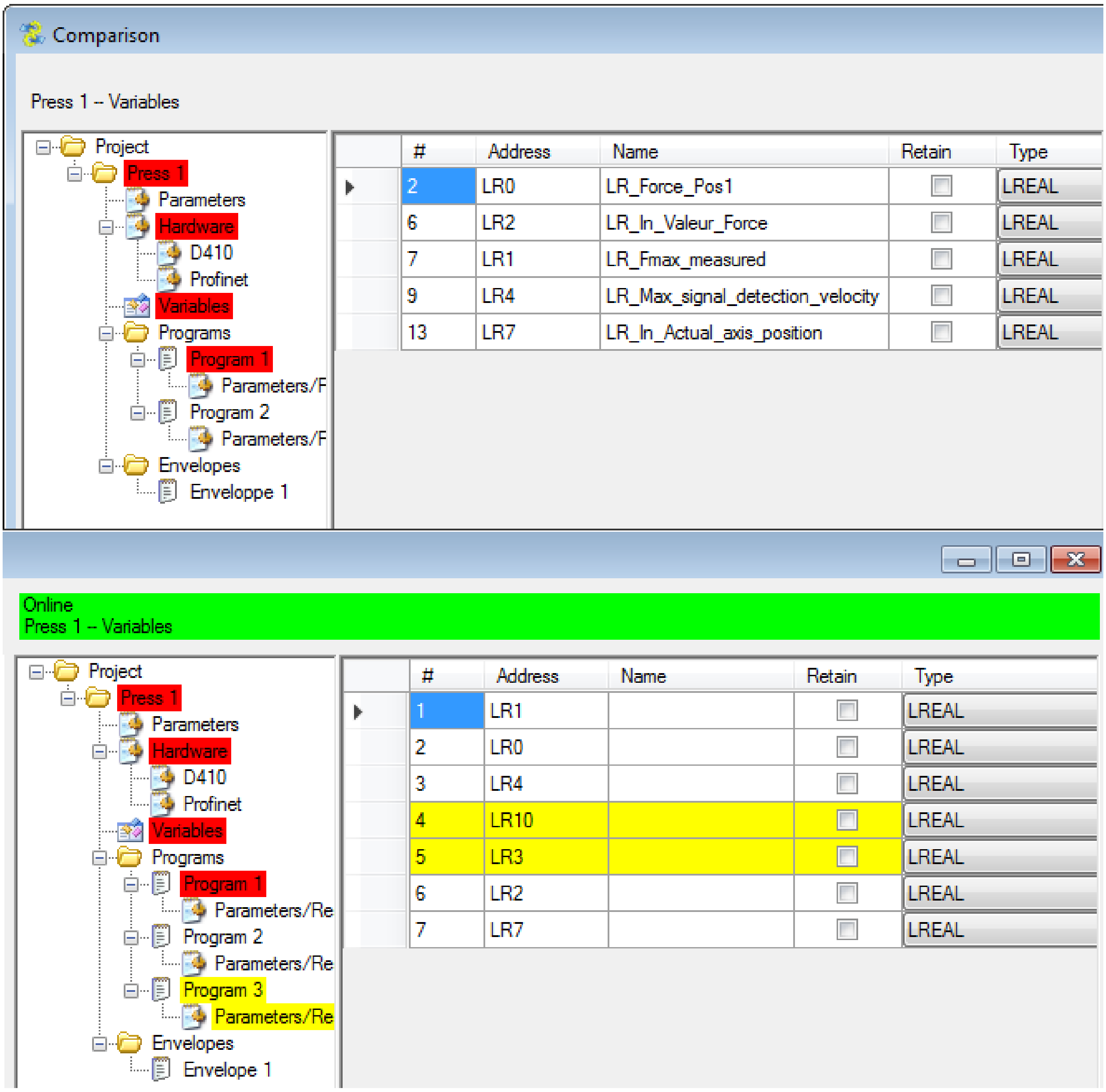


Figure 4: Comparison of user variables

Adding an attached file

If necessary, you can add a file to the project tree to access it from MecaMotion at any time.

To add a file, right-click on "press" and then "Attach file".

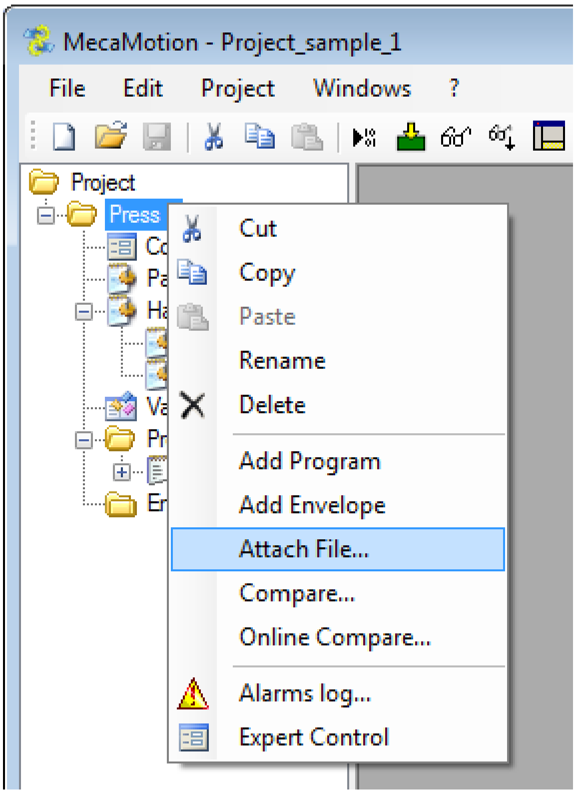


Figure 1: Adding an attached file

The added file is located at the bottom of the project tree and can be opened by double-clicking on it.

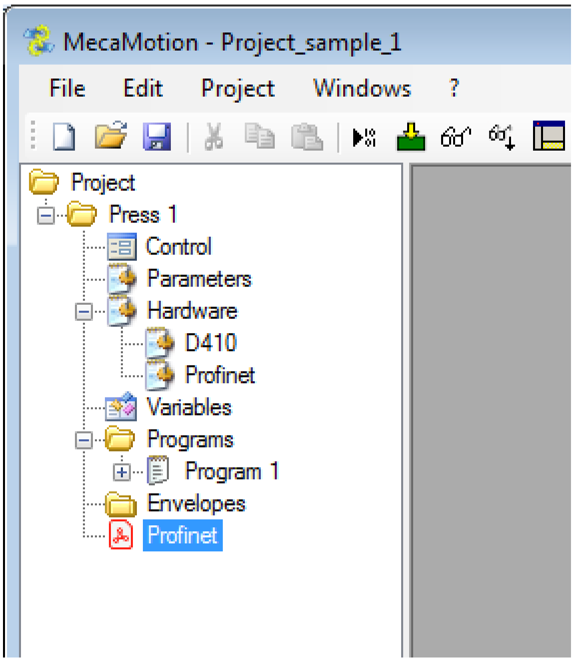


Figure 2: Adding the "Profinet" file

Simulator

The simulator allows you to test all the functions that can be used in a part-program without needing a real press.

It allows to simulate the movement of the press, the different forces involved and visualize the results obtained.

Access the simulator

The simulator can be accessed from the main menu by clicking on the icon circled in red in the figure below. (figure 1)

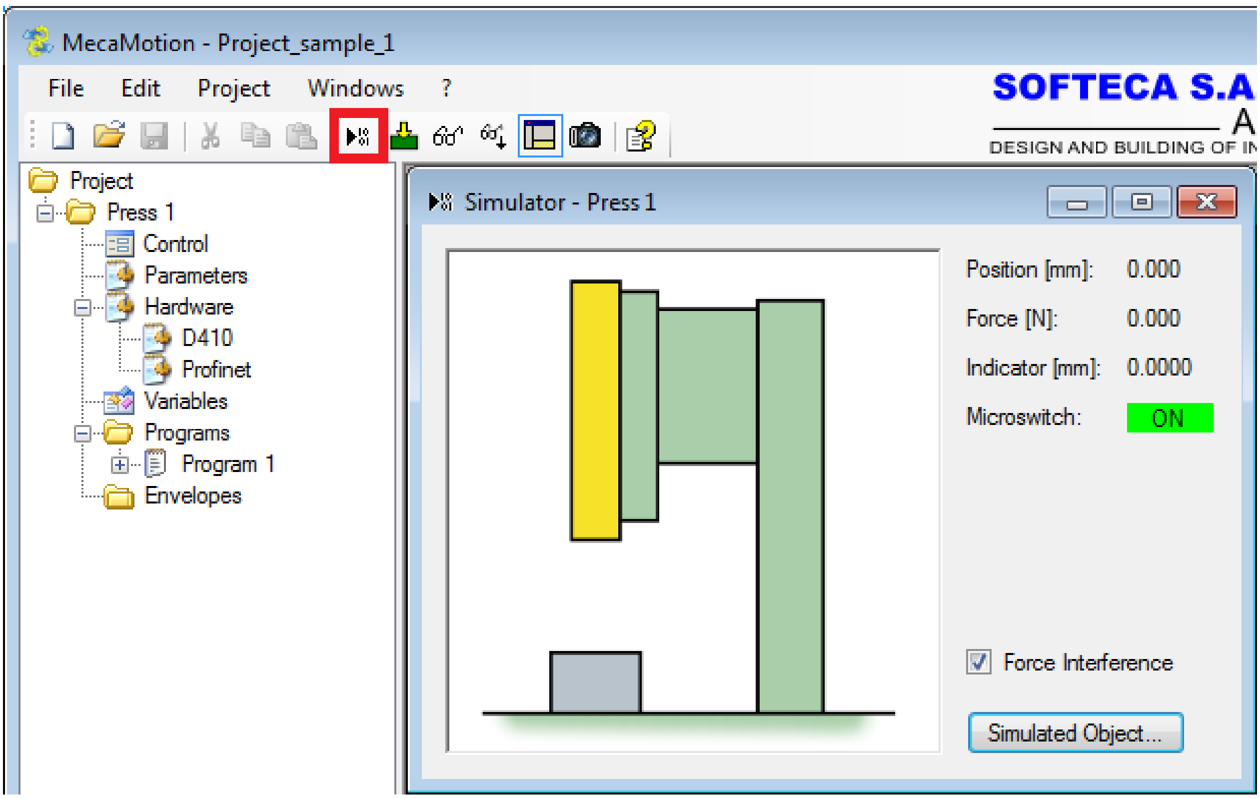


Figure 1: Simulator opening

The simulator works exactly like the real press. Once you have created the programs you want to test, you must load them into the simulator using the  button . Then you can start program cycles and view the results from the "control" window.

Simulator configuration

If you want to make the force/position curve more realistic, you can check the "Force interferences" box (Figure 1) to simulate the noise found on the actual force sensor signal.

In the simulator window (figure 1), if you click on the "simulated object" button you can define according to the position of the axis which force is applied on it, this allows you to simulate a force/position curve. (figure2)

Currently the monitoring of envelopes is not simulated, exceeding the tolerance will not result in the programmed reaction.

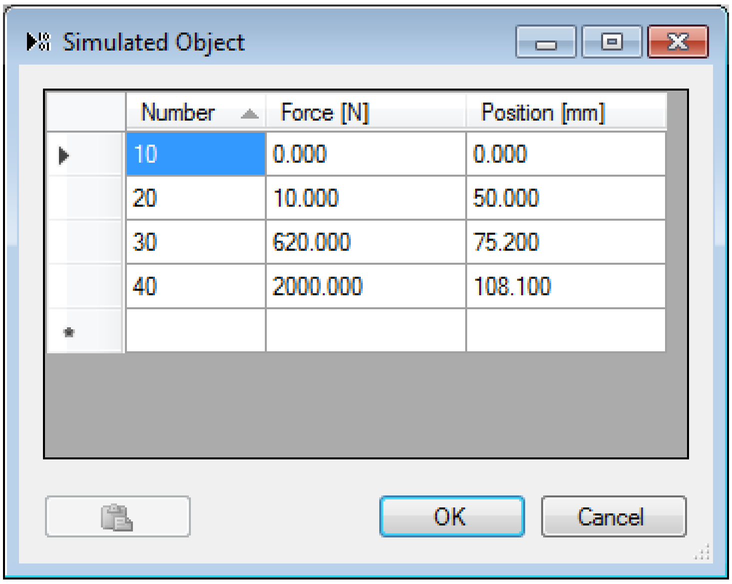


Figure 2: Simulation parameters

Visualization of program progress and variable values

When you execute a part-program, you can view its progress and view the status of user variables.

To activate the visualization of programs and variables you must select the "monitor " button (circled in red in figure 1).

Visualization of the progress of a program

When the "monitor " button is active, the press you are viewing and the program currently running are highlighted in light green in the project tree structure.

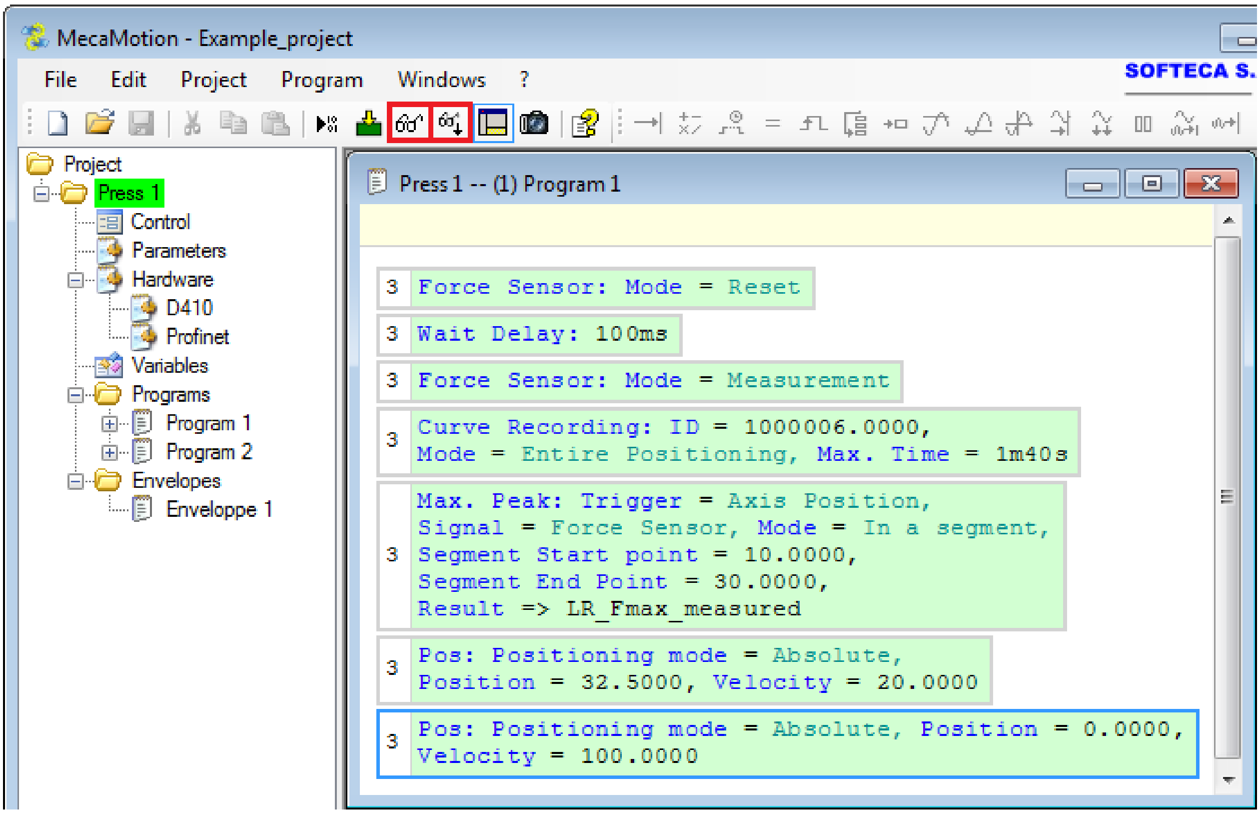


Figure 1: Enable visualization

To ensure that the instruction that is being executed is always displayed in the foreground, you must select the "follow executed instructions " button in addition to the "monitor " button. (figure 2)

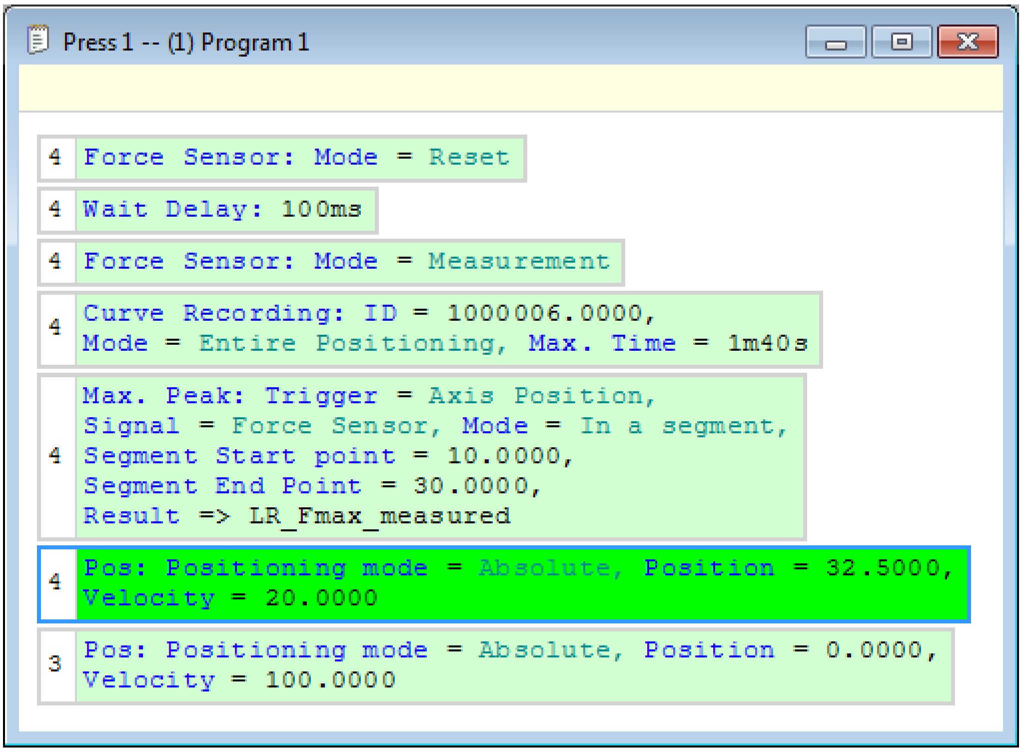


Figure 2: Visualization of the instruction is being executed

Viewing the value of user variables

When the visualization is active, you can see the status of a user variable by clicking in the "Value" field of the variable. (figure 3)

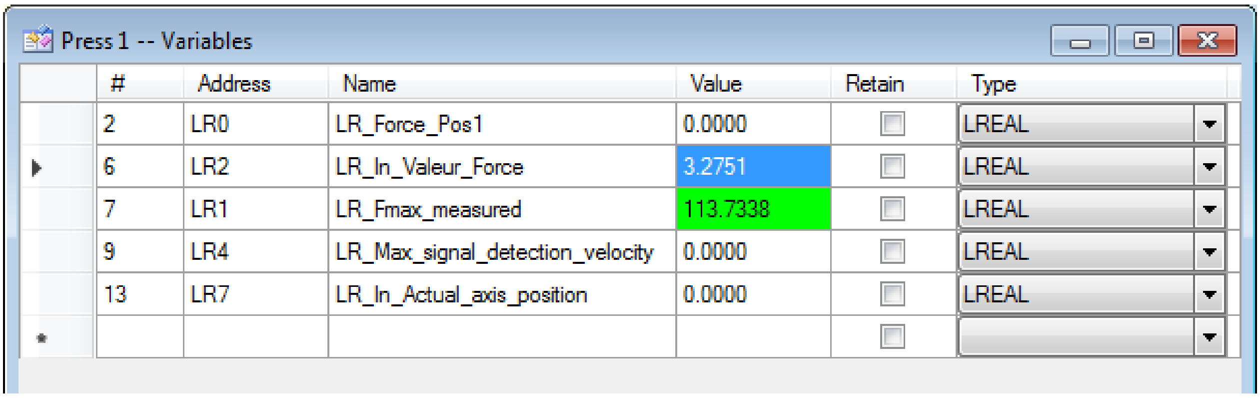


Figure 3: Viewing user variables

Part program instructions

List of instructions

- Wait delay
- Boolean On/OFF (Setting a Boolean variable to 0 or 1)
- Conditional/unconditional jump
- Arithmetic operations (Addition, Subtraction, Division, Multiplication)
- Assignment
- Positioning
- Max signal detection
- Min signal detection
- Signal measurement
- Stop on signal
- Curve recording
- Post-process force measurement
- Stop on force with velocity regulation
- Breakpoint
- Stopwatch
- Values recording in an array
- Force sensor management
- Clamping
- Stop clamping
- Position measurement
- Force regulator

To be able to use the instructions below, you must add a program and in it make a "right-click" and then "Insert function". You can also find all the functions in the general toolbar.

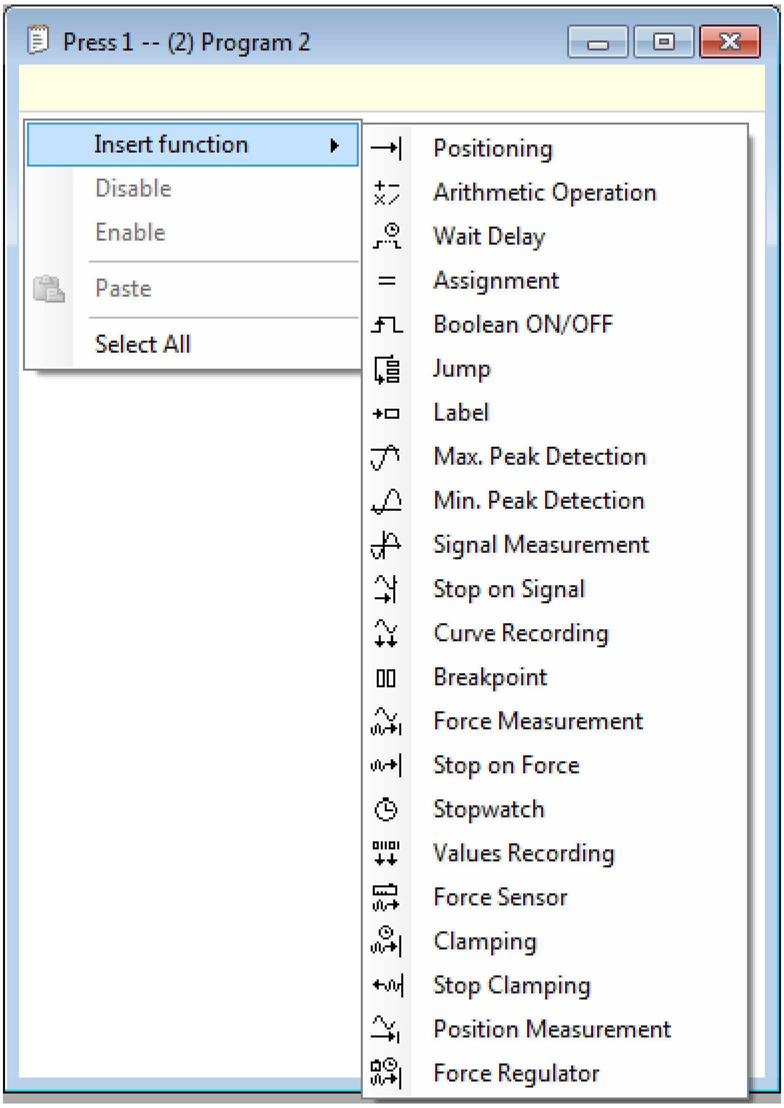


Figure 1: Adding an instruction

"Wait delay" instruction

Allow a period of time to elapse before the part-program continues to run.

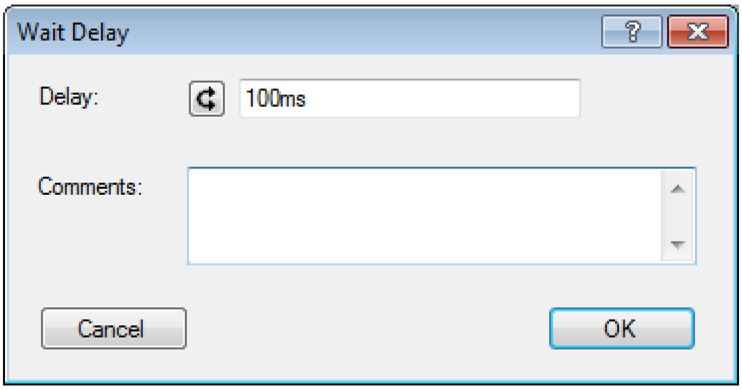


Figure 1: Window for programming a waiting delay

The time parameter is given in milliseconds[ms], the default value when the instruction is inserted in the part-program is 100[ms].

You can give a fixed (constant) time setpoint or give it using a user variable. The "" button next to the time parameter allows you to choose a variable as a time setpoint.

If the time parameter comes from a user variable, this one must be of type TIME.

In figure 1, the waiting time is a constant, unlike figure 2, or the waiting time comes from a user variable.

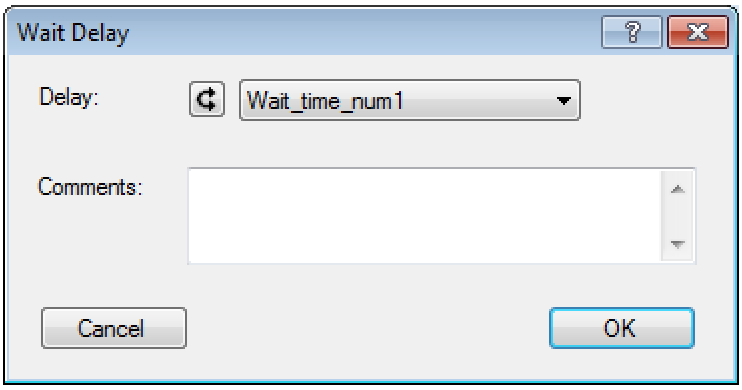


Figure 2: When the waiting time comes from a user variable

"Boolean ON/OFF" instruction

Instruction used in the part-program for "Set" or "Reset" (set to "1" or "0") a variable of type BOOL.

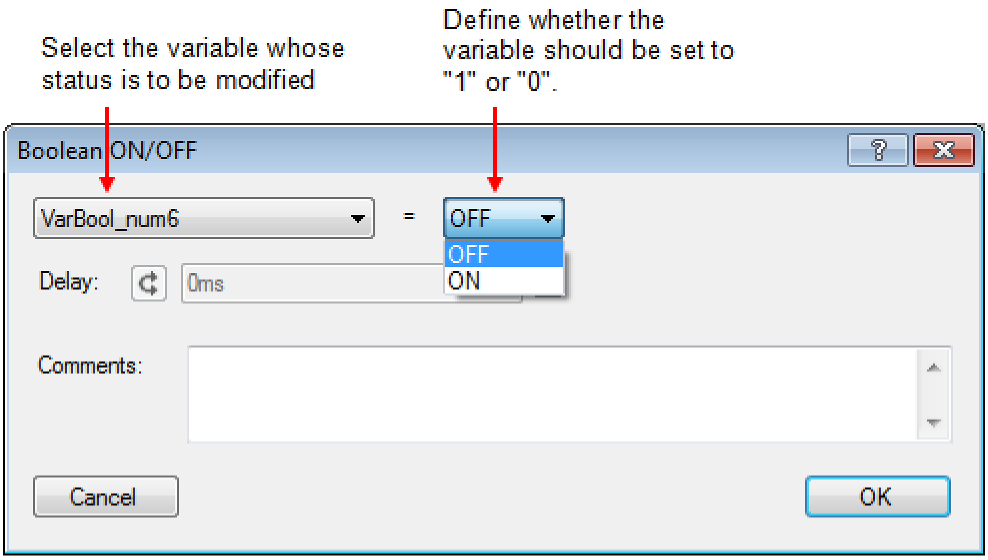


Figure 1: ON/OFF Boolean programming window

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Variable	Input	BOOL	--	Variable set to "1" or "0".
ON/OFF	Input	--	OFF	Setpoint for changing the status of the variable
Delay [ms]	Input	TIME	0[ms]	Time before changing the state of the variable

Table 1: List of instruction parameters

The basic function of the instruction is to set a BOOL variable to "1" or "0" for an indefinite period of time. But you can also define a time during which the variable must remain at "1" or "0", at the end of this time, the variable will take the opposite state.

The time set in parameter is always rounded up or down to a multiple of 50[ms], depending on whether the value set in parameter is respectively greater than "n X 50[ms] + 25[ms]" or smaller than "n X 50[ms] + 25[ms]".

Example:

If set time = 120[ms] => real time = 100[ms]

If set time = 135[ms] => real time = 150[ms]

The set time can be given by a constant or user variable in "TIME" format.

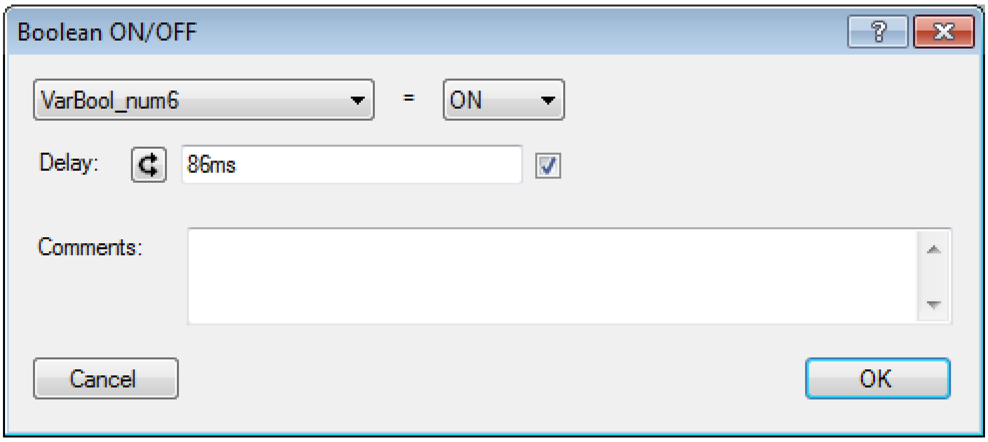


Figure 2: Programmation d'un temps avant la retombée du bit

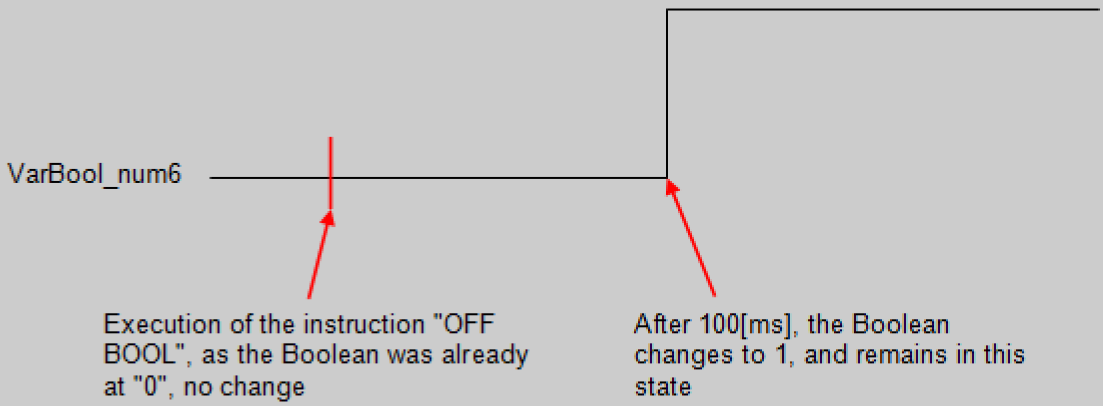
In the example in figure 2, the time that will elapse before the variable "VarBool_num6" set to "0" will actually be 100[ms], not 86[ms] as set.

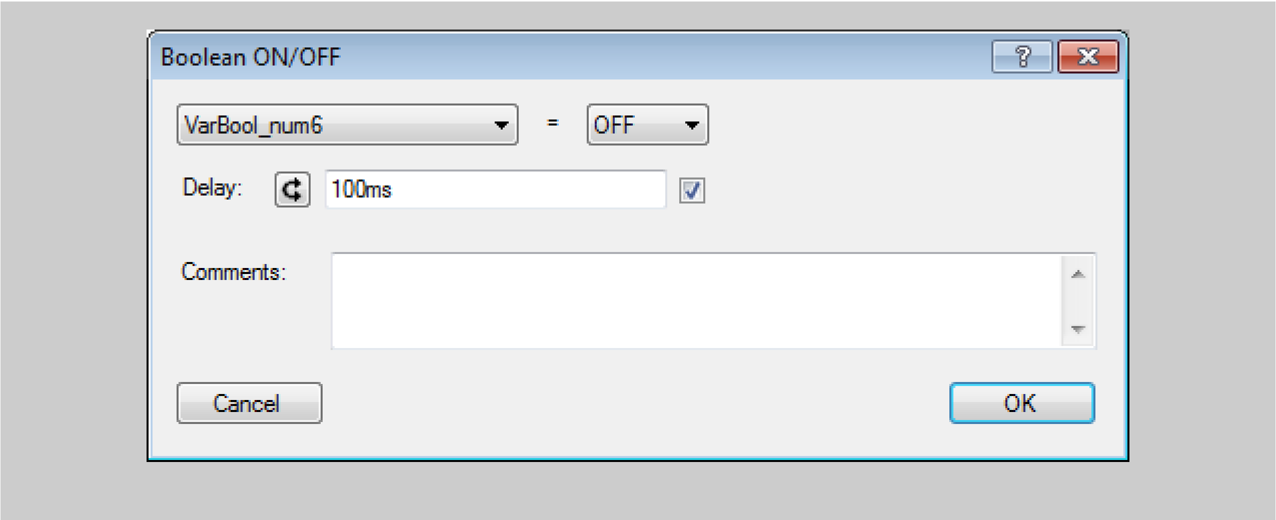
Tip:

If you want to reverse the state of a Boolean variable after a time, simply select in the instruction the state that the variable will have before the instruction is executed and add a time in the delay parameter. The variable will then keep its current state until the time delay has elapsed, then it will take the opposite state.

Example:

We want to set the value of a Boolean to "1" after 100[ms].




Boolean ON/OFF
VarBool_num6
=
OFF
Delay:
100ms
Comments:
Cancel
OK

"Conditional/unconditional jump" instruction

The jump instruction is used in conjunction with the label instruction. When executing the jump, the part-program will position itself on the selected label (jump destination).

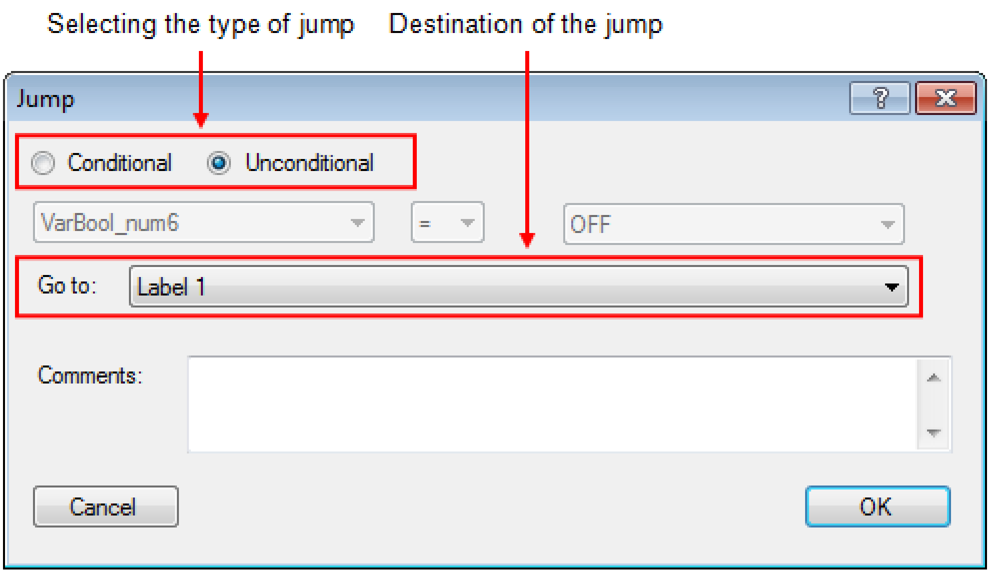


Figure 1: Jump programming window

There are two types of jumps:

- Unconditional Jump
- Conditional Jump

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
With or without condition	Input	--	Condition	Choice of jump type
Compared variable	Input	BOOL, REAL, LREAL, INT or DINT	--	Compared variable
Type of comparison	Input	--	=	Type of comparison, "=", ">", "<" ou "<>"
Comparing	Input	BOOL, REAL, LREAL, INT or DINT	--	Comparing variable
Go to	Input	--	--	Destination label

Table 1: List of instruction parameters

Unconditional jump

The unconditional jump is the easiest. As soon as the part-program arrives on this instruction, the program goes directly to the instruction that follows the destination label.

Conditional jump

The conditional jump allows you to add a condition to perform the jump. If the condition is not met, the jump is not performed and the following instruction is executed.

You can compare variables of type BOOL (0 or 1) or compare variables of type REAL, LREAL, INT or DINT with values.

- Compare if a Boolean variable is "=" or "<>" of "0" or "1".

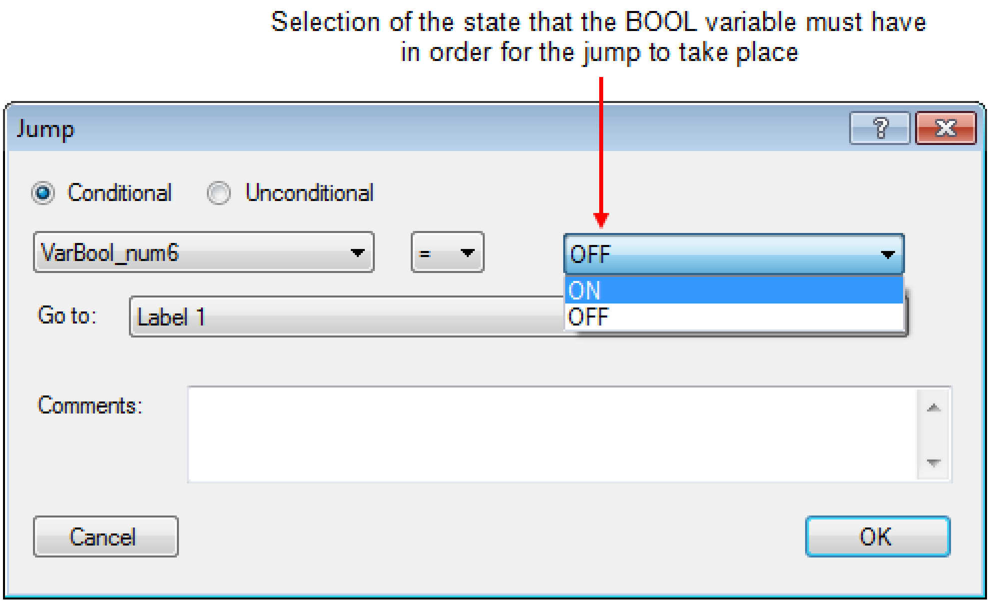


Figure 2: Comparison of a BOOL variable with a constant

- Compare whether a variable of type REAL, LREAL, INT or DINT is "<" ">" "=" or "<>" in relation to a constant/variable.

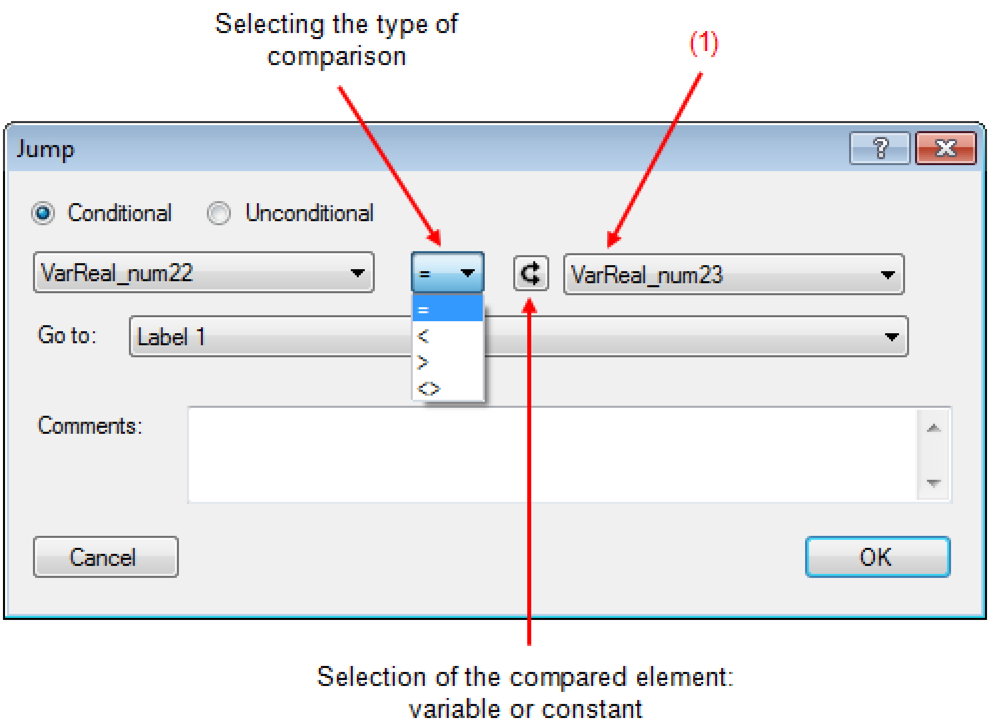


Figure 3: Comparison of a REAL variable with a REAL variable

(1) You can compare variables of different types.

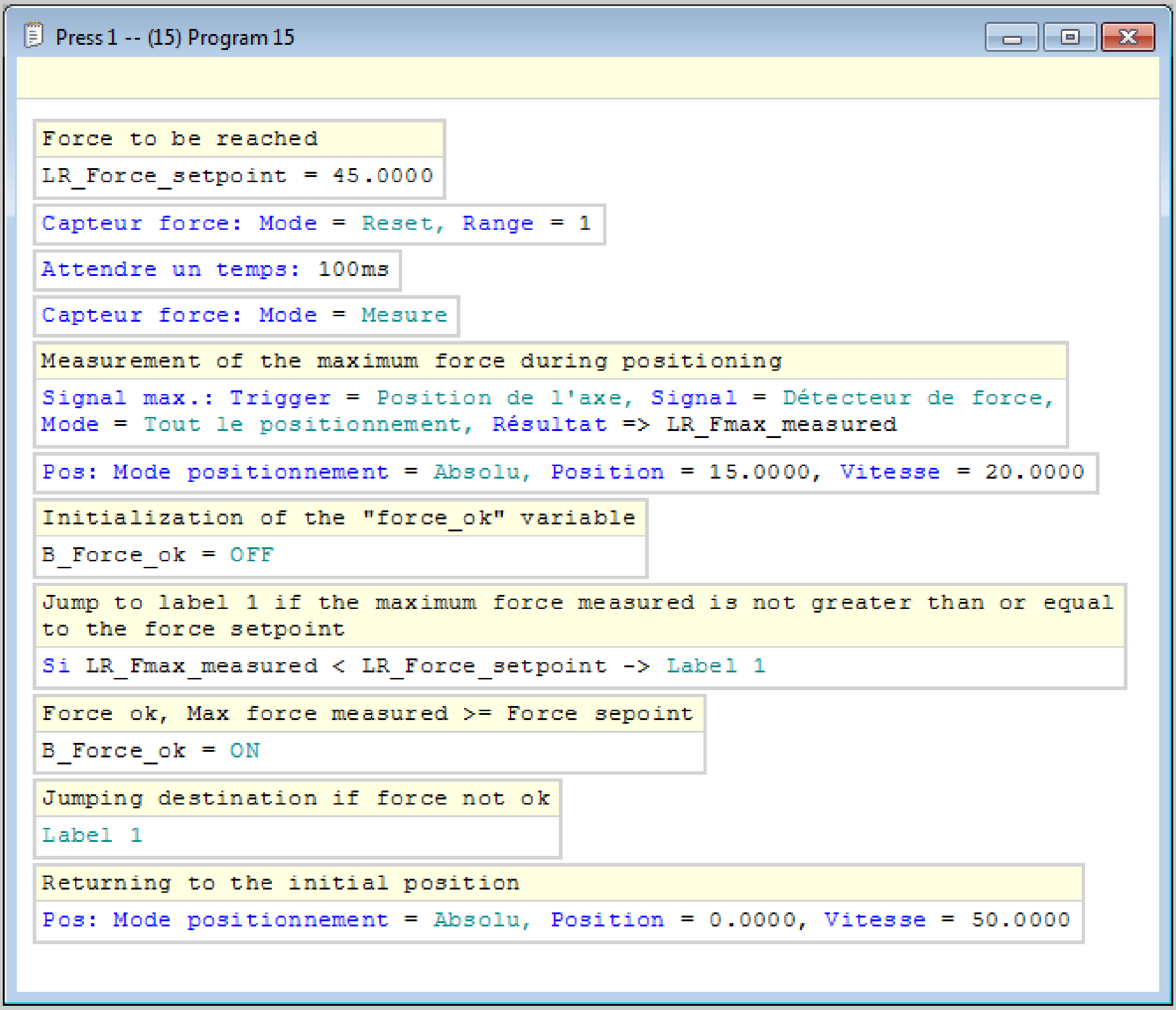
For example:

- Compare that the value of a REAL variable is greater than the value of a INT variable.
- Compare that the value of a DINT variable is equal to the value of a REAL variable.
- ...

Example of use:

We want to activate a Boolean variable when the maximum force measured during a positioning is greater than or equal to 45[N].

In this example, the jump instruction allows the maximum force measured during positioning to be compared with the setpoint force. If the maximum force is greater than or equal to the setpoint force, we set the Boolean variable "B_force_ok" to "1" and, if the maximum force is less than the setpoint force, the variable "B_force_ok" remains at "0", meaning that the force is not sufficient.



Press 1 -- (15) Program 15

```
Force to be reached
LR_Force_setpoint = 45.0000

Capteur force: Mode = Reset, Range = 1

Attendre un temps: 100ms

Capteur force: Mode = Mesure

Measurement of the maximum force during positioning
Signal max.: Trigger = Position de l'axe, Signal = Détecteur de force,
Mode = Tout le positionnement, Résultat => LR_Fmax_measured

Pos: Mode positionnement = Absolu, Position = 15.0000, Vitesse = 20.0000

Initialization of the "force_ok" variable
B_Force_ok = OFF

Jump to label 1 if the maximum force measured is not greater than or equal
to the force setpoint
Si LR_Fmax_measured < LR_Force_setpoint -> Label 1

Force ok, Max force measured >= Force sepoint
B_Force_ok = ON

Jumping destination if force not ok
Label 1

Returning to the initial position
Pos: Mode positionnement = Absolu, Position = 0.0000, Vitesse = 50.0000
```

Figure 4: Example of using the jump instruction

Arithmetic operations

In this section, you will find the different arithmetic operations that you can perform in the part-program.

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Result	Output	REAL, LREAL, INT ou DINT	--	Variable where the result of the operation is stored
Operand 1 variable or constant	Input	REAL, LREAL, INT ou DINT	--	First operand (variable or constant)
Operand 2 variable or constant	Input	REAL, LREAL, INT ou DINT	--	Second operand (variable or constant)
Type of operation	Input	"+", "-", "/", "x"	"+"	Type of operation, addition, subtraction, division, multiplication

Table 1: List of instruction parameters

The result of an arithmetic operation will always be stored in a user variable. This variable can be of any format (see figure 1).

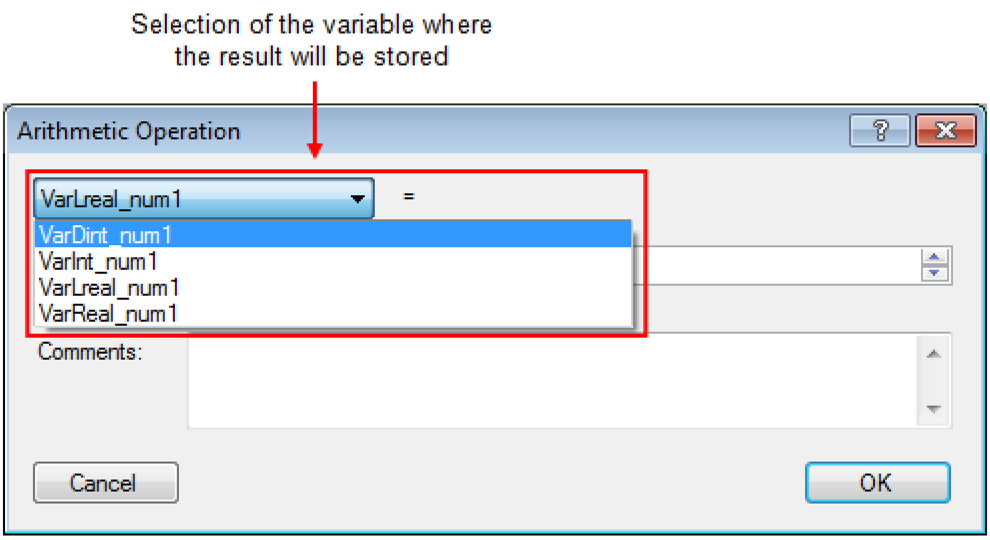


Figure 1: User variable selection containing operation result

In figure 1, the variable containing the result can be declared in the formats below:

- INT (16 bits)
- DINT (32 bits)
- REAL (32 bits floating)
- LREAL (64 bits floating)

These different operations can be done between different variable formats, for example the addition of a INT and a REAL and the result is returned in LREAL format.

Internally, each operand, whatever its format, is automatically transformed into LREAL (64 floating bits), all calculations are performed in this format. When the calculation is performed, the result is converted to the format of the result variable.

If the result is to be stored in a user variable of type INT, and the result is greater than 32760 or smaller than 32760, then error n°16 ("arithmetic overflow on integer") occurs. The execution of the program is then stopped.

When the result is to be transferred to a user variable of type DINT, the same error occurs if the result exceeds 2000000000 or -2000000000.

Below is a list of the arithmetic operations that can be performed in the part-program:

- Addition
- Subtraction
- Multiplication
- Division

"Addition" instruction

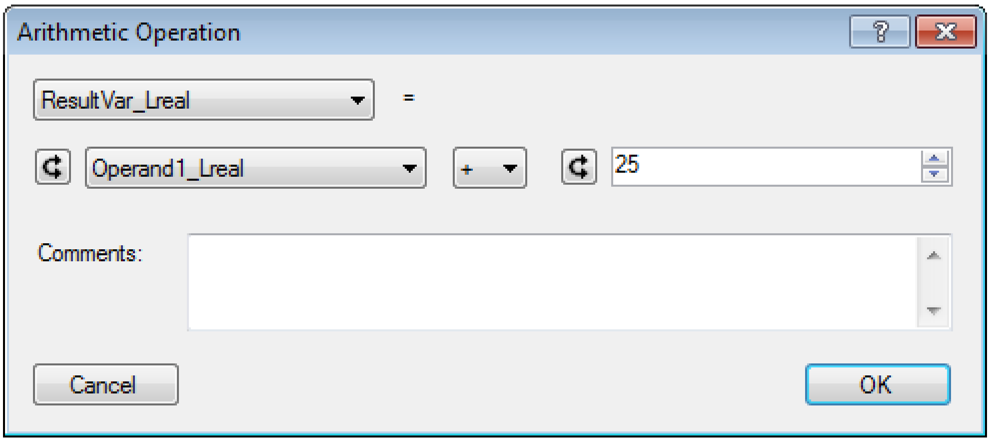


Figure 1: Example of a addition programmed in the part-program

"Division" instruction

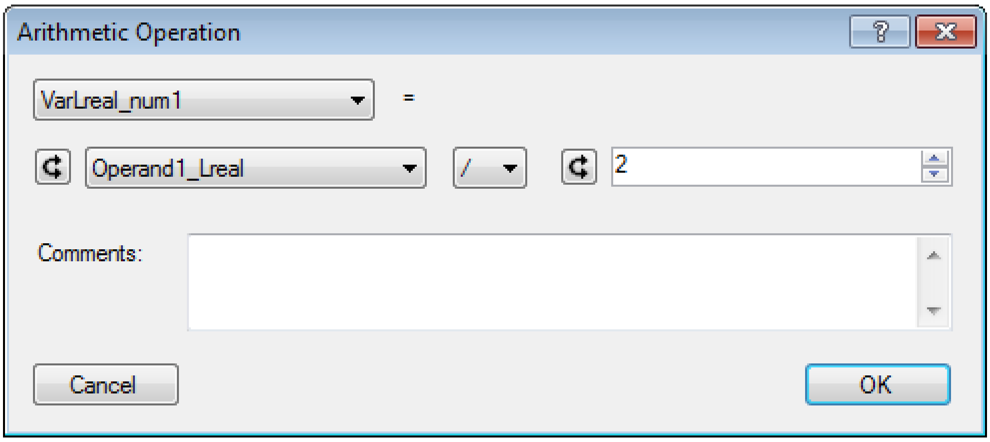


Figure 1: Example of a division programmed in the part-program

"Multiplication" instruction

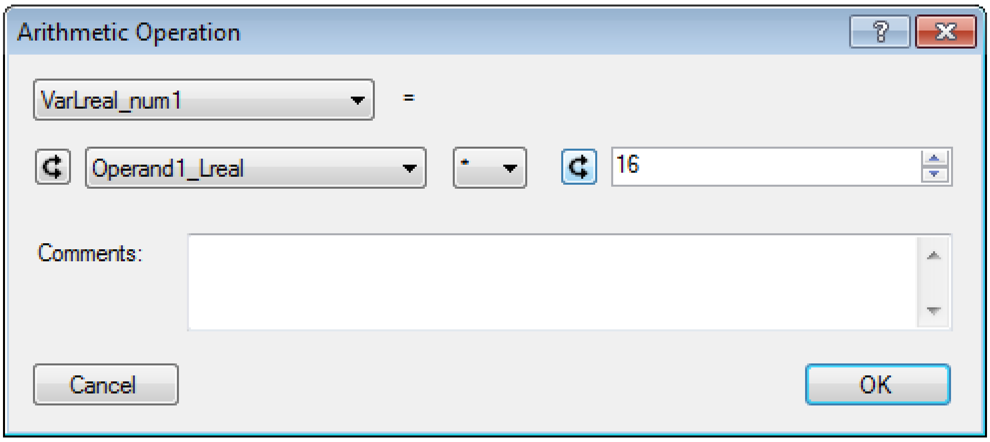


Figure 1: Example of a multiplication programmed in the part-program

"Subtraction" instruction

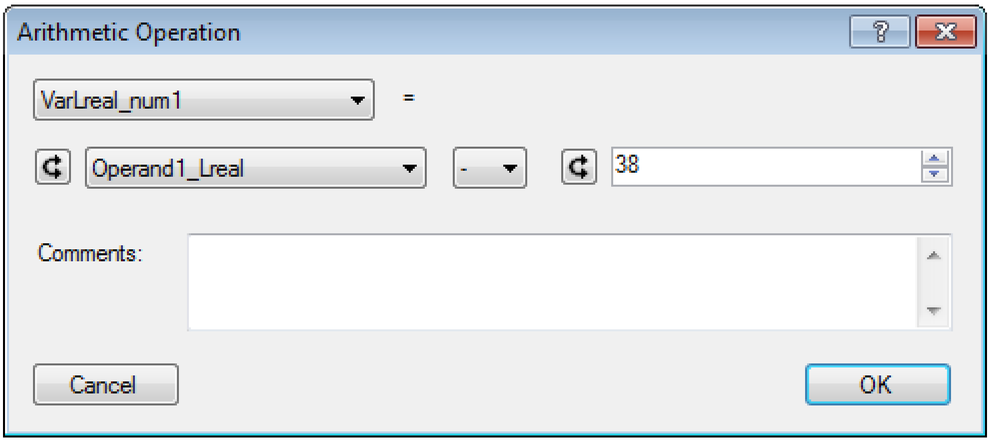


Figure 1: Example of a subtraction programmed in the part-program

"Assignment" instruction

This instruction is used to assign a value to a user variable.

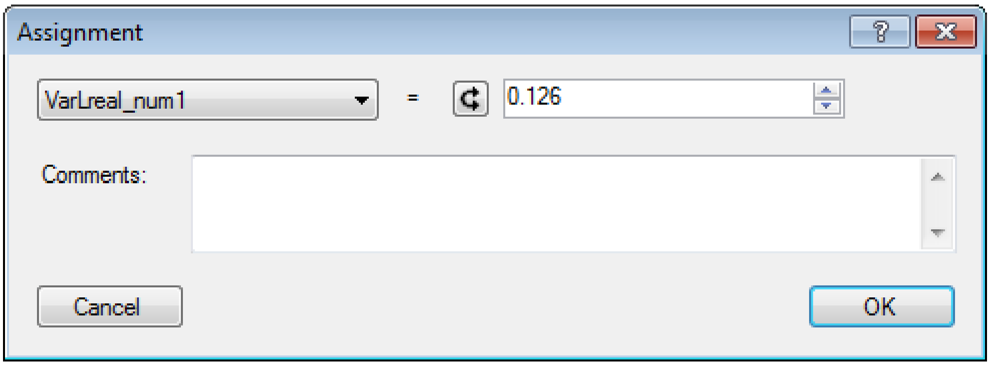


Figure 1: Example of using an assignment in the part-program

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Variable to be assigned	Output	REAL, LREAL, INT or DINT	--	Variable to be assigned
Assignment value	Input	REAL, LREAL, INT or DINT	0	Assignment value (variable or constant)

Table 1: List of instruction parameters

The user variable formats that can be assigned are as follows:

- INT (16 bits)
- DINT (32 bits)
- REAL (32 bits floating)
- LREAL (64 bits floating)

As shown in figure 2, you can assign a variable from one format to a variable from another format.

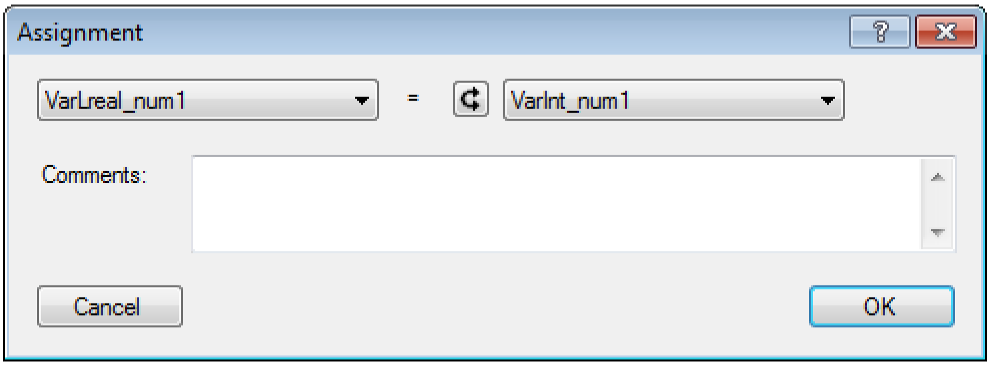


Figure 2: Assigning an INT variable to a LREAL variable

"Positioning" instruction

This instruction allows you to perform the movements with the press.

The parameters of this instruction are divided into four distinct tabs:

- Standard parameters: Speed, position and absolute/relative mode.
- Limits : Force, position, speed and acceleration/deceleration.
- Velocity profile: Type of profile, acceleration and jerk
- Expert parameters: Mainly used for transitions between commands.

When the press is lowered, the axis position increases.

Standard parameters

In this tab, you define the minimum parameters for an axis movement.

If no other tabs are modified, the speed profile will be the default one. That is to say, the "Smooth" profile, the acceleration, deceleration and various jerks values will be the values given by the default parameters.

Parameter name	Declaration	Type of data	Default value	Description
Position [mm]	Input	LREAL	0[mm]	Position setpoint [mm]
Velocity [mm/s]	Input	LREAL	20[mm/s]	Velocity setpoint [mm/s]
Positioning mode	Input	EnumPositionning Mode	Absolute	Positioning mode: absolute or relative

Table 1: List of standard parameters

Explanation of the positioning mode

Two positioning modes are possible, absolute or relative.

In absolute mode, the position setpoint is given with respect to the axis reference position (0[mm]).

In relative mode, the position setpoint is given with respect to the current axis position.

Example, if the position setpoint is 20[mm], and the actual axis position is 30[mm]. (see figure 1)

(1) In absolute mode, the position setpoint of 20[mm] is directly the actual setpoint position.

(2) In relative mode, the position setpoint is added to the actual position: Position setpoint 20[mm] + Actual position 30[mm] = Actual setpoint position 50[mm]

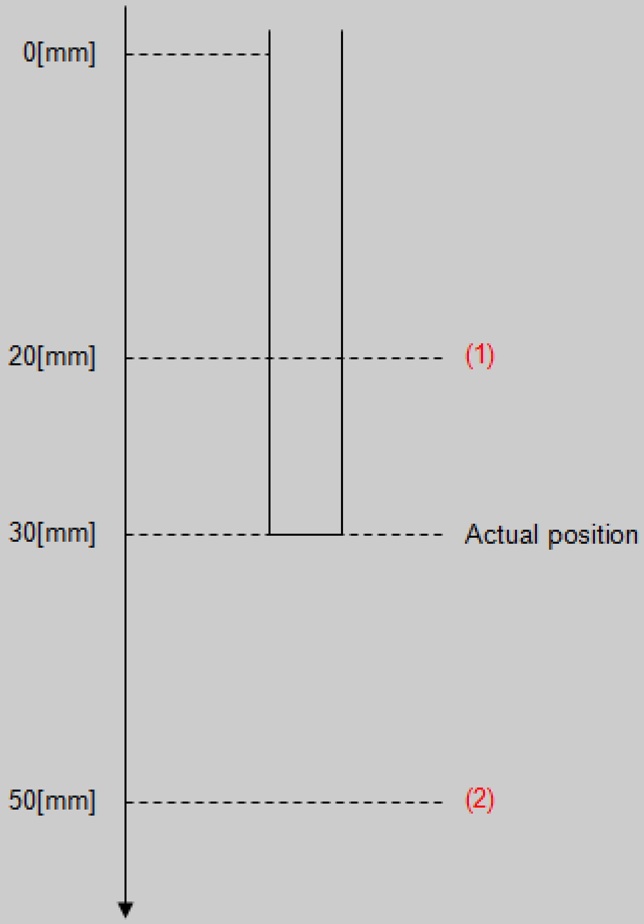


Figure 1: Difference between absolute and relative movement

Limits Tab

From this tab, you can define the force, position, speed and acceleration/deceleration limits during positioning.

These limits have been created to reduce the risk of collision and reduce damage in the event of mechanical shock.

Parameter name	Declaration	Type of data	Default value	Description
Minimum force [N]	Input	LREAL	0[N]	Minimum force limit
Maximum force [N]	Input	LREAL	0[N]	Maximum force limit
Minimum position [mm]	Input	LREAL	-5[mm]	Minimum position limit
Maximum position [mm]	Input	LREAL	90[mm]	Maximum position limit
Maximum velocity [mm/s]	Input	LREAL	200[mm/s]	Velocity limit

Parameter name	Declaration	Type of data	Default value	Description
Maximum acceleration/deceleration [mm/s2]	Input	LREAL	7000[mm/s2]	Acceleration/deceleration limit

Table 2: List of limits

Velocity Profile Tab

This tab allows you to program the speed profile, there are two different profiles to choose from.

- "Trapezoidal" velocity profile
- "Smooth" velocity profile

Depending on the selected velocity profile, it is possible to modify the different accelerations, decelerations and jerks.

Parameter name	Declaration	Type of data	Default value	Description
Positive acceleration [mm/s2]	Input	LREAL	6000[mm/s2]	Acceleration setpoint
Negative acceleration [mm/s2]	Input	LREAL	6000[mm/s2]	Deceleration setpoint
Positive acc. start jerk [mm/s3]	Input	LREAL	50000[mm/s3]	Jerk setpoint of the start of the acceleration phase
Positive acc. end jerk [mm/s3]	Input	LREAL	50000[mm/s3]	Jerk setpoint of the end of the acceleration phase
Negative acc. start jerk [mm/s3]	Input	LREAL	50000[mm/s3]	Jerk setpoint of the start of the deceleration phase
Negative acc. end jerk [mm/s3]	Input	LREAL	50000[mm/s3]	Jerk setpoint of the end of the deceleration phase
Velocity profile [-]	Input	EnumProfile	Smooth	Selection between the 2 profiles described below

Table 3: List of velocity profile parameters

Explanation of existing velocity profiles

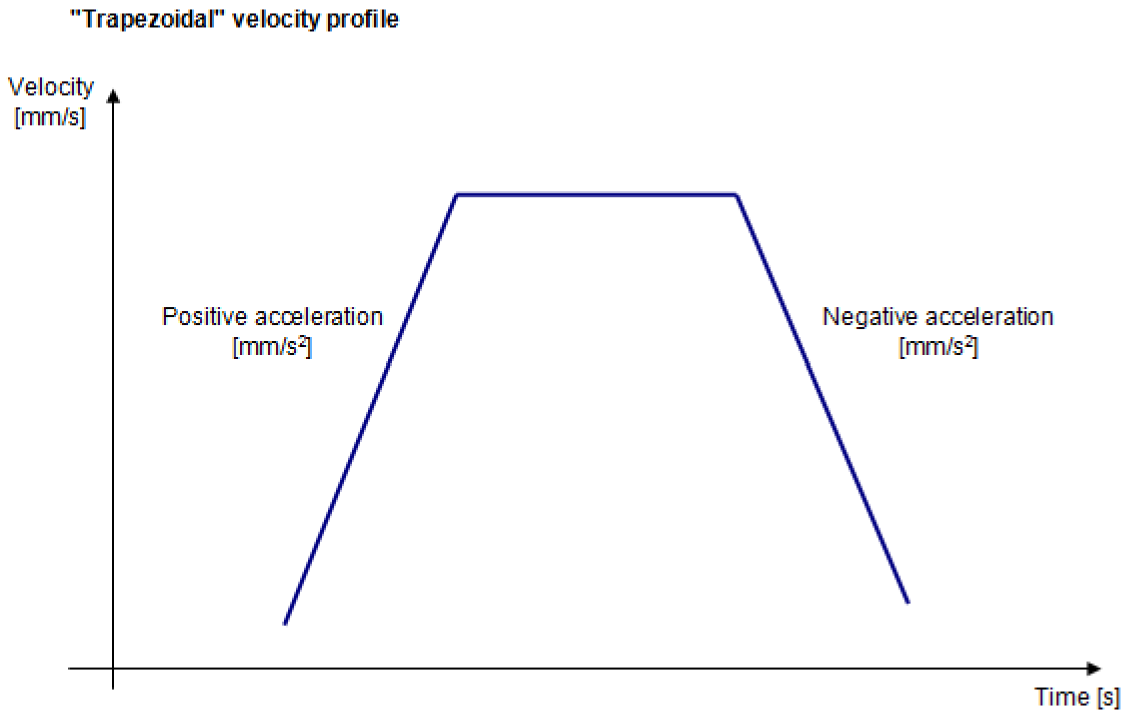


Figure 2: Trapezoidal profile

With the "Trapezoidal" velocity profile, you can change the acceleration[mm/s2] and deceleration[mm/s2] parameters.

The higher the acceleration, the faster the target velocity will be reached, and the higher the deceleration, the faster the axis will stop.

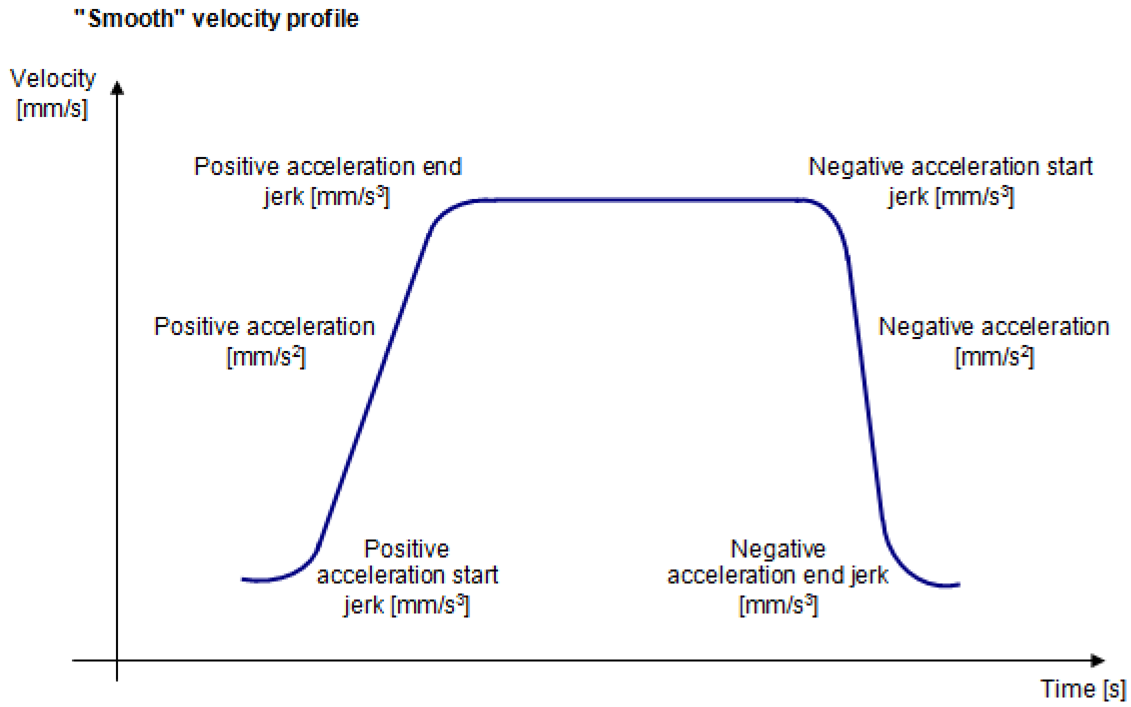


Figure 3: Smooth profile

With the "Smooth" velocity profile, in addition to the acceleration and deceleration parameters, you can adjust the different jerks of the movement.

The lower the jerks at the beginning of acceleration and beginning of deceleration, the slower the times for the acceleration and deceleration setpoints to be reached.

The adjustment of these different jerks is used to soften the movement.

Expert Tab

From this tab, you can define the types of transition between the controls and the positioning activation mode.

You can also choose to perform a positioning with the velocity, acceleration, deceleration and jerks setpoints from the previous positioning.

Parameter name	Declaration	Type of data	Default value	Description
Velocity type	Input	Enum	User default	Type of velocity definition. USER_DEFAULT (149) : User default EFFECTIVE (45) : Last programmed set velocity DIRECT(40) : Value entry CURRENT (33) : Current velocity RESULTING (312) : Resulting velocity at end of acceleration

Parameter name	Declaration	Type of data	Default value	Description
Positive acceleration type	Input	Enum	Direct	Type of positive acceleration definition. USER_DEFAULT (149) : User default EFFECTIVE (45) : Last programmed value DIRECT(40) : Value entry
Negative acceleration type	Input	Enum	Direct	Type of negative acceleration definition. USER_DEFAULT (149) : User default EFFECTIVE (45) : Last programmed value DIRECT(40) : Value entry
Positive acceleration start jerk type	Input	Enum	Direct	Type of definition for acceleration start jerk. USER_DEFAULT (149) : User default EFFECTIVE (45) : Last programmed jerk DIRECT(40) : Value entry
Positive acceleration end jerk type	Input	Enum	Direct	Type of definition for acceleration end jerk. USER_DEFAULT (149) : User default EFFECTIVE (45) : Last programmed jerk DIRECT(40) : Value entry
Negative acceleration start jerk type	Input	Enum	Direct	Type of definition for deceleration start jerk. USER_DEFAULT (149) : User default EFFECTIVE (45) : Last programmed jerk DIRECT(40) : Value entry
Negative acceleration end jerk type	Input	Enum	Direct	Type of definition for deceleration end jerk. USER_DEFAULT (149) : User default EFFECTIVE (45) : Last programmed jerk DIRECT(40) : Value entry
Blending mode	Input	Enum	Inactive	Specifies blending mode. USER_DEFAULT (149) : User default EFFECTIVE (45) : The last programmed blending value INACTIVE (61) : No blending ACTIVE (4) : Blending

Parameter name	Declaration	Type of data	Default value	Description
Merge mode	Input	Enum	Sequential	Specifies when the command takes effect relative to the motion. IMMEDIATELY (60) : Substitute current motion immediately SEQUENTIAL (119) : Command is attached to existing motion commands. NEXT_MOTION (89) : Command is attached to existing motion commands and buffer is cleared. SUPERIMPOSED_MOTION_MERGE (142) : Current motion is superimposed.
Next command	Input	Enum	When motion done	Specifies condition for transition to next command. IMMEDIATELY (60) : Immediate command advance WHEN_BUFFER_READY (159) : After entry in the command queue AT_MOTION_START (13) : At start of interpolation WHEN_ACCELERATION_DONE (156) : At end of acceleration phase AT_DECELERATION_START (12) : At start of deceleration phase WHEN_INTERPOLATION_DONE (162) : At end of setpoint interpolation WHEN_MOTION_DONE (163) : When motion is completed

Table 4: Expert parameters

Explanation on the use of "Controls"

There are 7 different control instructions, which are listed below:

1. Max signal detection
2. Min signal detection
3. Signal measurement
4. Stop on signal
5. Curve recording (Force/Position)
6. Stop on force with velocity regulation
7. Position measurement
8. Post-process force measurement

All these instructions are used in conjunction with the positioning instruction.

In a program, the first 7 control instructions must be placed before a positioning because the control is performed in the following positioning. (see example figure 1 below)

In the example in figure 1, the "maximum signal" control will only be active for positioning 1. At the end of this first positioning, the control is automatically deactivated.

If you want to detect a "maximum signal" in the second positioning, you must add a second "Max Signal" control before positioning 2.

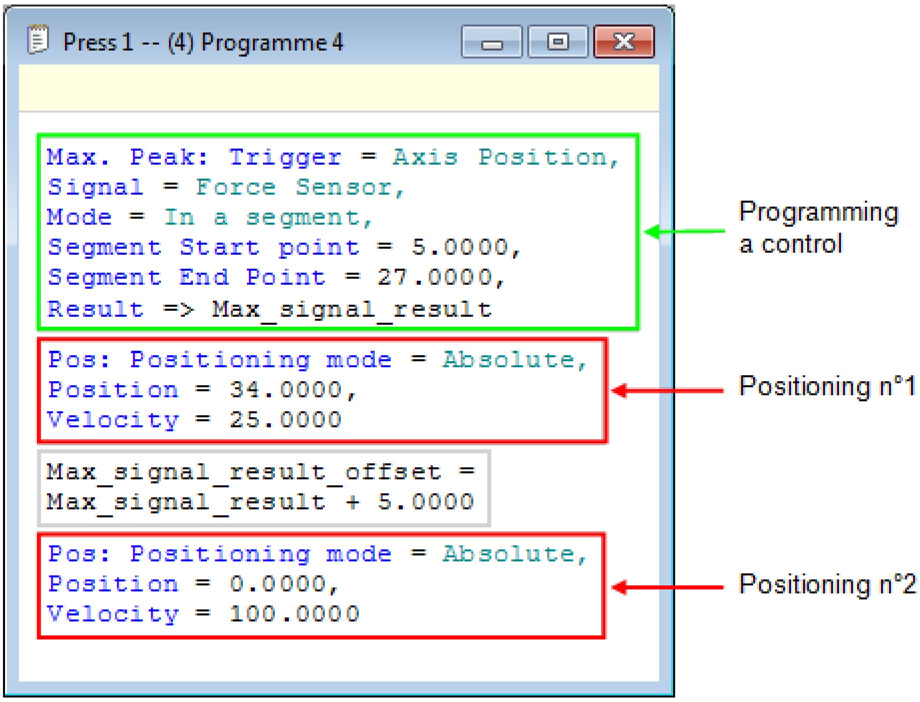


Figure 1: Programming a control

For the 8th instruction ("Post-process force measurement"), this control must be placed after the positioning instruction. In addition, the positioning for which this control will work must be preceded by a registration control, see figure 2 below.

In the example in figure 2, controls 1 and 2 refer to positioning 1. Neither of these two controls is active during positioning 2.

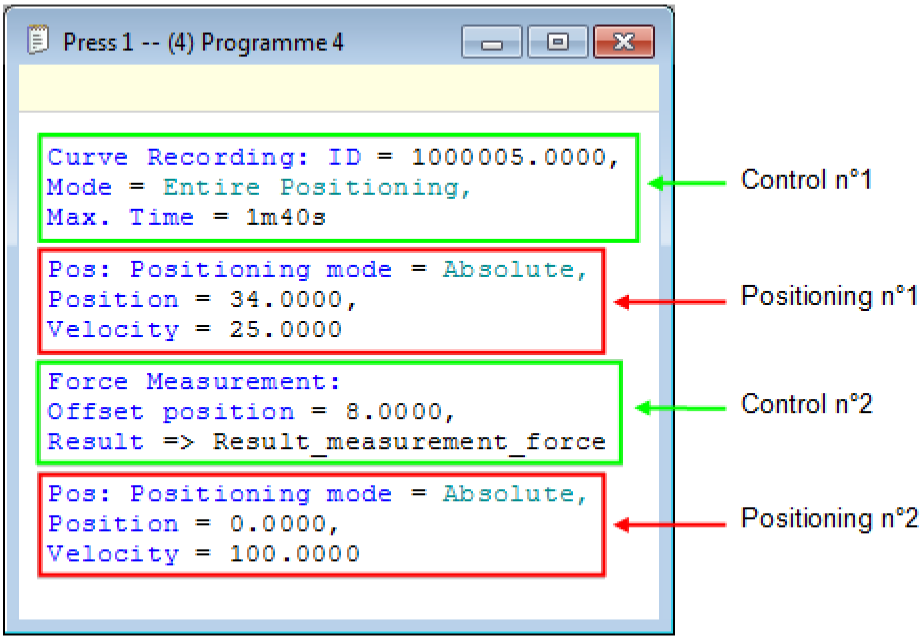


Figure 2: Programming the "Post-process force measurement" control

"Max signal detection" control instruction

This instruction allows to detect in a control area given by a trigger signal, the maximum value that the signal to be measured will reach.

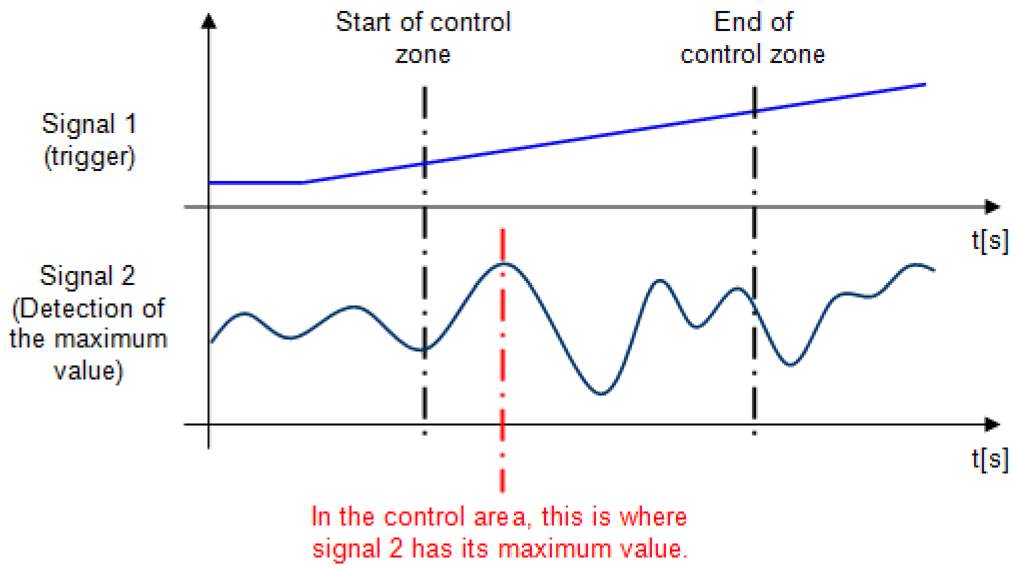


Figure 1: Explanatory graph of the control function

The signals that can be defined as triggers are as follows:

- Axis position
- Force sensor
- Measurement sensor signal

For each of these signals, it is possible to give the area in which the control function will work, or perform the control throughout the positioning.

It is possible to search for the maximum value of three different signals:

- Axis position
- Force sensor
- Measurement sensor signal

As soon as the trigger signal leaves the control area, the maximum value found will be stored in a user variable in LREAL format.

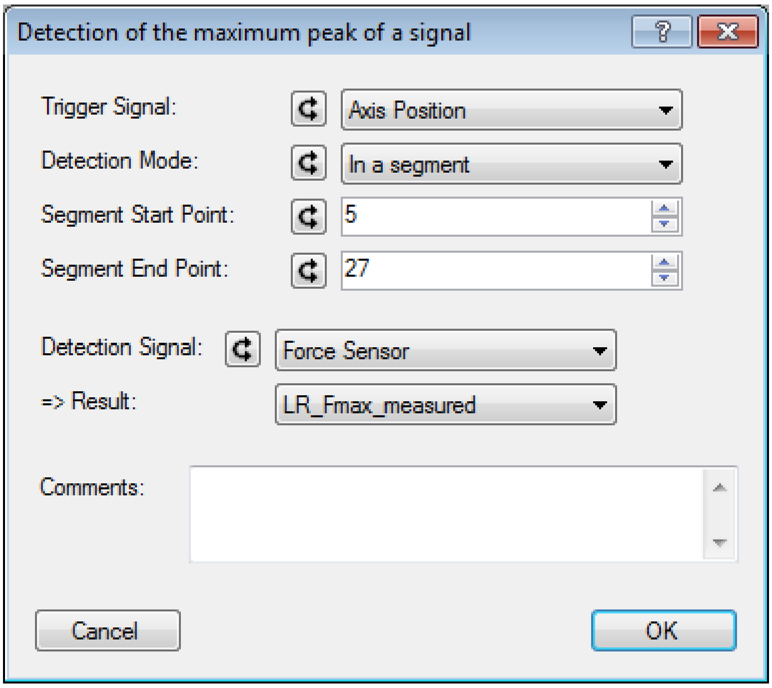


Figure 2: Example of programming to detect the maximum force when the axis position is between 5[mm] and 27[mm].

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Trigger signal	Input	Enum [Source signal]	Axis position	Function trigger signal, measuring sensor, axis position or force sensor
Detection mode	Input	Enum [In a segment or entire positioning]	In a segment	Detection mode, In a segment or entire positioning
Segment start point	Input	LREAL	0	Start position of the control zone
Segment end point	Input	LREAL	0	End position of the control zone
Detected signal	Input	Enum [Source signal]	Force sensor	Detected signal, measuring sensor, axis position or force sensor
Résult	Output	LREAL	--	Maximum value measured during detection

Table 1: List of instruction parameters

Example of use:

We want to detect the maximum force measured by the force sensor when the axis moves from the absolute position 25[mm] to the absolute position 45[mm].

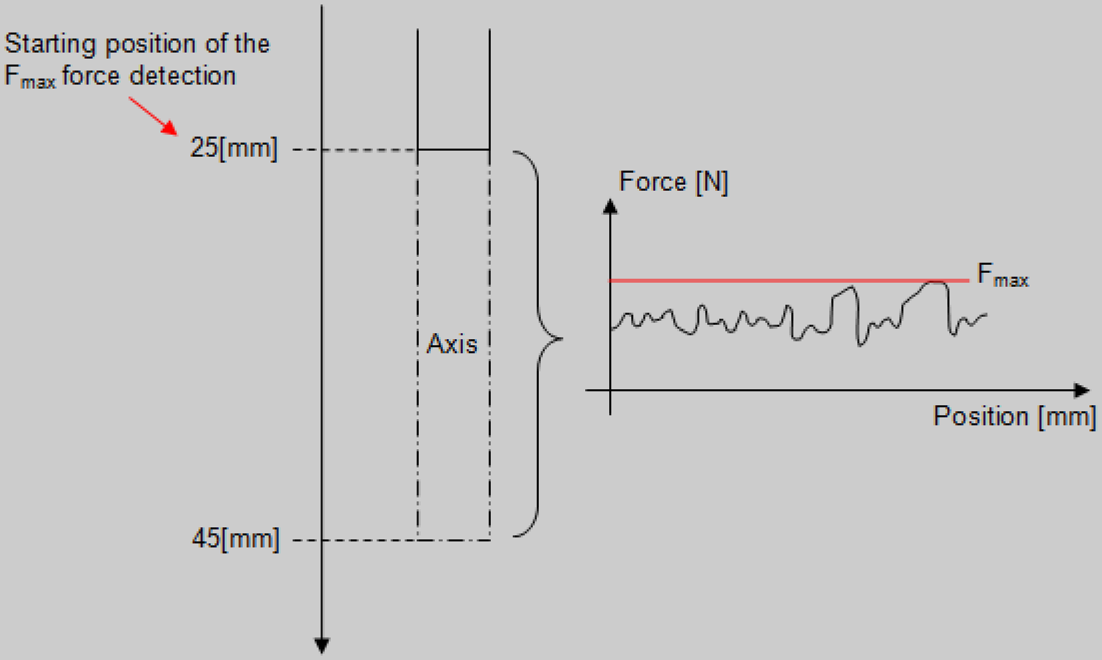


Figure 3: Example of using max signal detection

Below, the program used for the example above.

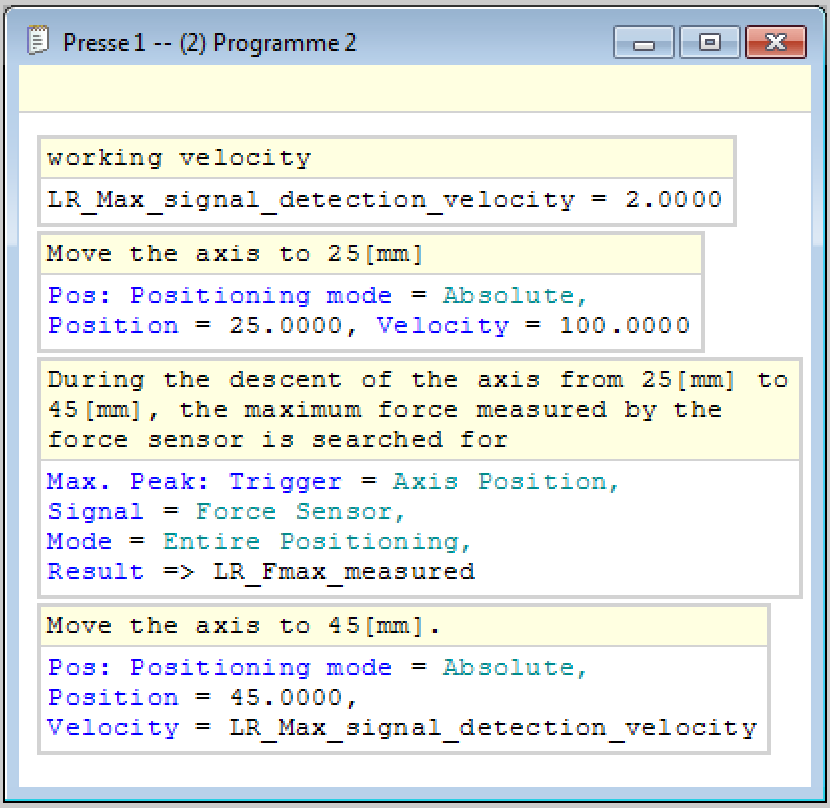


Figure 4: Program for maximum force detection between 25 and 45[mm]

Note: The force sampling is done every 2[ms]. If a peak of the signal to be measured occurs between two samples, it will not be detected.

"Min signal detection" control instruction

This instruction allows to detect in a control area given by a trigger signal, the minimum value that the signal to be measured will reach.

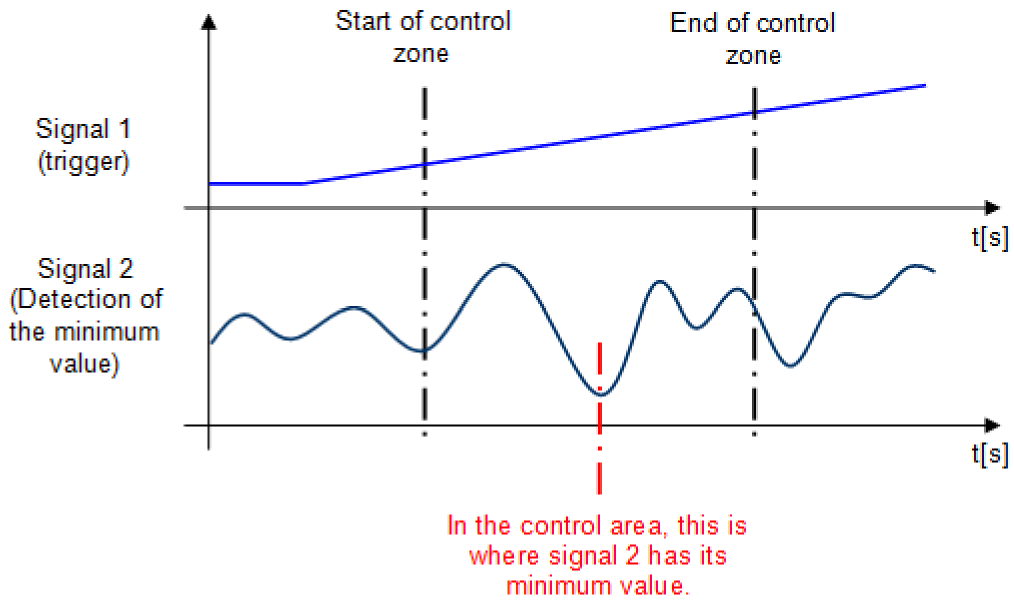


Figure 1: Explanatory graph of the control function

The signals that can be defined as triggers are as follows:

- Axis position
- Force sensor
- Measurement sensor signal

For each of these signals, it is possible to give the area in which the control function will work, or perform the control throughout the positioning.

It is possible to search for the maximum value of three different signals:

- Axis position
- Force sensor
- Measurement sensor signal

As soon as the trigger signal leaves the control area, the minimum value found will be stored in a user variable in LREAL format.

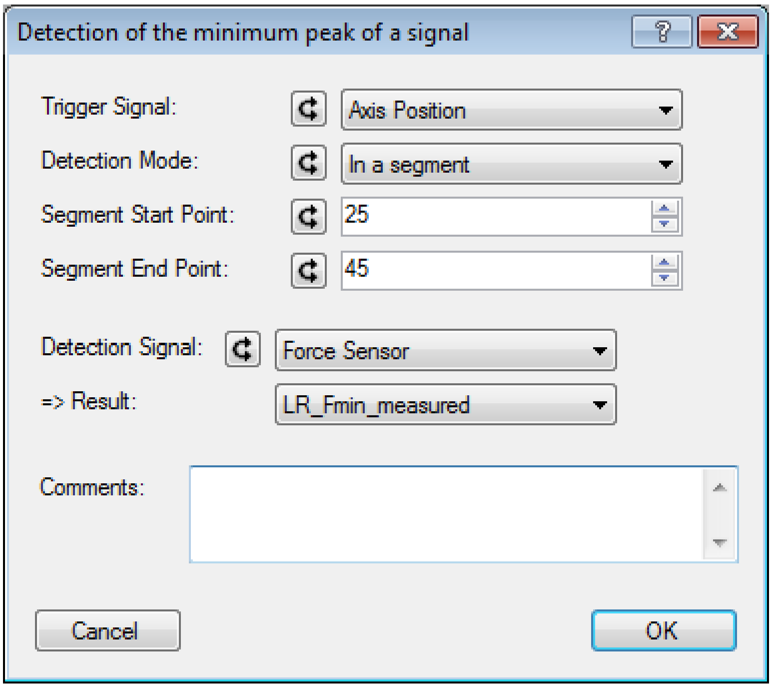


Figure 2: Example of programming to detect the minimum force when the axis position is between 5[mm] and 27[mm].

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Trigger signal	Input	Enum [Source signal]	Axis position	Function trigger signal, measuring sensor, axis position or force sensor
Detection mode	Input	Enum [In a segment or entire positioning]	In a segment	Detection mode, In a segment or entire positioning
Segment start point	Input	LREAL	0	Start position of the control zone
Segment end point	Input	LREAL	0	End position of the control zone
Detected signal	Input	Enum [Source signal]	Force sensor	Detected signal, measuring sensor, axis position or force sensor
Résult	Output	LREAL	--	Minimum value measured during detection

Table 1: List of instruction parameters

Example of use:

We want to detect the minimum force measured by the force sensor, when the axis moves from the absolute position 25[mm] to the absolute position 45[mm].

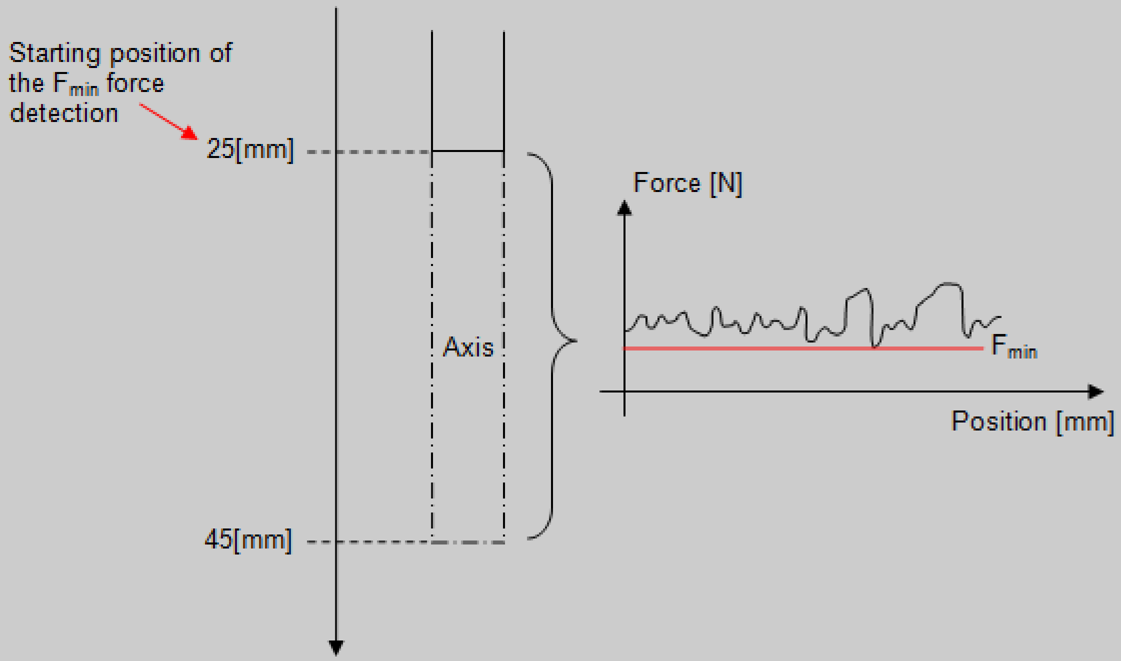


Figure 3: Example of using min signal detection

Below, the program used for the example above.

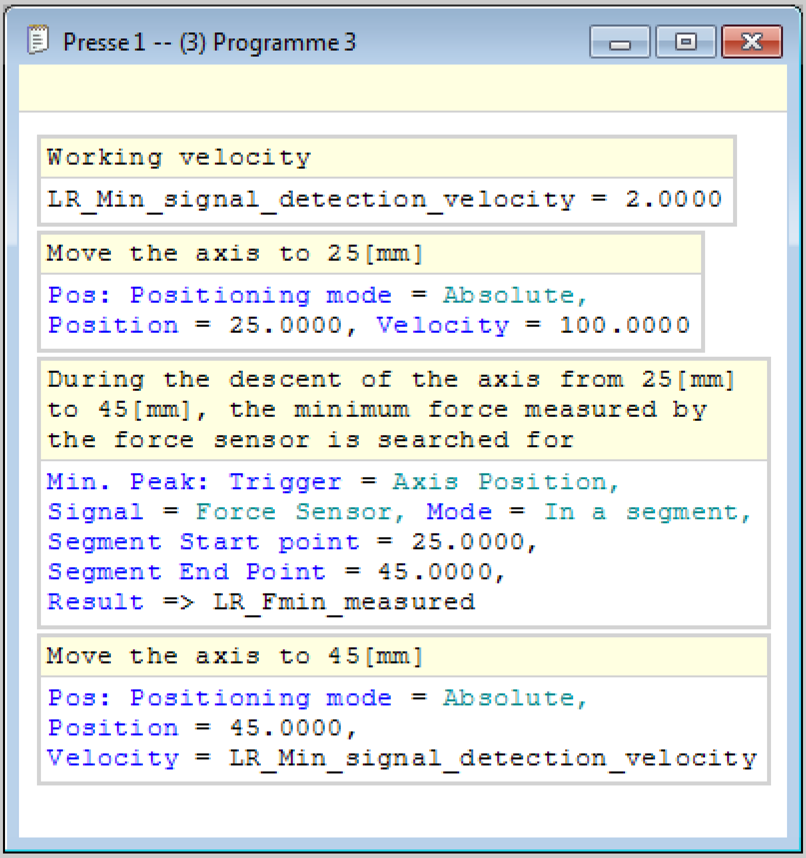


Figure 4: Program for minimum force detection between 25 and 45[mm]

Note: The force sampling is done every 2[ms]. If a peak of the signal to be measured occurs between two samples, it will not be detected.

"Signal measurement" control instruction

This instruction is used to measure the value of an "X" signal when a trigger signal "Y" reaches a certain threshold.

For a measurement of the "X" signal to be made, the trigger signal must be either larger or smaller (depending on the type of comparison) than the value of the threshold parameter.

When the signal "X" is measured, it is multiplied by a factor "A" and an offset "B" is added. This factor and offset are function parameters (by default, the factor is set to 1 and the offset is set to 0). These parameters can come from a user variable or be given as constants.

The result stored in the return variable will be equal to:

Resulting measured value = Measured value at threshold * Factor A + Offset B

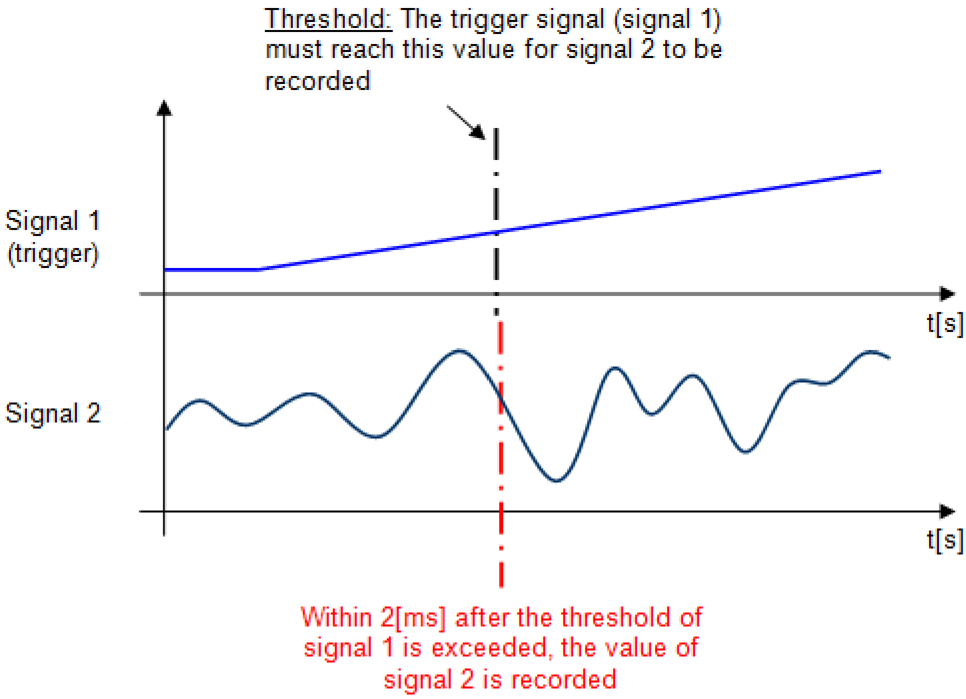


Figure 1: Graph explaining how the "Signal Measurement" control works

The signals that can be defined as triggers are as follows:

- Axis position
- Force sensor
- Measurement sensor signal

There are two types of trigger-to-threshold comparisons:

- Comparison ">': If the trigger signal is greater than or equal to the threshold => Recording the value of the measured signal.
- Comparison'<': If the trigger signal is smaller or equal to the threshold => Recording the value of the measured signal.

When the trigger signal passes the given threshold as a parameter, it is possible to record the value of one of the three signals below:

- Axis position
- Force sensor
- Measurement sensor signal

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Trigger signal	Input	Enum [Signal source]	Axis Position	Comparison trigger signal, measuring sensor, axis position or force sensor
Comparison	Input	Enum [Comparison]	>	Comparison type ">" or "<"
Threshold	Input	LREAL	0	Threshold that the trigger signal must reach to record the measured signal
Measured signal	Input	Enum [Signal source]	Force sensor	Measured signal, measuring sensor, axis position or force sensor
Factor	Input	LREAL	1	Factor multiplied by the result
Offset	Input	LREAL	0	Offset added to the result
Result	Output	LREAL	--	Measured value when the comparison becomes true * factor + offset

Table 1: List of instruction parameters

Tip:

If the "result" variable must contain the value that has actually been measured, simply set the following offset and multiplication parameters:

Factor A = 1

Offset B = 0

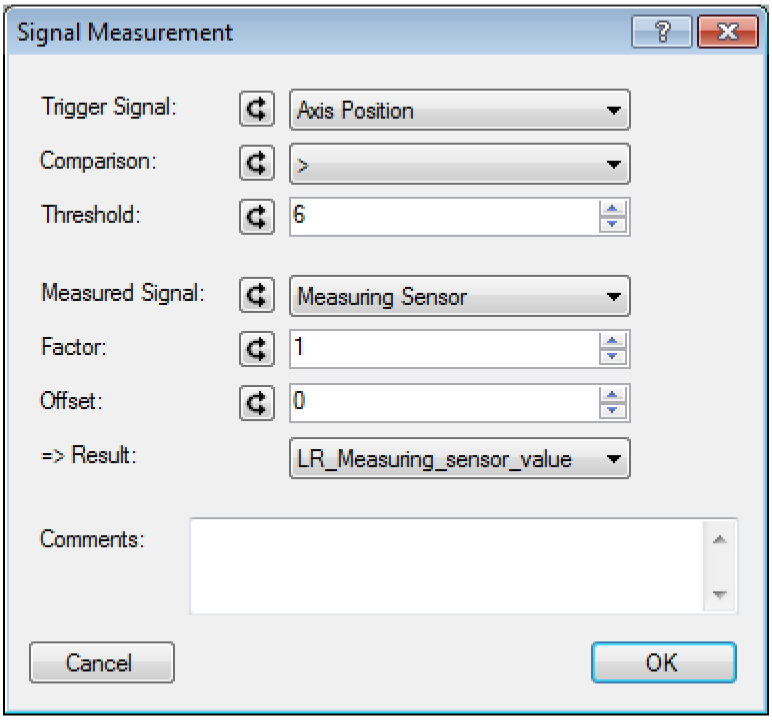


Figure 2: Example of programming the signal measurement control

Example of use:

We would like to record the position of the axis at which the press begins to exert force on the part.

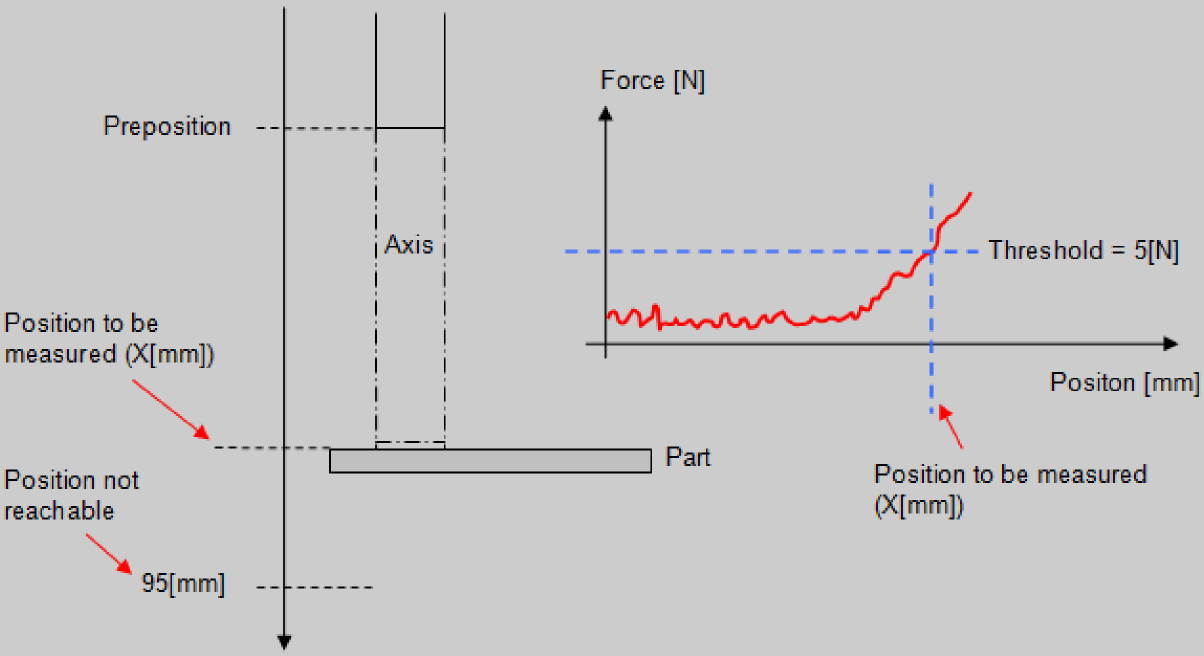
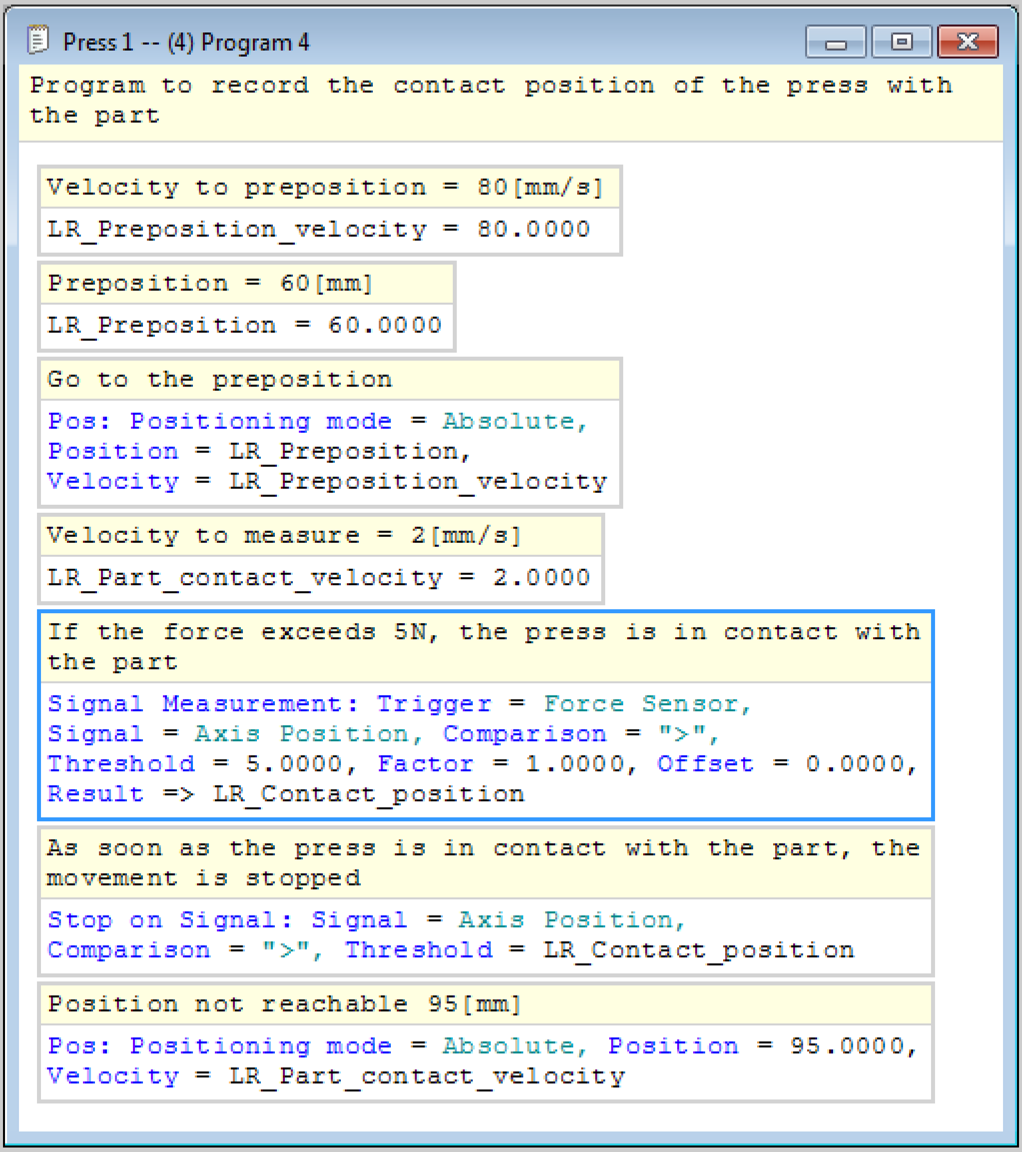


Figure 3: Schematic explanation of the operation

In the program below, we make a movement towards the preposition at high velocity, then a movement at low velocity to come into contact with the part (Force > 5[N]), at this moment we record the position of the axis and we stop the movement.



```
Press 1 -- (4) Program 4

Program to record the contact position of the press with
the part

Velocity to preposition = 80[mm/s]
LR_Preposition_velocity = 80.0000

Preposition = 60[mm]
LR_Preposition = 60.0000

Go to the preposition
Pos: Positioning mode = Absolute,
Position = LR_Preposition,
Velocity = LR_Preposition_velocity

Velocity to measure = 2[mm/s]
LR_Part_contact_velocity = 2.0000

If the force exceeds 5N, the press is in contact with
the part
Signal Measurement: Trigger = Force Sensor,
Signal = Axis Position, Comparison = ">",
Threshold = 5.0000, Factor = 1.0000, Offset = 0.0000,
Result => LR_Contact_position

As soon as the press is in contact with the part, the
movement is stopped
Stop on Signal: Signal = Axis Position,
Comparison = ">", Threshold = LR_Contact_position

Position not reachable 95[mm]
Pos: Positioning mode = Absolute, Position = 95.0000,
Velocity = LR_Part_contact_velocity
```

Figure 4: Part program to record the position of the axis when the force > 5[N]

"Stop on signal" control instruction

This instruction is used to stop the press axis as soon as a signal has exceeded a certain threshold (larger or smaller).

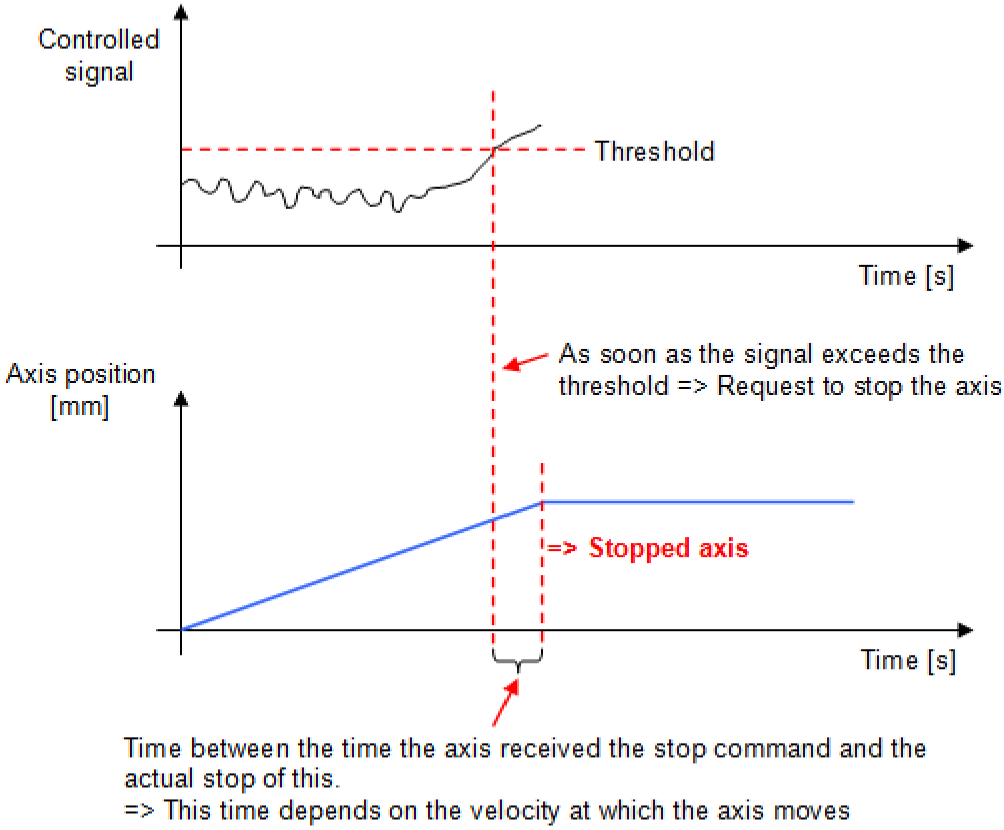


Figure 1: General operation of the signal stop control

The signals that can be controlled are as follows:

- Axis position
- Force sensor
- Measurement sensor signal

There are two types of trigger-to-threshold comparisons:

- Greater than ">": If the signal passes above the threshold => Axis stop
- Smaller than "<": If the signal falls below the threshold => Axis stop

The threshold value can come from a user variable or be given as a constant.

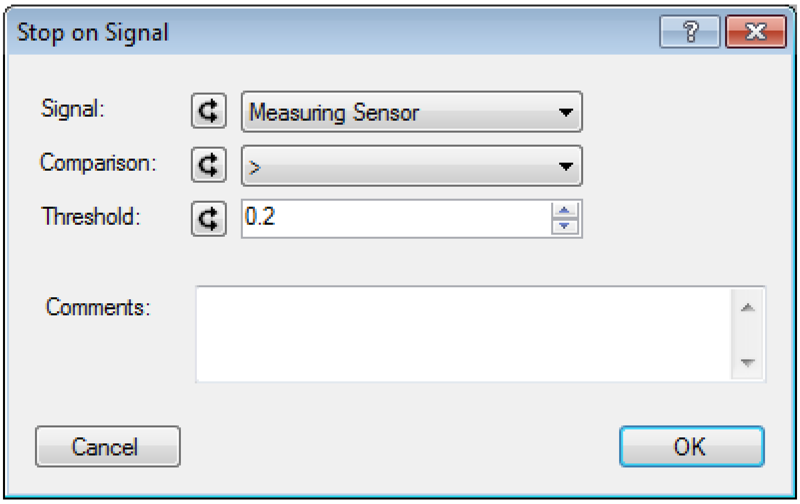


Figure 2: Example of programming a stop control on signal. Axis stop if measuring sensor position > 0.2[mm]

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Signal	Input	Enum [Source signal]	Axis position	Comparison signal, measuring sensor, axis position or force sensor
Comparison	Input	Enum [Comparison]	>	Comparison type ">" or "<"
Threshold	Input	LREAL	0	Threshold that the signal must reach to generate a stop of the axis

Table 1: List of instruction parameters

Example of use:

We want to detect a measuring sensor problem, if the measuring sensor position exceeds the threshold of 0.2[mm] before the axis is in preposition, we create an error.

If the value of the measuring sensor has not exceeded 0.2[mm] during the movement to the preposition, we move to the absolute position 25.4[mm] otherwise the jump instruction brings us to the end of the program.

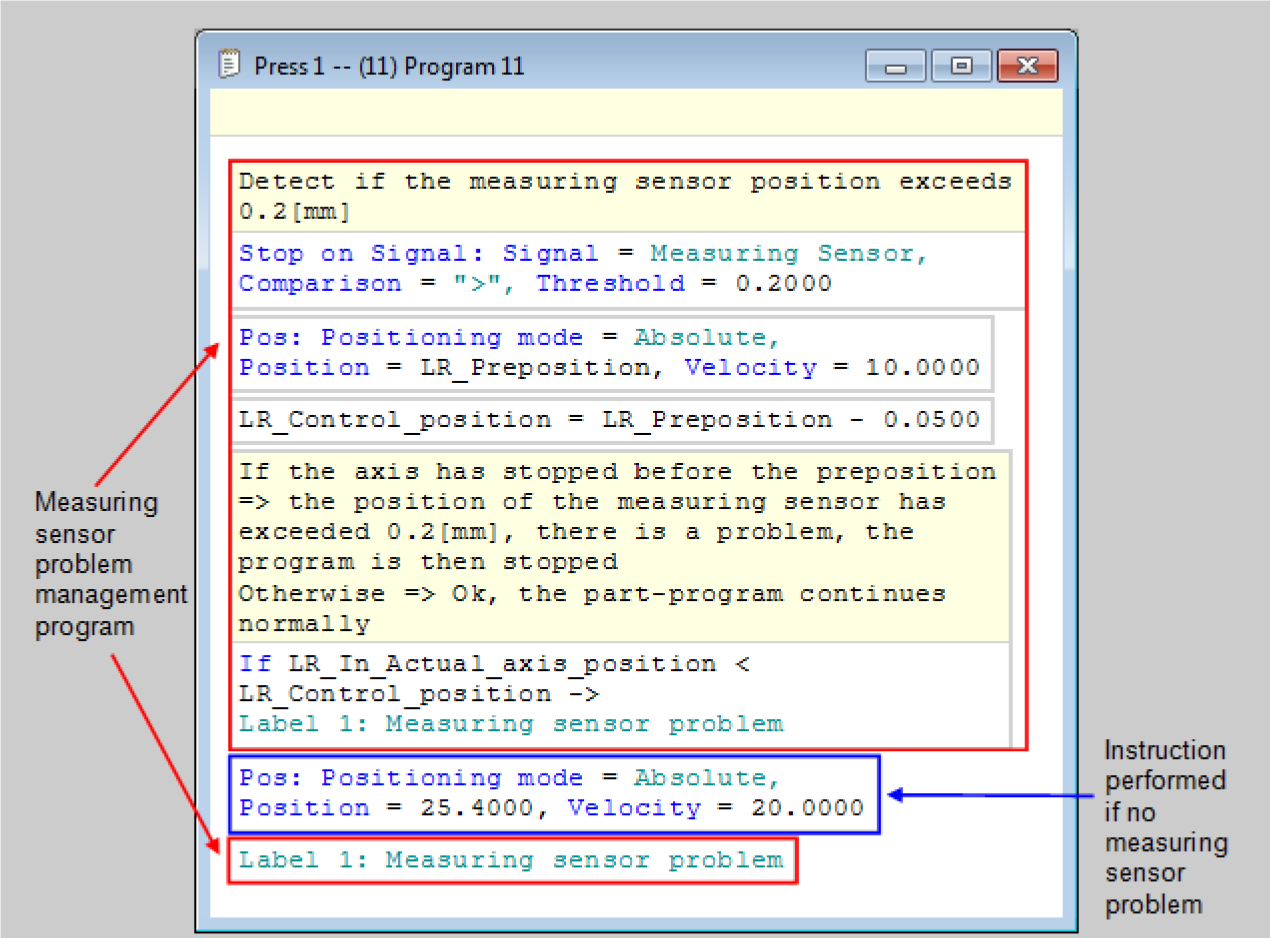


Figure 3: Using the signal stop function to detect a measuring sensor problem

"Curve recording" control instruction

This instruction is used to record the force values and position of the axis during positioning.

For each positioning of a part-program, it is possible to make this recording. Since all the records made in the same part-program are recorded in the same table, each record must be given a different "ID" (Identifier) in order to be able to differentiate them.

Since this "ID" is also recorded in the records, the minimum value that an "ID" can have is equal to 1e+6 (1 million).

It is possible to record the force and the position throughout the positioning, or only on a part of it (Recording mode = in a segment)

For each curve recording you must choose the maximum time that the positioning can last, if you increase the time it increases the sampling time, the minimum being 2[ms]. **Be careful, if you use several curve recording instructions in a program, the number of values recorded may exceed the maximum number.**

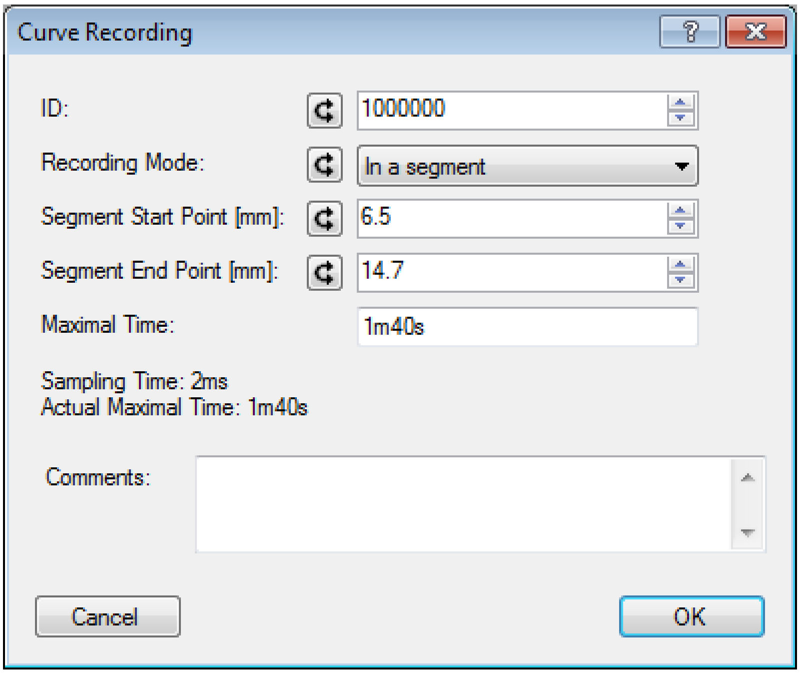


Figure 1: Example of programming a recording in a positioning area

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
ID	Input	REAL	0	Curve identifier
Recording mode	Input	Enum [In a segment or entire positioning]	Entire positioning	Recording mode, in a segment or entire positioning
Segment start point [mm]	Input	LREAL	0[mm]	Start position of the control zone

Parameter name	Declaration	Type of data	Default value	Description
Segment end point [mm]	Input	LREAL	0[mm]	End position of the control zone
Maximal time [ms]	Input	TIME	1m40s	Maximum time that the recording can last

Table 1: List of instruction parameters

To retrieve the curves from MecaMotion, simply go to the project tree structure and open the "control" window (see figure 2).

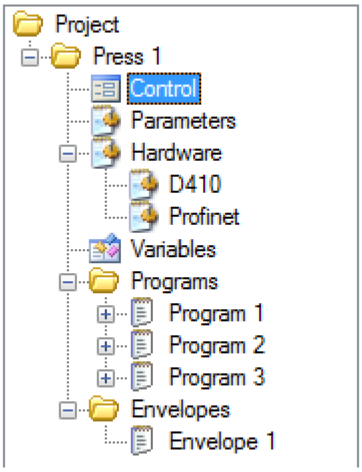


Figure 2: Project tree structure

In it, you must choose a production order and an operation in which the curves will be stored. Then run a program containing curve records and when it is finished the curves will be displayed in the force/position graph.

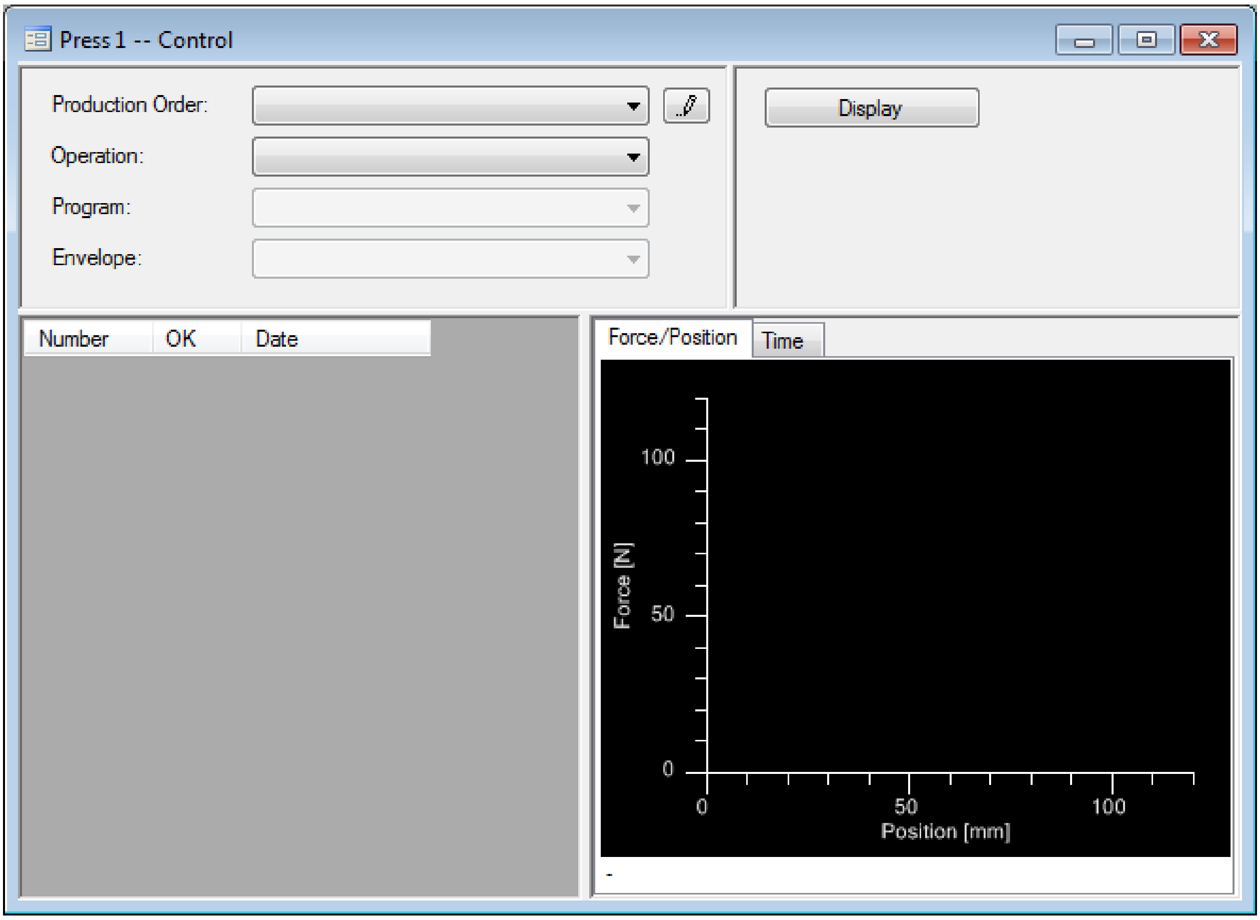


Figure 3: Curve reception window

Example:

Below is an example of how to use the curve recording control instruction.

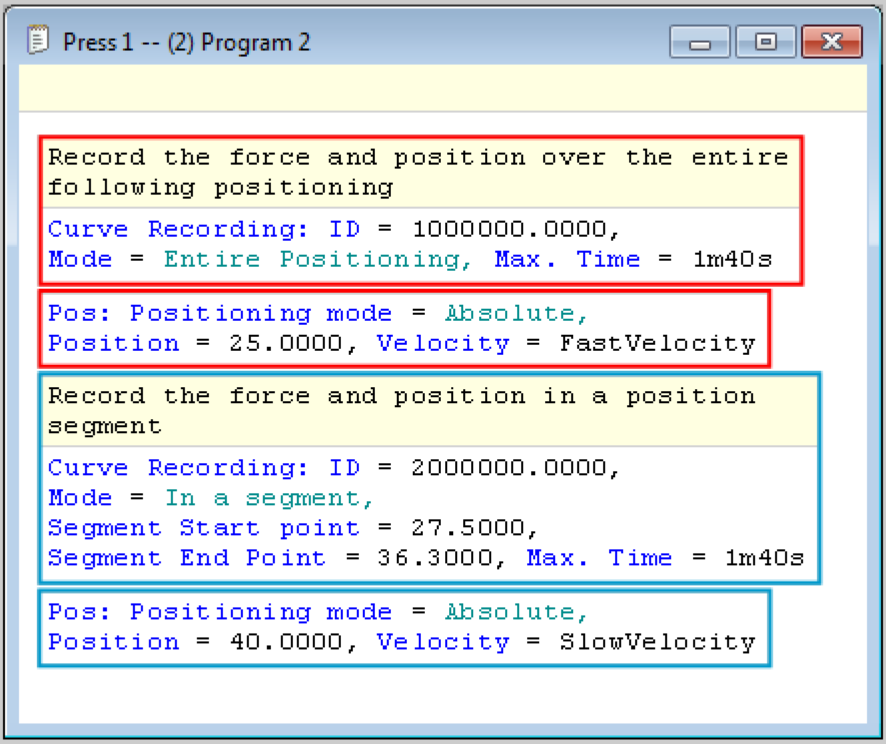


Figure 4: Example of programming the curve recording control

The first recording (circled in red) will be made throughout the next positioning with a velocity equal to the value of the "FastVelocity" variable (in LREAL format).

The second recording (circled in blue) will be made from position 27.5mm to 36.3mm during the next positioning with a velocity equal to the value of the "Slow Velocity variable (in LREAL format).

"Stop on force" control instruction

Unlike the "stop on signal" instruction, which stops the axis as soon as a signal passes a certain threshold, the "stop on force" control is dedicated to stopping the axis in relation to the force, with a regulation of the feed velocity in order to have, a final force very close to the setpoint given.

The regulator incorporated in this control is of type "P" (proportional). With the gain adjustable directly in the instruction.

This regulator works as follows:

- Every 2[ms] the value of the force sensor is read and compared with the setpoint.
- As the difference between the setpoint and the measured force decreases, the axis velocity will decrease (proportionally) until it reaches the minimum velocity (parameter "minimal velocity").

The regulator will start to operate as soon as the axis reaches the working position. At this point, the velocity will decrease by "X"[%] of its previous velocity (parameter "working velocity [%]") and after that, the velocity will be regulated.

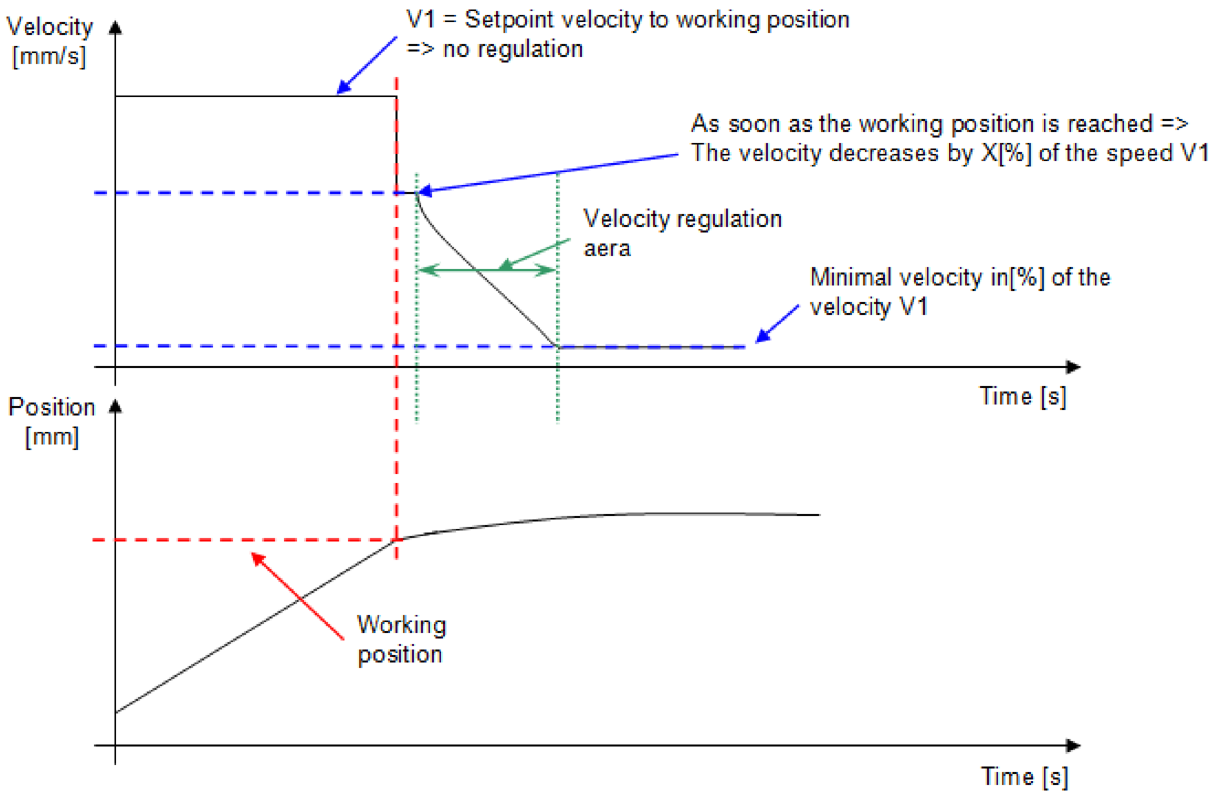


Figure 1: Graphical explanation of how the "stop on force" control works

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Working position [mm]	Input	LREAL	0[mm]	Position from which the feed velocity will first be reduced to X[%] of its initial value and then regulated
Force setpoint [N]	Input	LREAL	10[N]	Force setpoint to be reached
Velocity regulator gain [-]	Input	LREAL	1	Proportional gain of the regulator. The higher this value, the faster the approach velocity will decrease.
Minimal velocity [%]	Input	LREAL	10[%]	Minimum velocity that the regulation can reach, the velocity cannot go below this value. This one is given as a percentage of the initial axis velocity.
Working velocity [%]	Input	LREAL	50[%]	Percentage decrease in velocity from the initial velocity when the working position is reached.

Table 1: Summary of the different parameters of the instruction

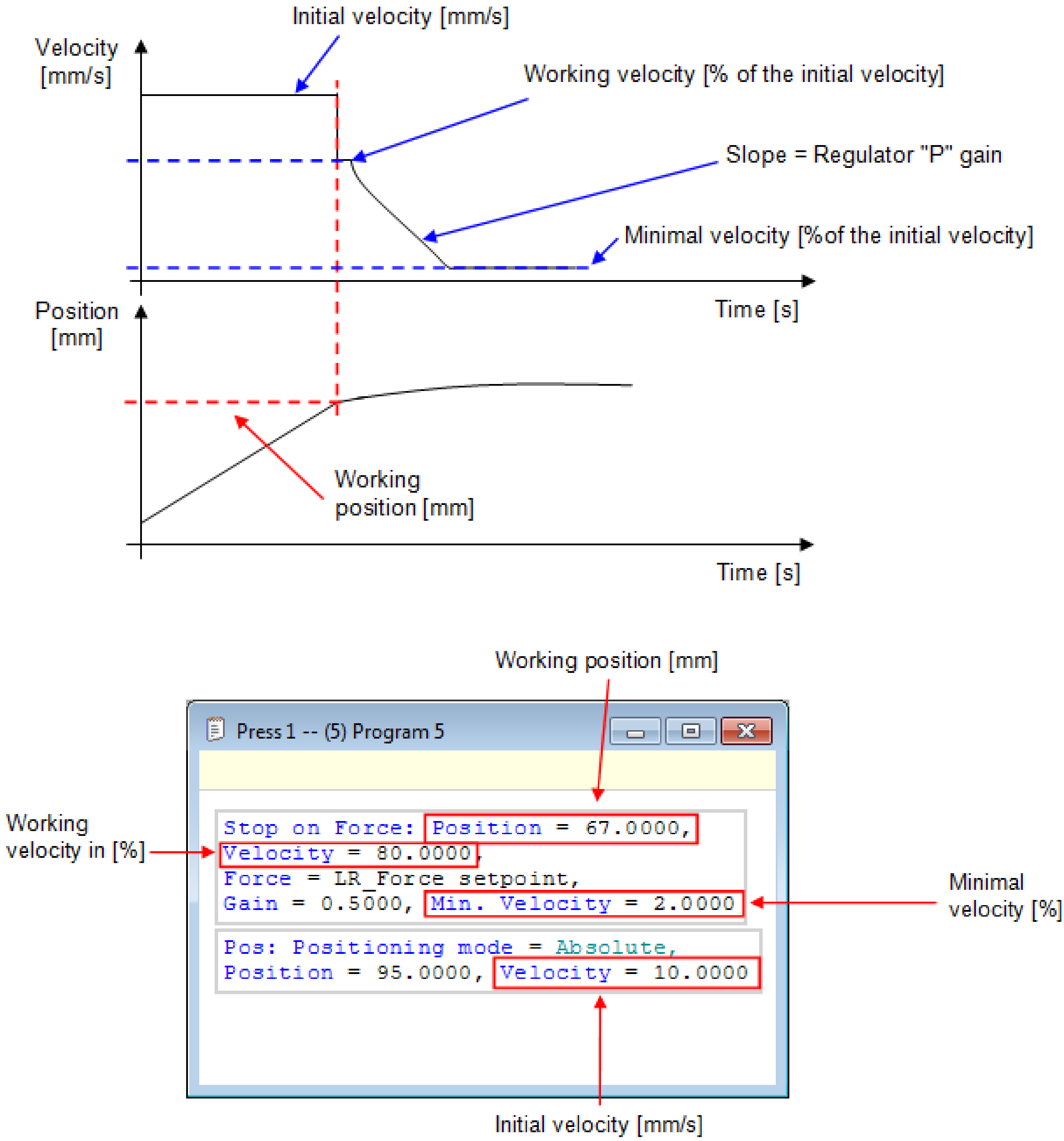


Figure 2: Programming window and graphic explanation

Below you will find the calculation of the working velocity, the initial velocity is equal to 10[mm/s].

- When switching to the working position, the axis velocity will change to:
  Working velocity = 80[%] of 10[mm/s] = 8[mm/s] working velocity
- The controller can reduce the axis velocity to:
  Minimum velocity = 2[%] of 10[mm/s] = 0.2[mm/s]

Example of use:

We want to insert one part into another with a force of 150[N]. The final force must be as close as possible to the set point, which is why it is not possible to use the signal stop control, which is much less accurate.

The gain, minimum velocity and working velocity parameters must be adjusted on test pieces, in order to be as precise as possible.

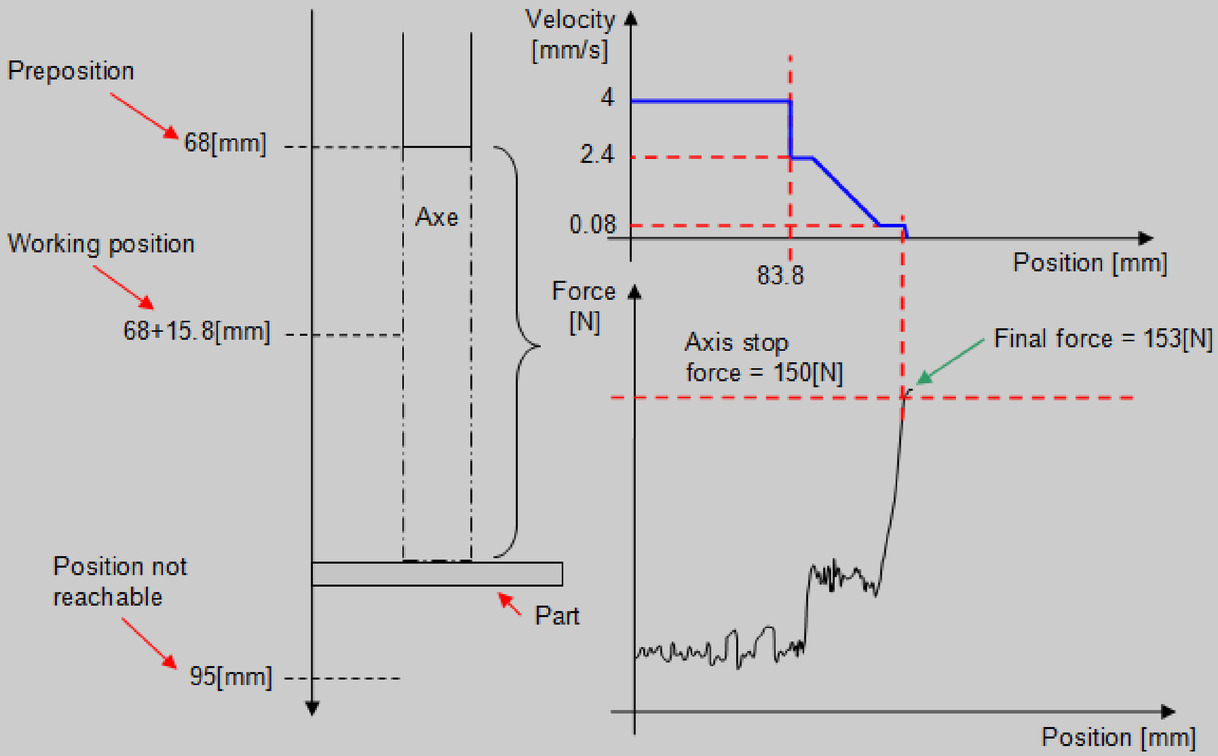


Figure 3: Graphical explanation of the application

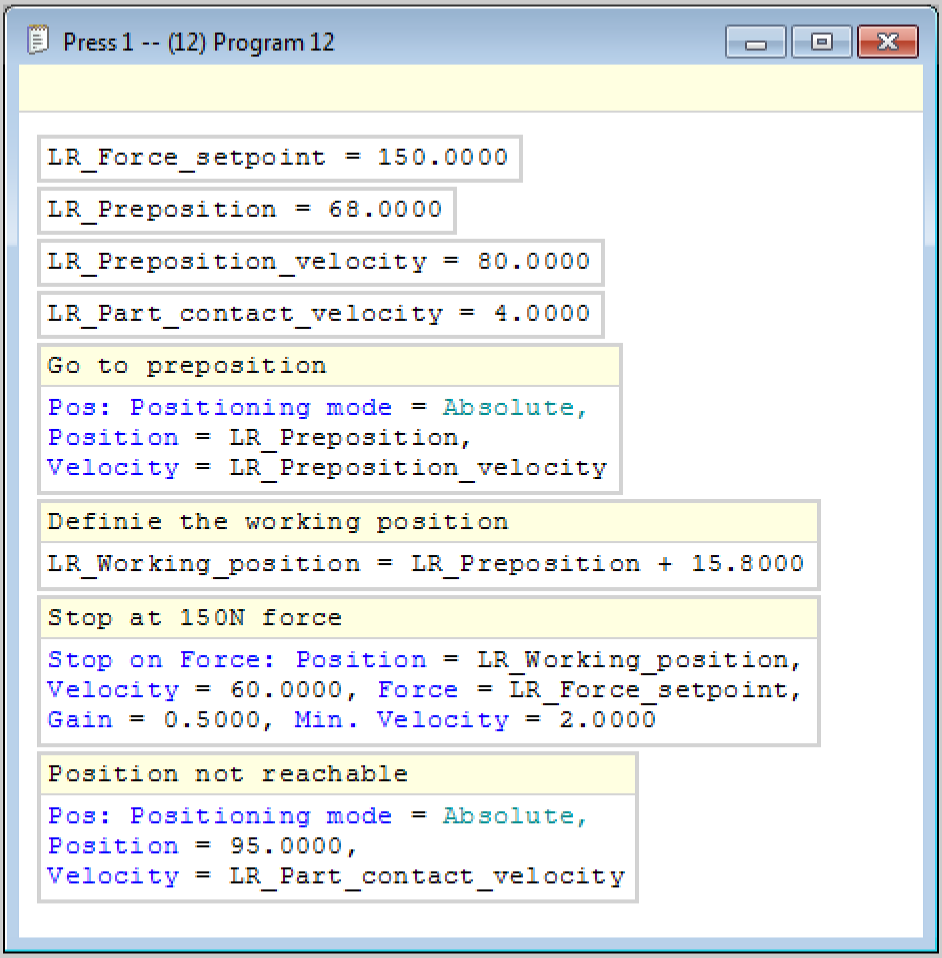


Figure 4: Programming a "stop on force" instruction

"Position measurement" control instruction

This instruction is used to record the position of the axis when the precision switch is operated.

The purpose is to be able to measure the reference of a part to perform the offset of the envelope or to have a starting position to achieve an insertion.

This function can be used with a maximum press velocity of 10mm/s.

Use of the instruction

This instruction must be before the positioning in which you want to measure.

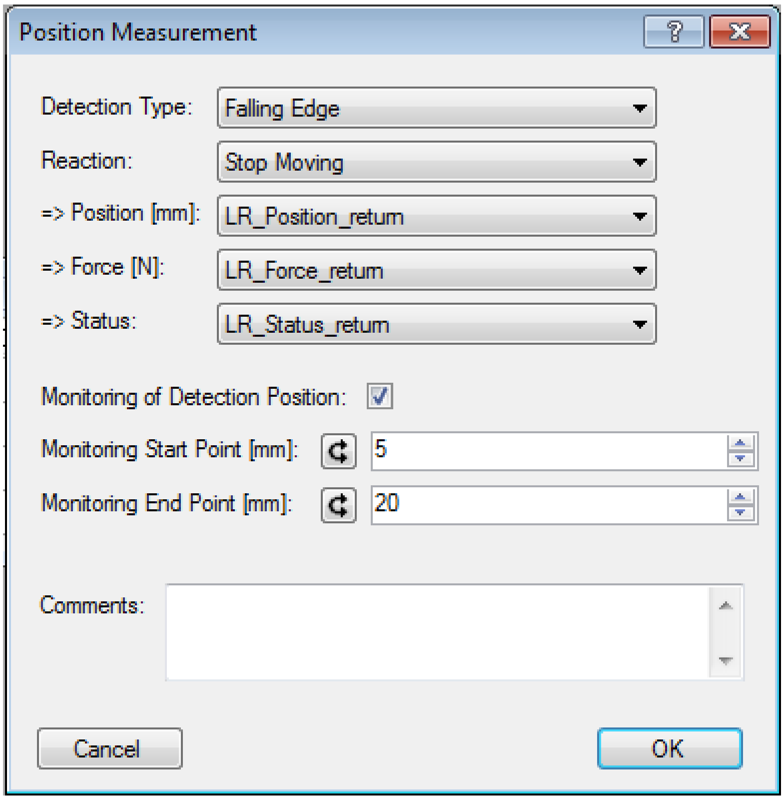


Figure 1: Parameters of the instruction position measurement

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Detection type	Input	--	Falling edge	Detection type, falling edge or rising edge
Reaction	Input	--	Stop moving	Reaction after measurement, stop moving or continue moving
Position [mm]	Output	LREAL	--	Position of the axis measured at the time of detection

Parameter name	Declaration	Type of data	Default value	Description
Force [N]	Output	LREAL	--	Force measured at the time of detection
Status	Output	Enum [Position measurement status]	--	Status of the measurement function (see description below)
Monitoring of detection position	Input	--	--	Perform detection only in a position area
Monitoring start point [mm]	Input	LREAL	0[mm]	Detection start position
Monitoring end point [mm]	Input	LREAL	0[mm]	Detection end position

Table 1: List of instruction parameters

Detail of the parameters to be filled in for this function:

- Detection type: Falling edge if you want to record the axis position when the sensor is pressed or rising edge if you want to record the axis position when the sensor is released (up axis).
- Reaction: Stop moving or continue moving after switching the sensor.
- Return variables : The position and force feedback variables must be of the "LREAL" type and the state feedback variable must be of the "position measurement status" type. The possible status are: 1 = Waiting for the sensor to switch (the measurement function is active), 2 = The measurement sensor has switched, 3 = The measurement is completed with error (the measurement could not be performed).

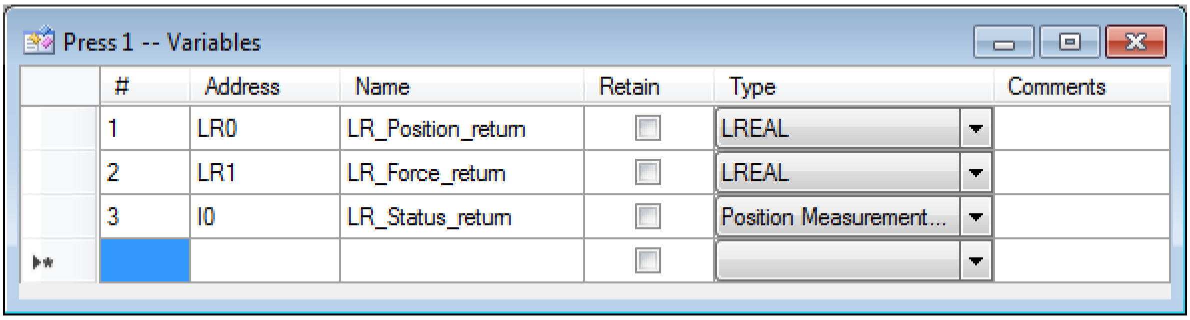


Figure 2: Declaration of variables

If necessary, you can perform this measurement within a certain position area of the axis. To do this, check the "monitoring of detection position" box and enter a start and end value for the area in[mm]. If you want to give the start and end position of the area using variables, they must be in LREAL format.

Example of use:

We want to measure a reference position with the precision switch.

We perform the measurement at low speed to obtain high accuracy.

In the example below, we move to the preposition at high speed and then measure at low speed. When the precision switch switches, the actual position of the axis is stored in the "LR_Reference_position" variable.

```
Press 1 -- (16) Program 16

LR_Preposition = 27.0000
Force Sensor: Mode = Reset, Range = 1
Wait Delay: 100ms
Force Sensor: Mode = Measurement
Rapid movement to preposition
Pos: Positioning mode = Absolute,
Position = LR_Preposition, Velocity = 50.0000
Measurement of the reference position
Position Measurement: Falling Edge, Stop Moving,
Status => Measurement_status, Force => LR_Contact_force,
Position => LR_Reference_position
Positioning during which the measurement is performed
Pos: Positioning mode = Absolute, Position = 55.0000,
Velocity = 2.0000
Returning to the initial position
Pos: Positioning mode = Absolute, Position = 0.0000,
Velocity = 20.0000
```

Figure 4: Example of using the position measurement instruction

"Post-process force measurement" control instruction

This instruction allows the value of the force to be recorded in a "result" variable "X"[mm] before the end of a positioning.

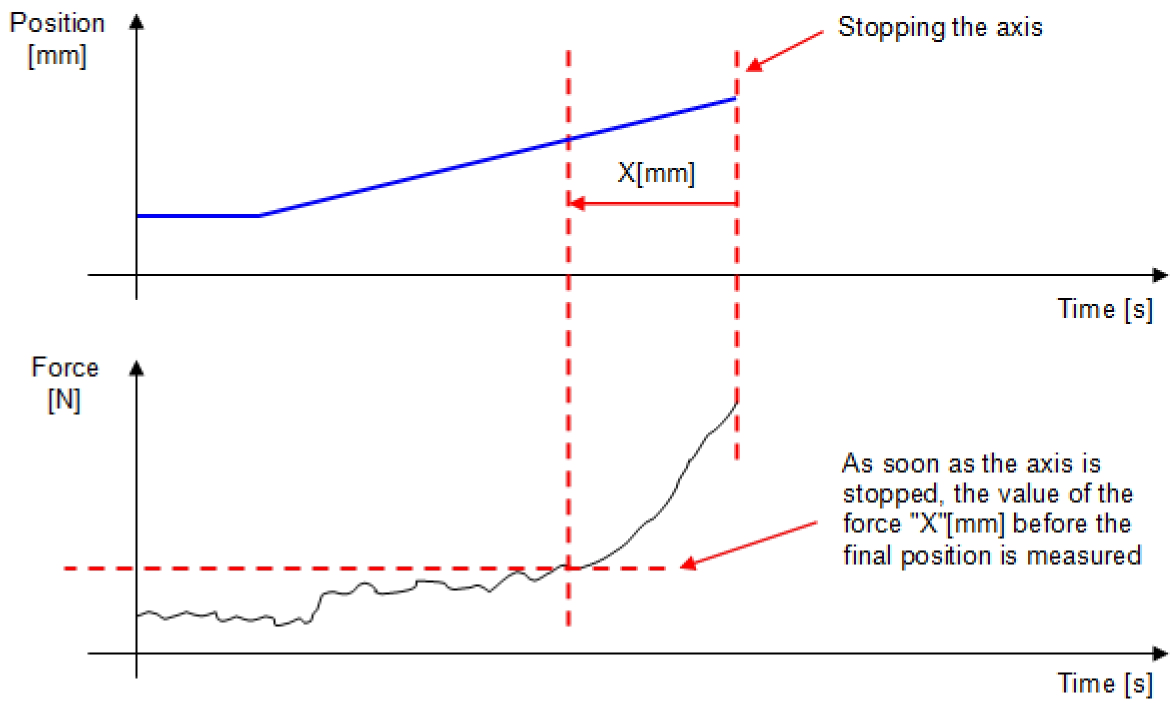


Figure 1: Schematic explanation of the instruction Post-Process force measurement

This instruction must be used in conjunction with the curve recording instruction. Indeed, the value of the force "X"[mm] before the final position will be searched in the registration table.

This "post-process" instruction, unlike other controls, comes **after the positioning** in which the measurement is to be performed.

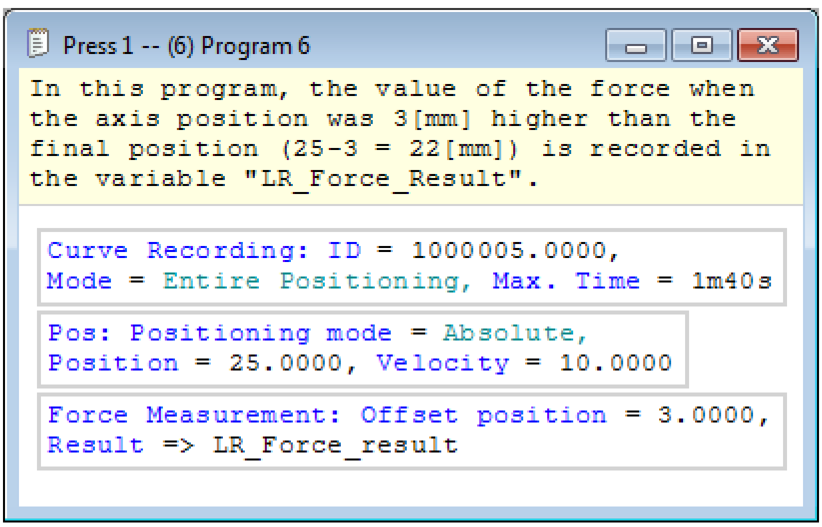


Figure 2: Example explaining how to use the "Post-process force measurement" control

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Position offset measurement [mm]	Input	LREAL	0[mm]	Distance before the final position where the force must be recorded
Result [N]	Output	REAL	--	Measured force X[mm] before the end position

Table 1: List of instruction parameters

Example of use:

We want to know the force exerted on a part, 0.05 millimeter before the final force of 150[N] is reached.

To do this, in the program in figure 4, we make a movement to the pre-position, then a movement with a stop on signal control if the force is greater than 150[N] and when this force is reached we stop the axis.

Then, the instruction "post-process force measurement" will read from the registration table the force exerted on the part 0.05 millimeter before the final force has been detected (axis stop).

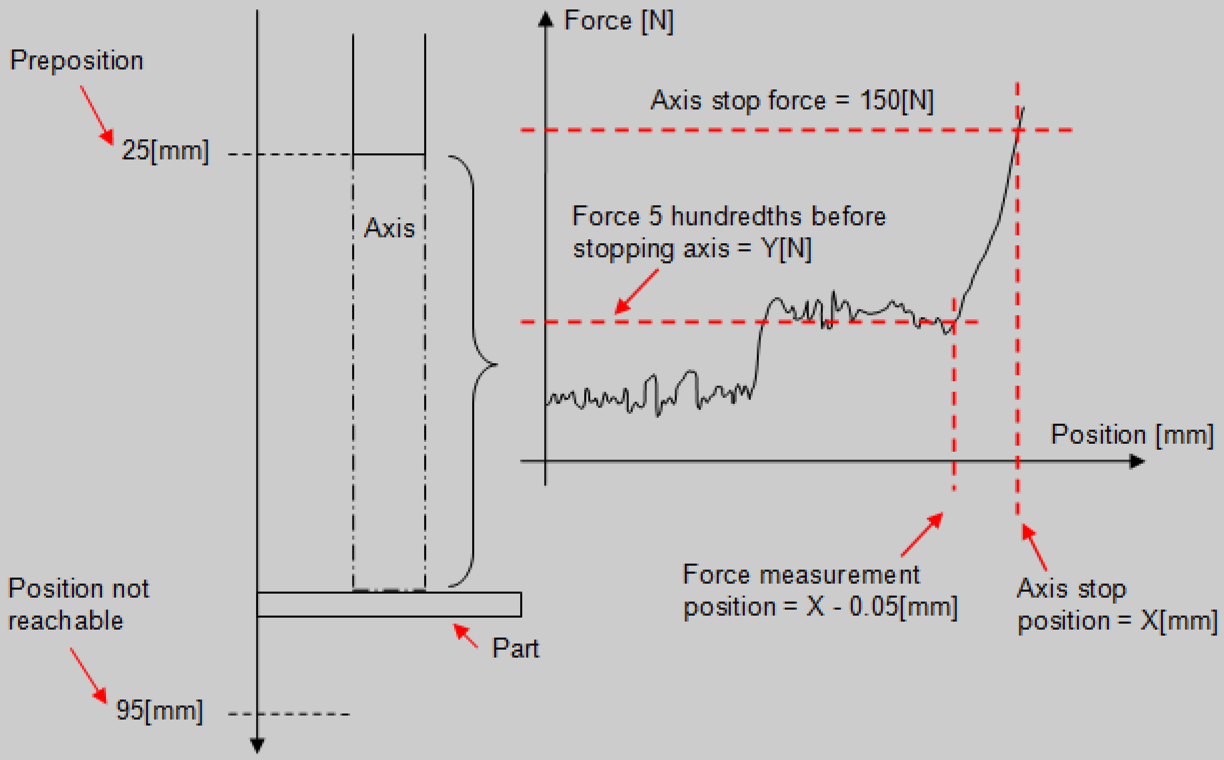


Figure 3: Graphical explanation of the example

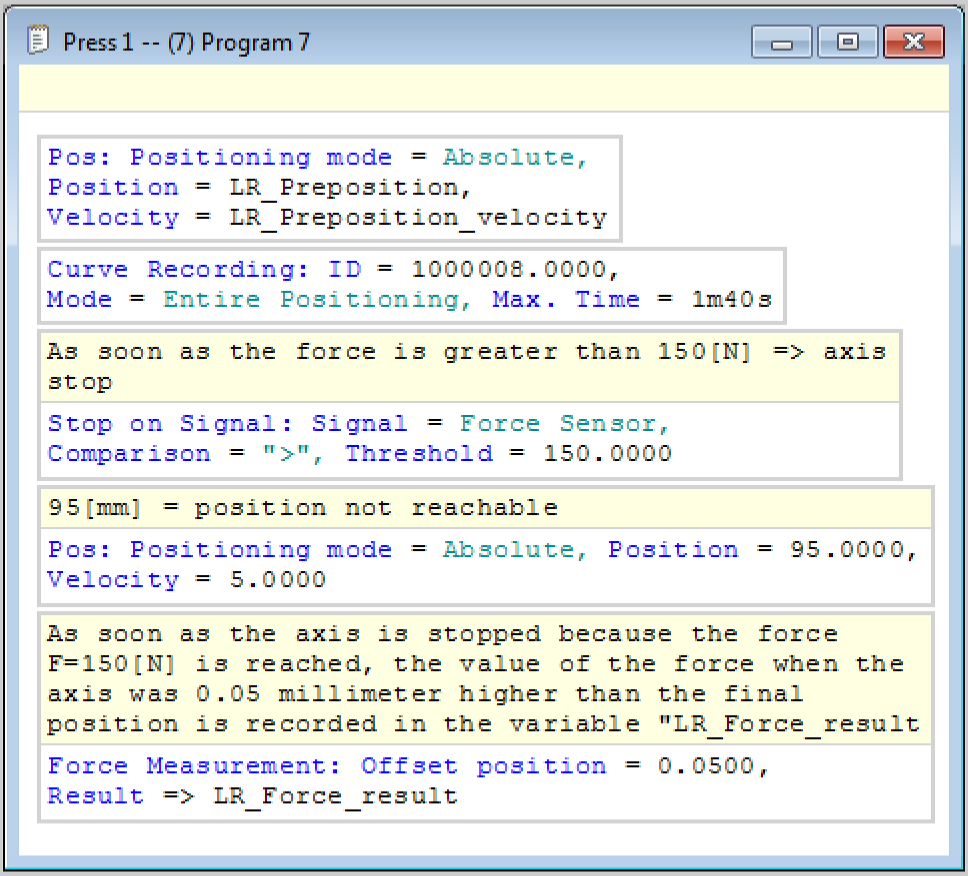


Figure 4: Part program to measure the force 0.05[mm] before the final position

"Breakpoint" instruction

This instruction is used to pause the execution of the part-program.

You can choose whether or not to pause the program execution at the breakpoints. In automatic mode, setting Profinet input bit n°240.2 to "1" allows pauses to be made at breakpoints, if you leave this bit at "0", pauses will not be made.

To continue the execution of the part-program after a breakpoint, you must activate the Profinet input bit n°240.3 for 50ms (rising edge detection).

In standalone mode, in the control window, you can choose by checking the "breakpoint" box to work with or without the breakpoints, in this mode, the start button allows you restart the execution of the part program.

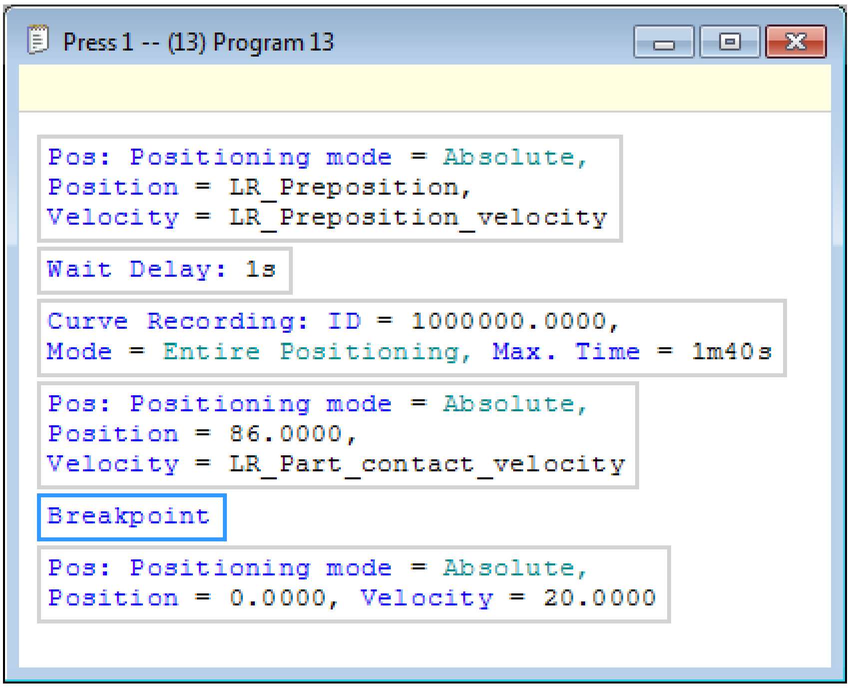


Figure 1: Example of using the breakpoint instruction

"Stopwatch" instruction

This instruction is used to measure a time during the execution of the part-program.

The time measured is accurate to two thousandths of a second.

It is possible to integrate up to 5 stopwatches into a part-program. These are numbered according to the position of the instruction that starts the stopwatch in the program, so stopwatch n°1 is the one that is placed at the top of the program, the stopwatch placed at the bottom will be n°2 etc.

The feedback variable of this stopwatch corresponds to the time that elapsed between the start and stop of the stopwatch. This time is returned in TIME format (in [ms]).

The reset function stops the stopwatch and resets its value.

Use of the instruction

To use the stopwatch instruction, you must first declare a variable of type "TIME" to be able to view the time value (see figure 1). You can activate the visualization using the "visualize" button on the main menu.

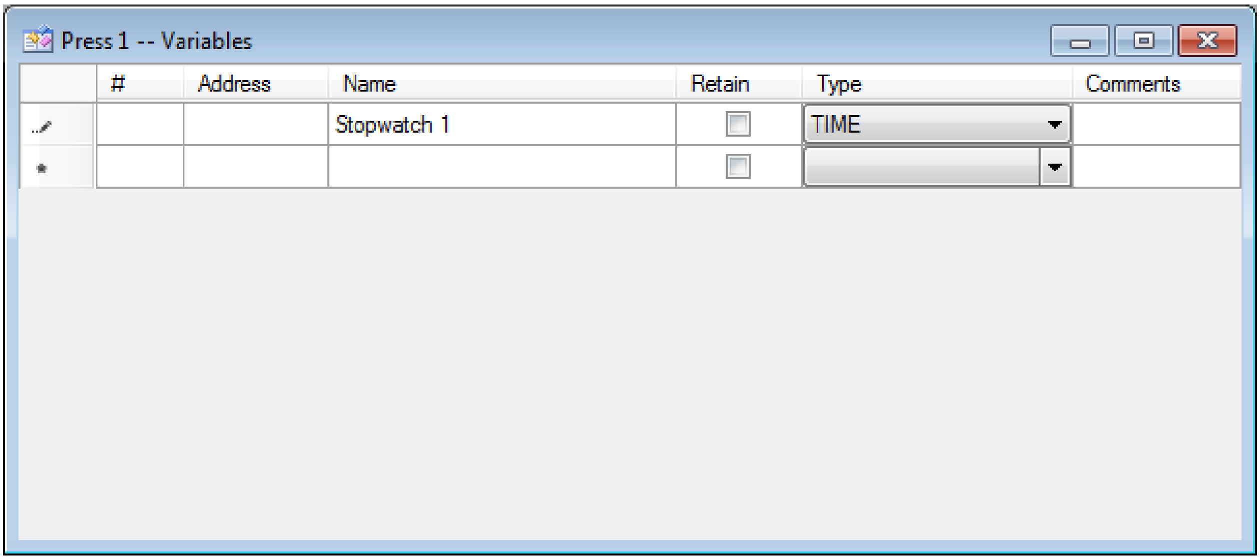


Figure 1: Declaration of the stopwatch feedback variable

Then, you must place the instruction in the part-program where you want to start, stop and reset the stopwatch.

You must also, in the variable line, select the "TIME" return variable created previously.

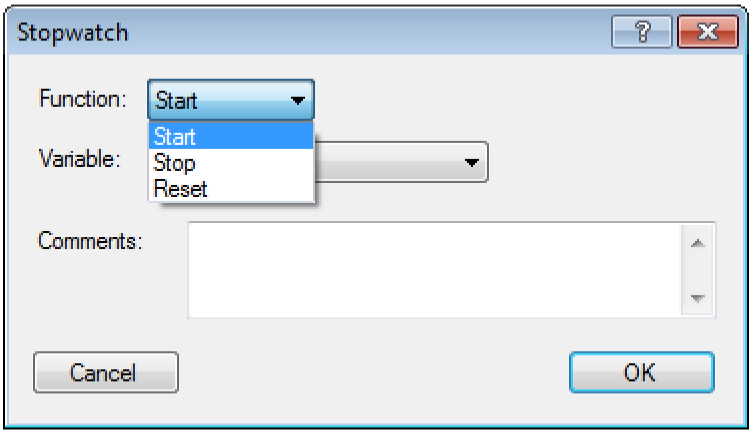


Figure 2: Using the stopwatch instruction

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Type of data	Default value	Description
Function	Input	--	Start	Start, stop or reset the stopwatch value
Variable [ms]	Output	TIME	--	Elapsed time

Table 1: List of instruction parameters

Warning, if no stop or reset is programmed, the chronometer value will continue to increase until its value reaches 2,000,000,000,000 ms.

Example of use:

In this example, we want to measure over several cycles of a program, the time required to achieve an absolute positioning of 2.5mm.

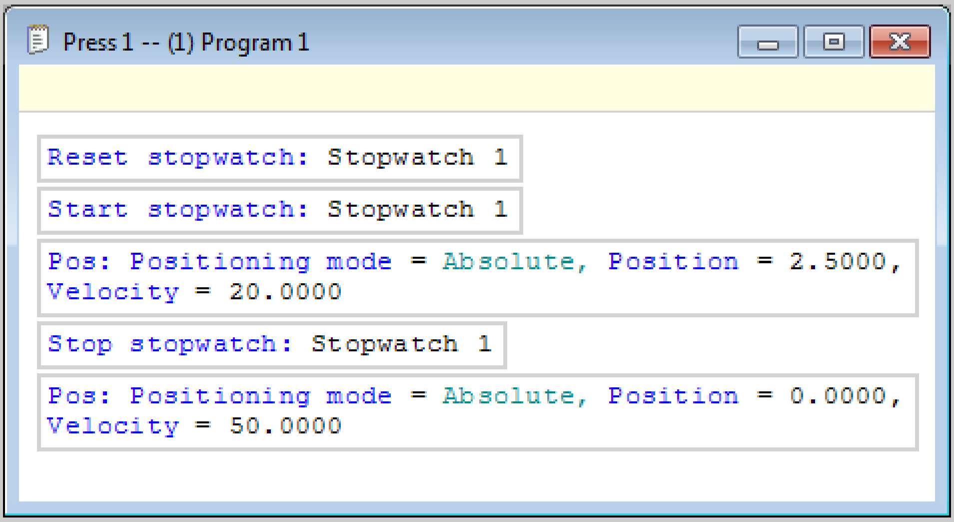


Figure 3: Using the stopwatch instruction

"Force sensor management" instruction

This instruction changes the range of the force sensor and reset the value of the force sensor.

Range 1 corresponds to the smallest range of the force value (example 0 to 150N) and range 2 corresponds to the largest range (example 0 to 1500N).

At the beginning of a program, you must reset the force sensor and choose the range you want to work with.

Use of the instruction

When you change the range, it is imperative to reset the force sensor.

If you use the instruction to switch consecutively to "reset" mode and then to "measurement" mode, you must insert a waiting delay of at least 100[ms] between the two. (figure 1)

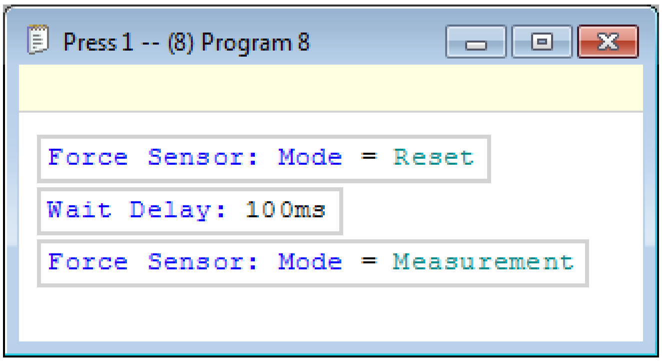


Figure 1: Reset/Measurement of the force sensor

Insert the instruction into the part-program and choose "reset" or "measurement" mode.

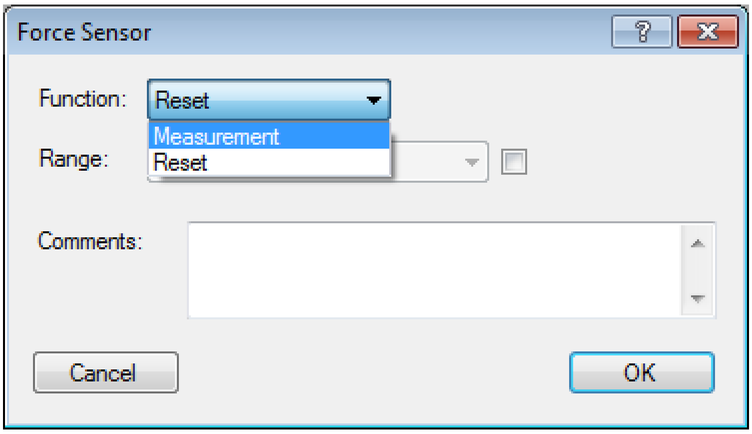


Figure 2: Force sensor measurement/reset

If you want to change the "range" you must check the box to the right of the field to enable the range selection. (see figure 3)

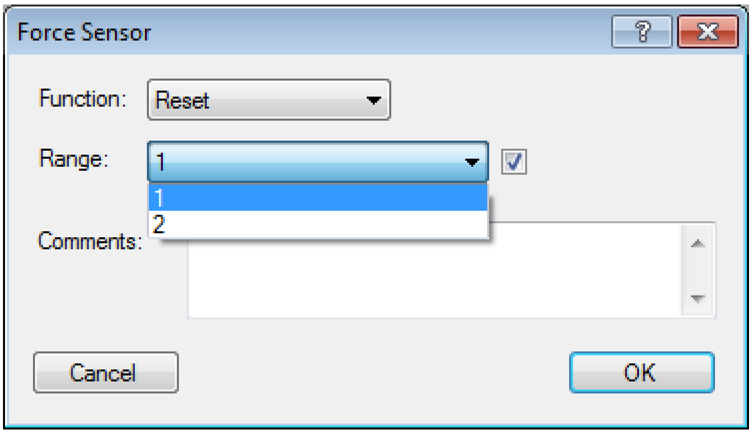


Figure 3: Selection of the force sensor range

Example of use:

In this example, we want to activate the "range 1" of the force sensor. The procedure for changing the force sensor range is as follows: 1) activate the reset mode, 2) wait 50ms, 3) change the range always in reset mode, 4) wait 100ms, 5) activate the measurement mode.

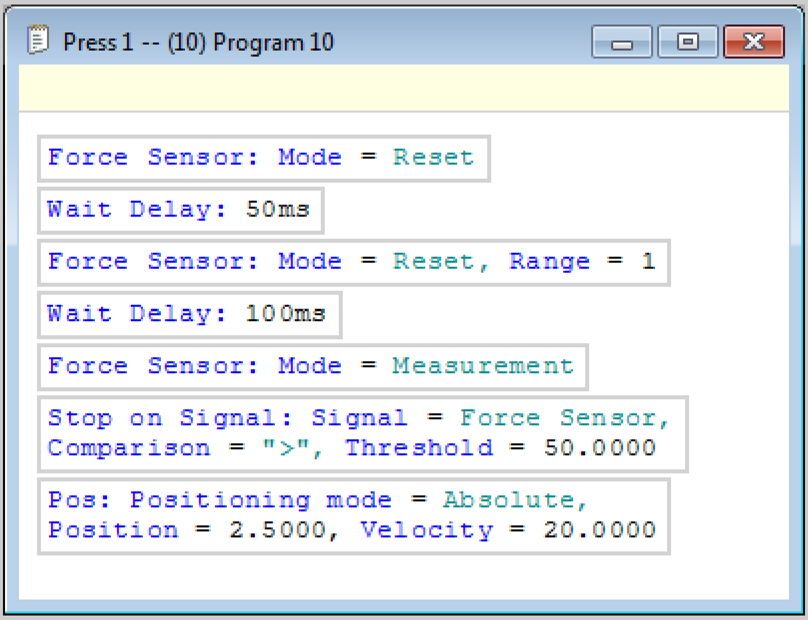


Figure 4: Use of the force sensor management instruction

"Clamping" instruction

This instruction allows you to apply a force with the press for a given time using the internal force value of the motor expressed in newton [N].

This instruction uses the torque limiting function of the motor to work, so the axis moves until it reaches the force setpoint and when it is reached the axis maintains this force.

Use of the instruction

To use the press in torque limiting, proceed as follows:

1. Insert a positioning instruction with a setpoint position lower than the position of the stop. The "Next_command" parameter must have the value "At_motion_start" or "Immediately". (By default, when you insert the clamping instruction, the "Next_command" parameter of the previous positioning will automatically take the value "Immediately")
2. Insert a clamping instruction, and in it, fill in the force with which the press should press against the stop. If you wish to give a force setpoint using a user variable, it must be in "LREAL" format.
3. If the press has to maintain the setpoint force for a certain time, you must use the "wait delay" instruction.
4. Insert the instruction "stop clamping" to stop the torque limitation.
5. To release the press, if you perform a positioning in relative mode, give a position with a negative sign and if you perform a positioning in absolute mode, give a position value higher than the position of the stop.

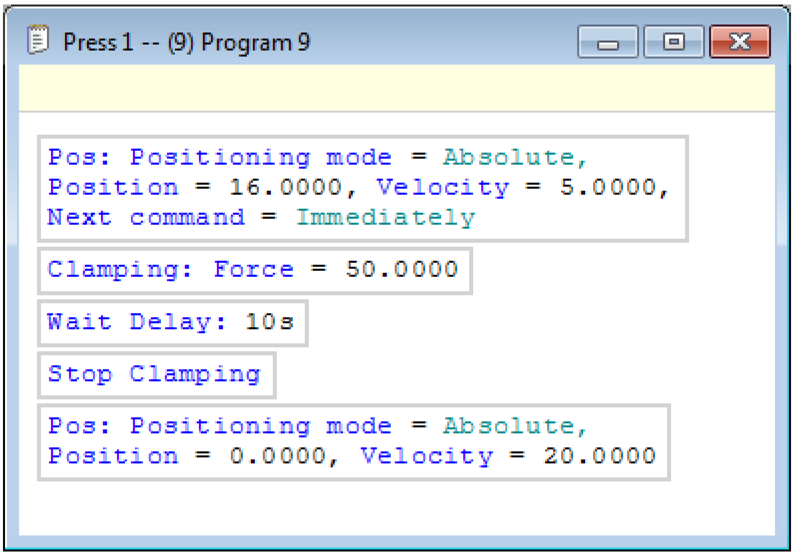


Figure 1: Example of a program for using the clamping

Below, the details of the instruction's input parameter:

Parameter name	Declaration	Type of data	Default value	Description
Force[N]	Input	LREAL	100[N]	Force setpoint of the clamping

Table 1: Instruction parameter

Cancel torque limitation

To stop the torque limitation, you must use the instruction "stop clamping".

"Values recording in an array" instruction

This instruction is used to record the value of a user variable for several cycles of a program.

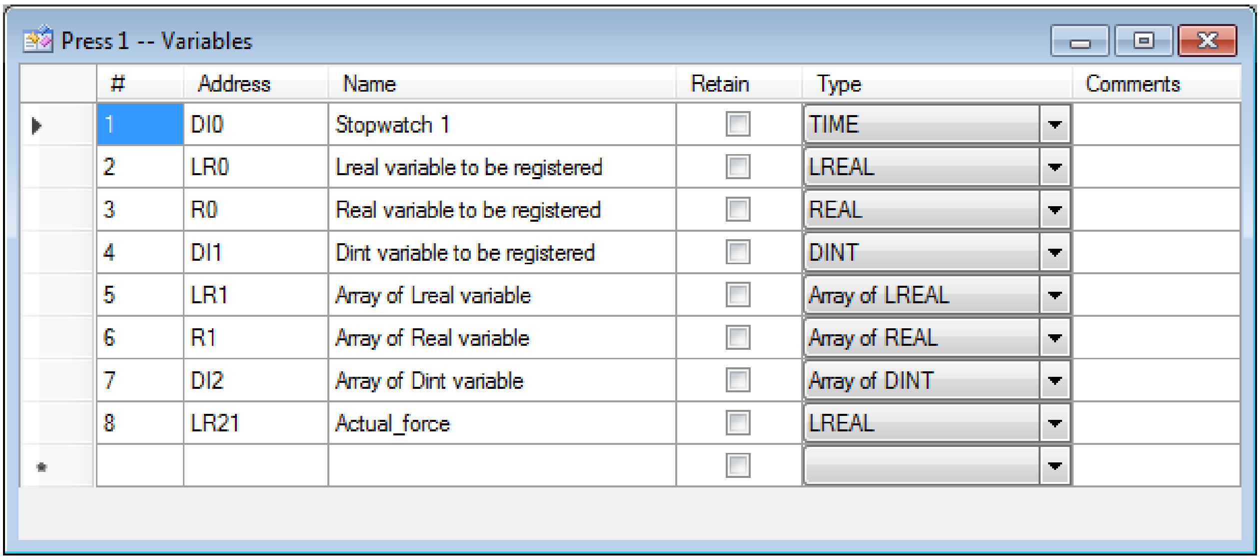
It is possible to register three types of variables: LREAL, REAL or DINT.

You can use this instruction only once per program and save a maximum of 20 values. (20 program cycles)

These values are returned in an array that is part of the user variables.

Use of the instruction

To start, you must create an array type user variable followed by the type of variable you want to save. (see figure 1)


Press 1 -- Variables

#	Address	Name	Retain	Type	Comments
1	DI0	Stopwatch 1	☐	TIME	
2	LR0	Lreal variable to be registered	☐	LREAL	
3	R0	Real variable to be registered	☐	REAL	
4	DI1	Dint variable to be registered	☐	DINT	
5	LR1	Array of Lreal variable	☐	Array of LREAL	
6	R1	Array of Real variable	☐	Array of REAL	
7	DI2	Array of Dint variable	☐	Array of DINT	
8	LR21	Actual_force	☐	LREAL	
*			☐		

Figure 1: Declaration of the type of variables in the array

Then insert the instruction into the part-program.

In this one, enter the variable you want to save and the array in which you want to save it. (figure 2)

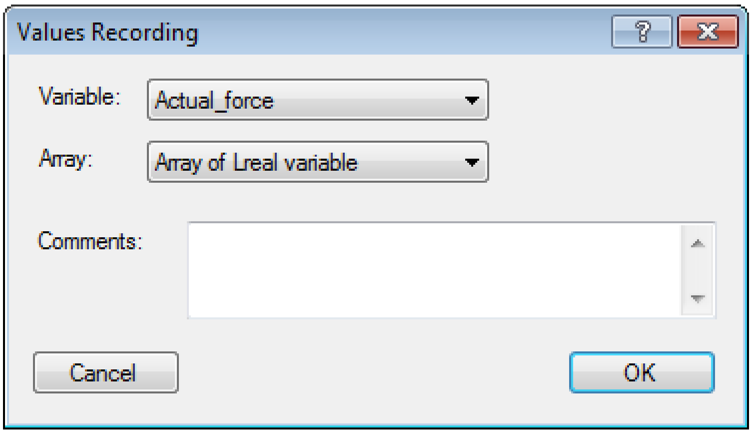


Figure 2: Using the instruction "values recording in an array"

Below is a list of the instruction's input and output parameters:

Parameter name	Declaration	Description
Variable	Input	Variable to be recorded
Array	Output	Array in which the values of the recorded variable are stored

Table 1: List of instruction parameters

In the variable declaration window, you can view in real time the values recorded in the array by clicking on the "view" icon in the main menu.

To reset the array values, right-click and select "Initialize array" (see figure 3)

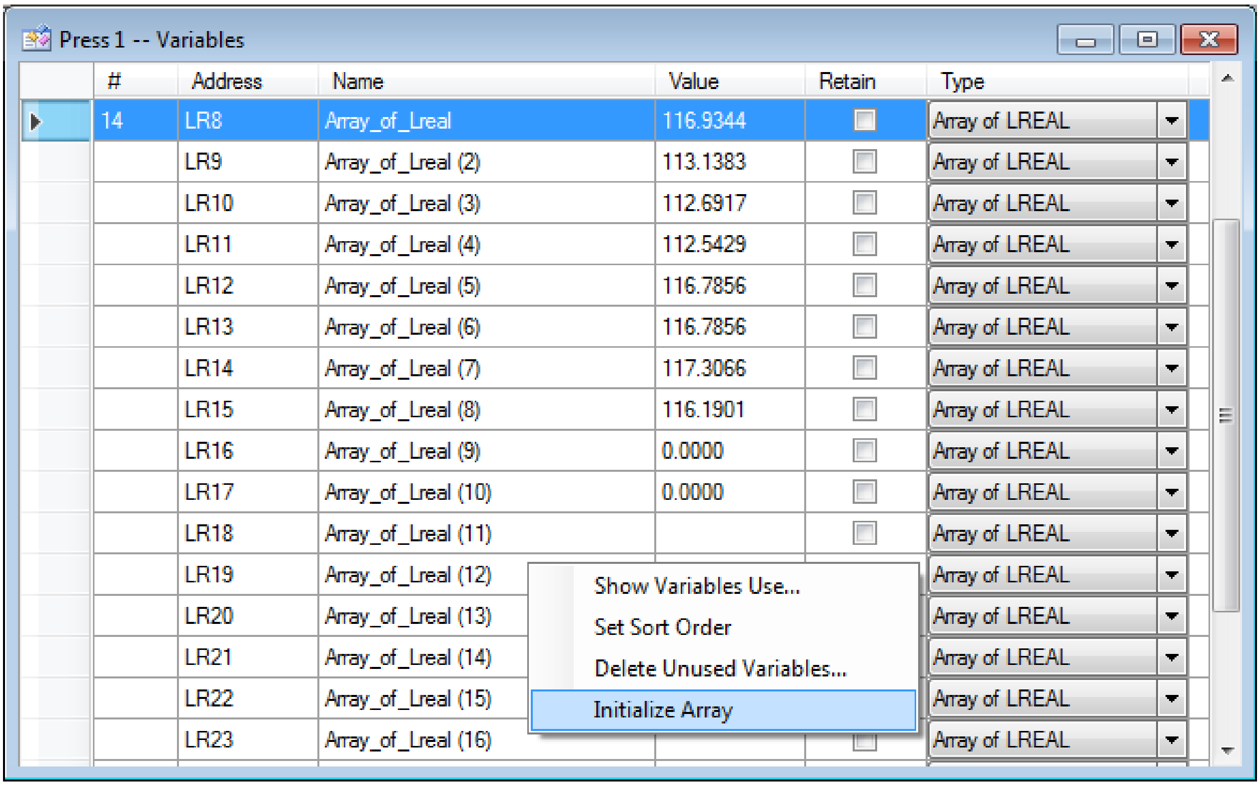


Figure 3: Viewing the values in the array

"Force regulator" instruction

This instruction allows a constant force to be applied with the press for a defined time.

This force is measured with the external force sensor.

Use of the instruction

This instruction must be used when the press is in mechanical prestressing.

It can therefore be carried out after an "stop on force" instruction or after a positioning that brings the press to prestressing.

Standard parameters of the instruction:

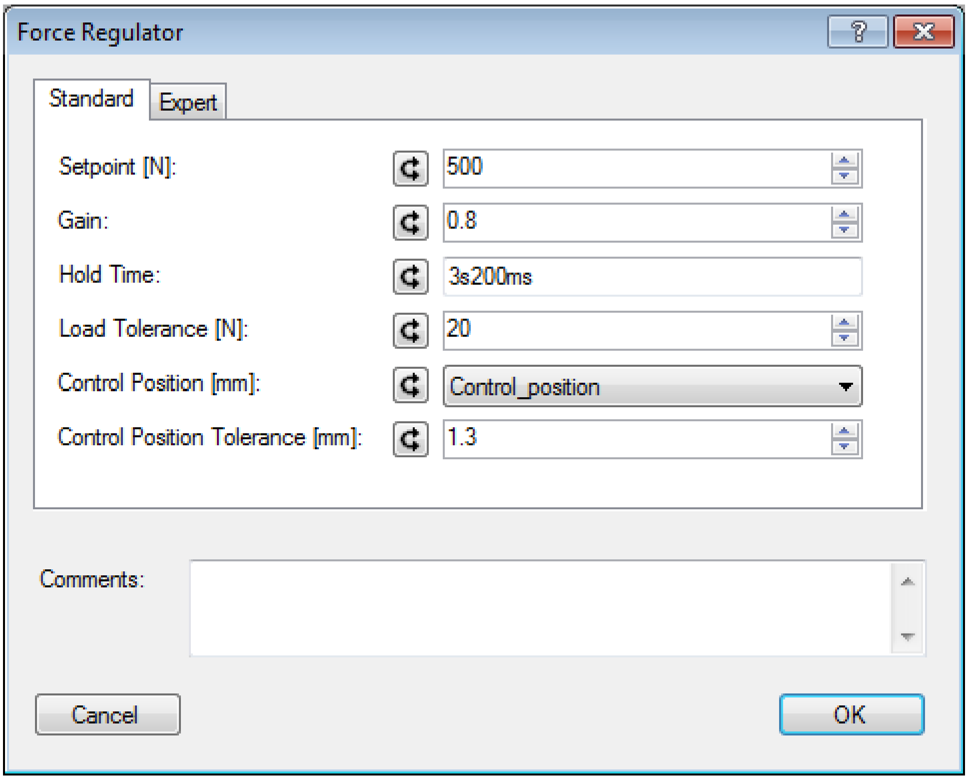


Figure 1: Standard parameters of the force regulation instruction

The standard parameters of this instruction are as follows:

Parameter name	Declaration	Type of data	Default value	Description
Setpoint [N]	Input	REAL	1[N]	Force setpoint to be applied
Gain	Input	REAL	1	Proportional gain of the regulation function
Hold time [ms]		TIME	100[ms]	Holding time of the applied force

Parameter name	Declaration	Type of data	Default value	Description
Load tolerance [N]	Input	REAL	0.5[N]	Tolerance +/- of the force setpoint in which the axis must be located at the end of the regulation so that there is no error
Control position [mm]	Input	LREAL	0[mm]	Position that the axis must have at the end of the regulation. If position = 0 no control
Control position tolerance [mm]	Input	LREAL	1[mm]	Tolerance +/- of the position in which the axis must be at the end of the regulation so that there is no error

Table 2: List of standard parameters of the instruction

If necessary, it is possible to refine the regulation settings using the expert parameters.

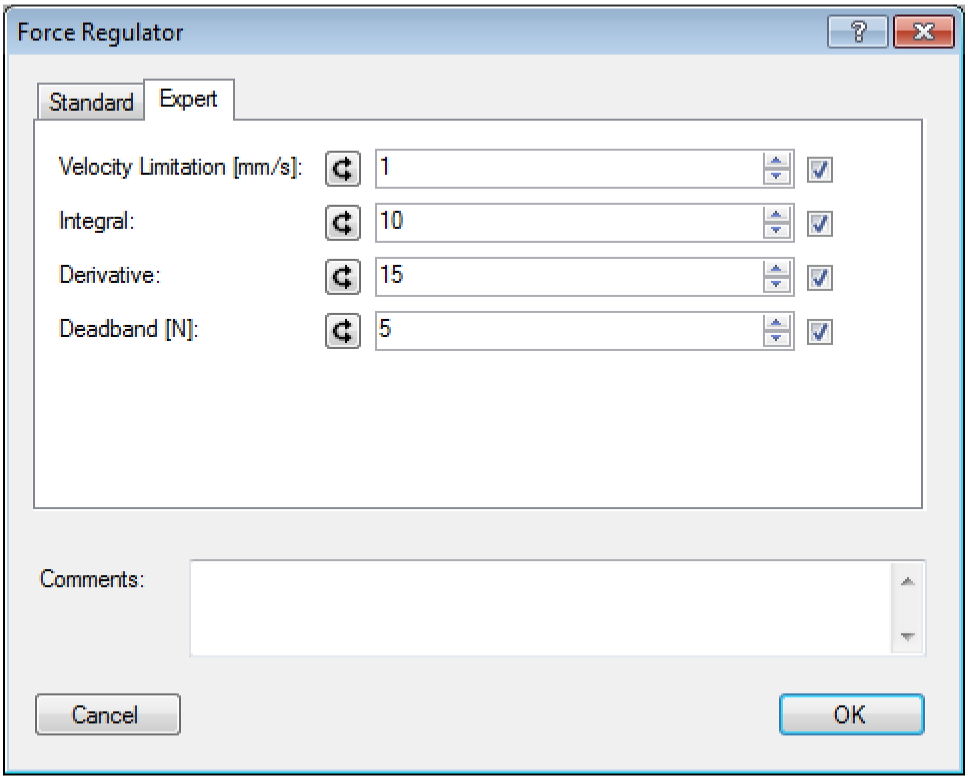


Figure 2: Expert parameters of the force regulation instruction

The expert parameters available for this instruction are as follows:

Parameter name	Declaration	Type of data	Default value	Description
Velocity limitation [mm/s]	Input	REAL	1[mm/s]	Maximum velocity that the axis can reach when it performs the regulation
Integral	Input	REAL	1[ms]	Integral of the regulation function

Parameter name	Declaration	Type of data	Default value	Description
Derivative	Input	REAL	1[ms]	Derivative of the regulation function
Deadband [N]	Input	REAL	1[N]	Zone +/- of the force set point in which the axis regulation is stopped

Table 2: List of expert parameters of the instruction

Results of the instruction

When the instruction is completed, the position and force measured at the end of the holding time are returned. (In progress)

Errors in the instruction

The possible errors for the force regulation instruction are as follows: (Errors returned by Profinet)

- Error N°24: Position out of tolerance
- Error N°25: Force out of tolerance

Example of use:

In this example, we perform a positioning with a force stop to bring the press to prestressing, then we apply a force of 500N for 3.2s.

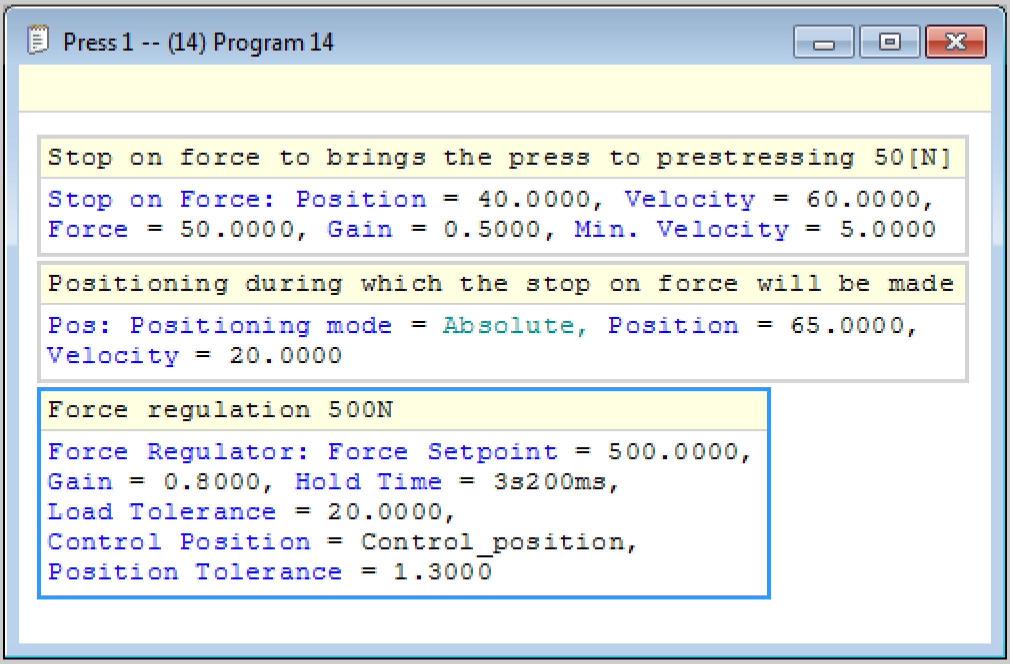


Figure 3: Use of the force regulation instruction

Standalone mode

The "Standalone" mode allows to work with the press without PLC.

All the commands required to control the press are then gathered in MecaMotion (press programming software).

Switch to "Standalone" mode

To operate the press in "Standalone" mode, you must modify parameter n°28 (communication) of the press and choose "Standalone". (see figure 2)

The parameters window is accessible from the project tree. (see figure 1)

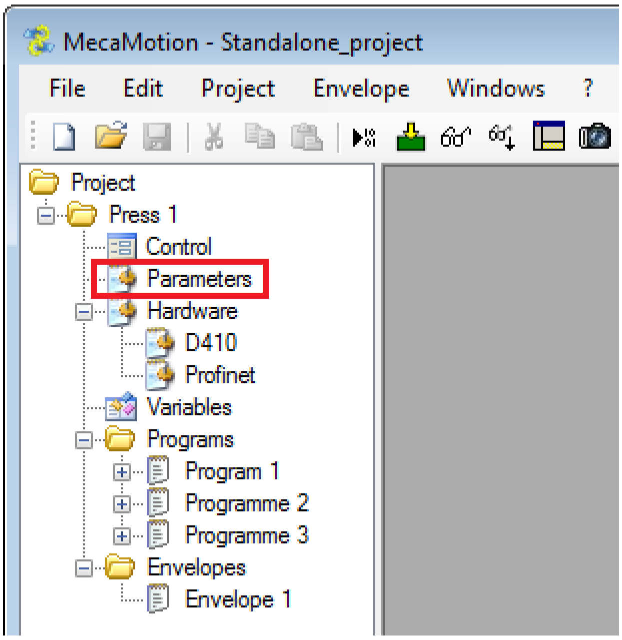


Figure 1: Project parameters

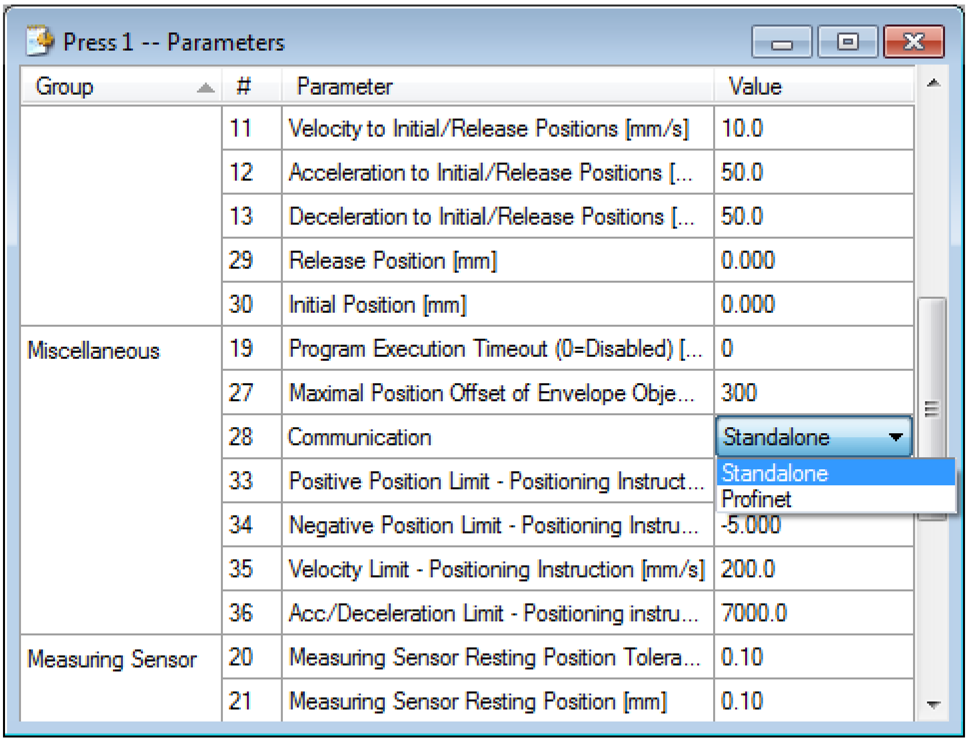


Figure 2: Standalone mode project parameters

Parameters and results in a program

When working in "Standalone" mode, it is possible, for each program, to add variables to enter parameters and variables to receive results. These variables can then be used directly in the part-program.

You can also define a return variable to view the good/bad parts and whether or not you want to display the envelope on the graph in the control page. (see figure 4)

In the project tree structure, the "parameters/results" window is located below the program. (see figure 3)

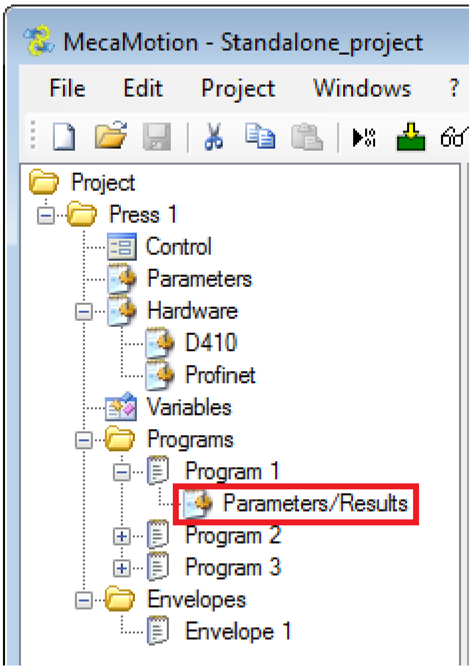


Figure 3: Parameters/results

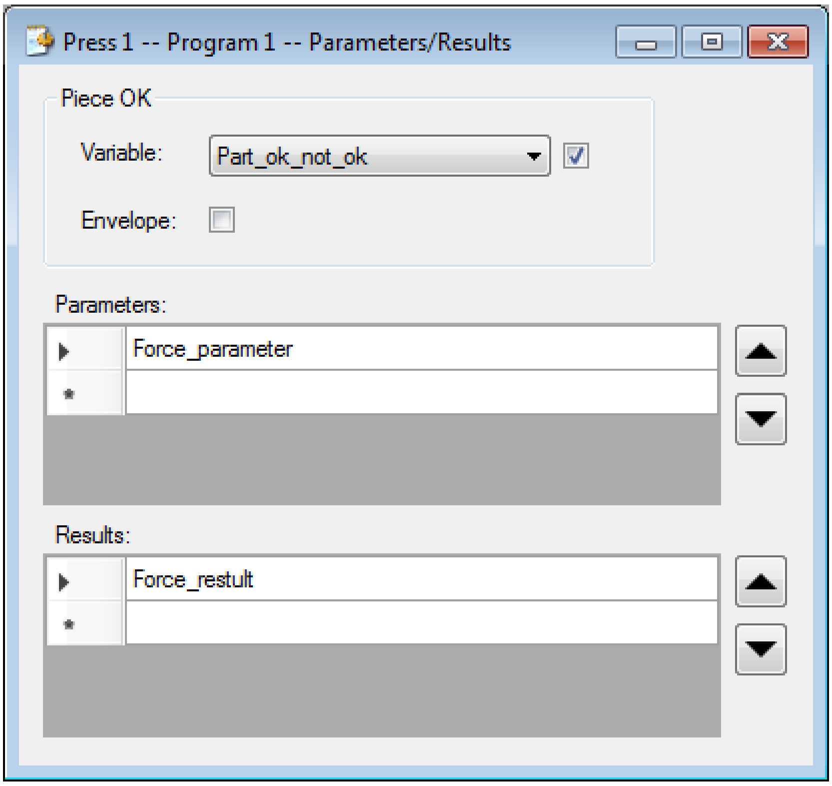


Figure 4: Parameters/results window

Once you have completed a cycle, you will find the results variables in the results table (bottom left of figure 5).

In the window below, you will find the different values of the result variable "Force_result" measured over several cycles.

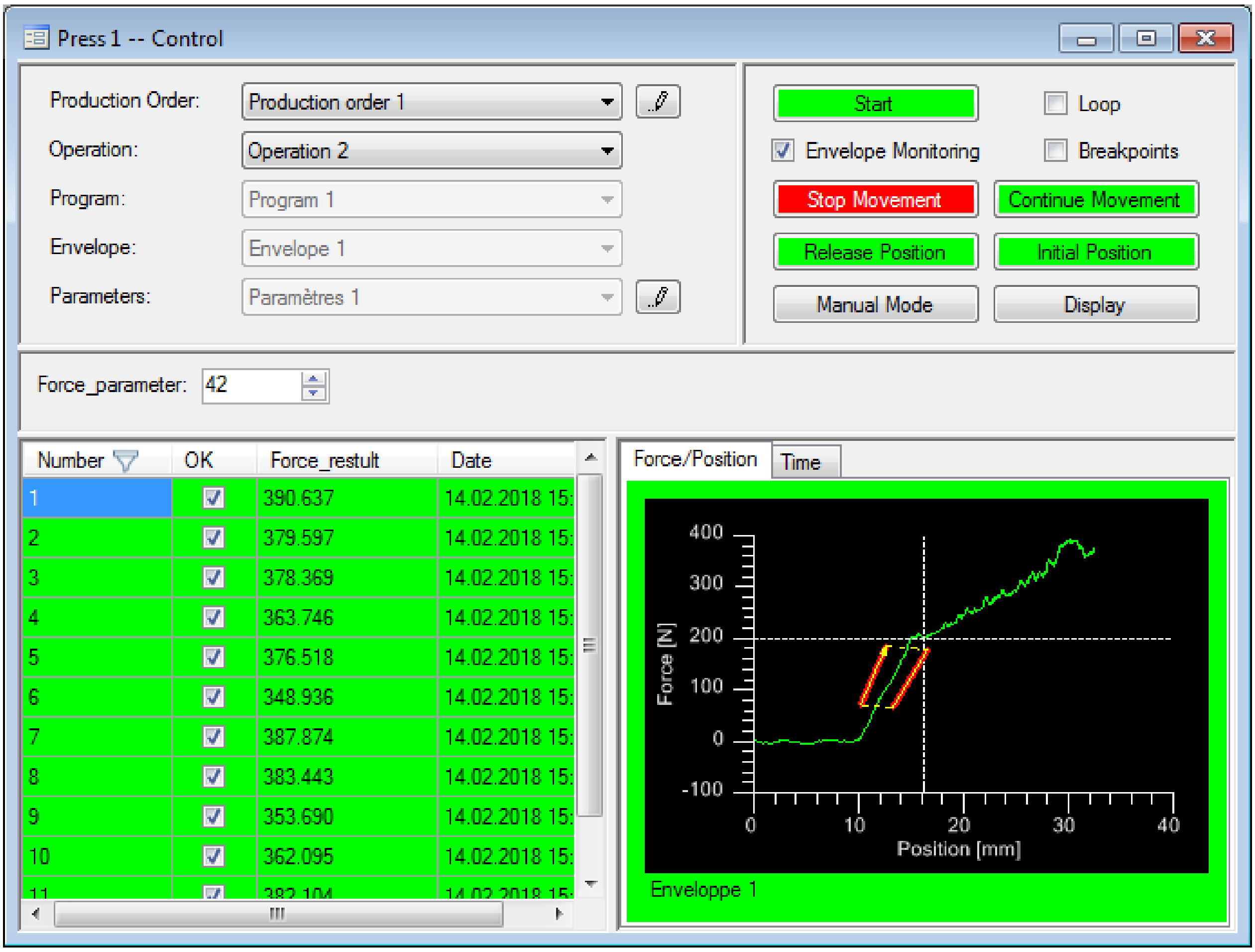


Figure 5: Viewing the value of the results

Press control

In "Standalone" mode, the press is controlled from the "control" window which is accessible in the project tree structure.

In this window, you can select and create production orders, enter parameter values and view the results. (see figure 6)

When you change an input parameter of the program you are going to start (example: Force_parameter), you are not required to load the project, the new value of the parameter will be sent when the program is started.

Important, in order to be able to save the results, the database must be active.

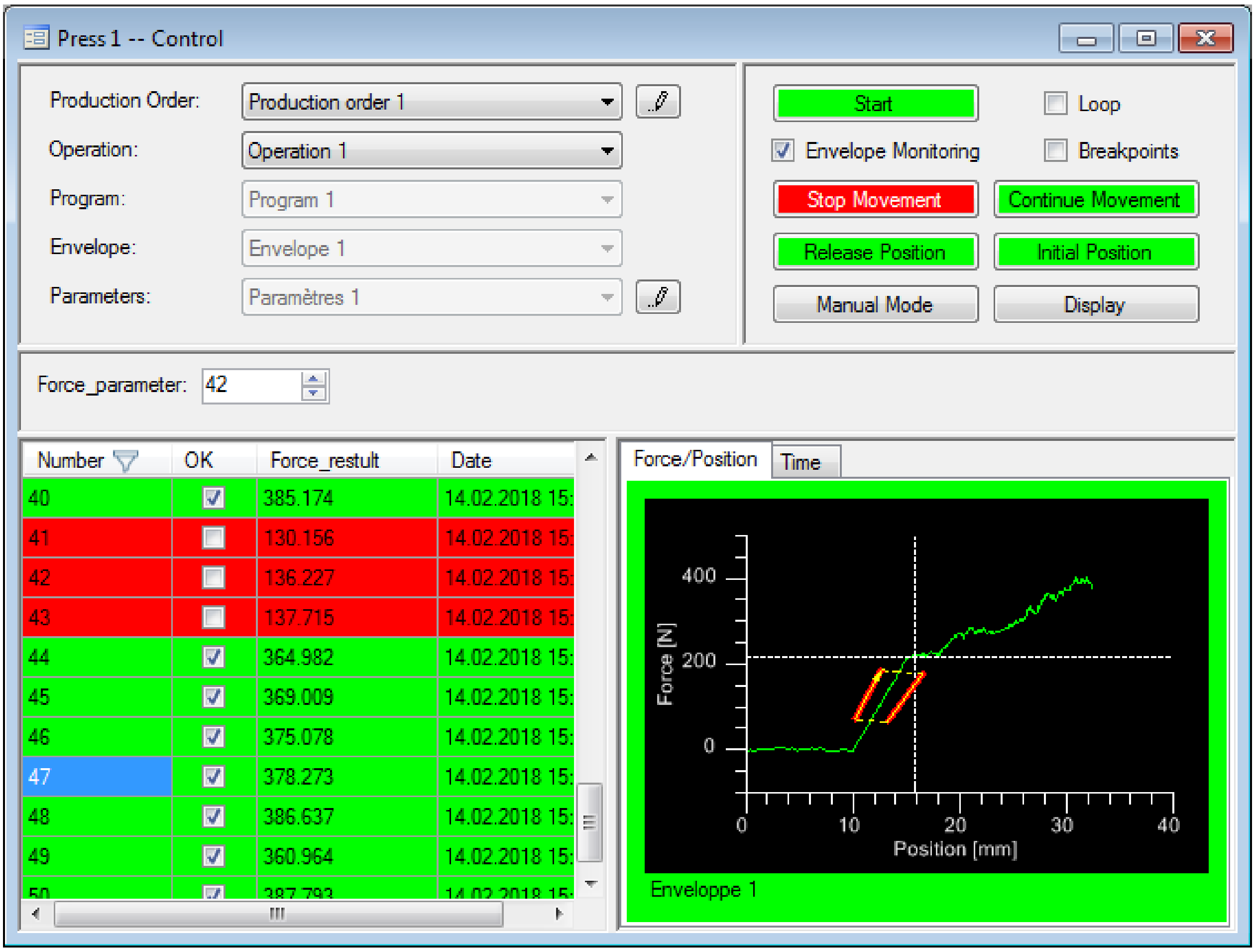


Figure 6: Control window

At the top right of the "control" window are all the buttons necessary to control the automatic cycle but also access to the manual mode.

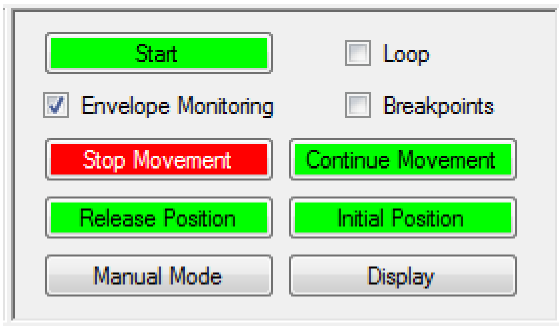


Figure 7: Automatic controls

The buttons on the control page (figure 7) have the following functions:

- The "start" button is used to start the selected program.
- The "stop movement" button stops the axis movement and pauses the execution of the part-program.
- The "continue movement" button restarts the axis movement and the execution of the part-program.

- The "release position" button is used to stop the movement of the axis and then make a movement towards the release position defined in the parameters.
- The "initial position" button is used to stop the movement of the axis and then make a movement towards the initial position defined in the parameters.
- If the "loop" box is checked, the selected program will run in loop.
- The "envelope monitoring" box is used to activate/deactivate the programmed reaction if the envelope tolerances are exceeded.
- The "breakpoints" box is used to carry out the programmed breakpoints. To continue after a breakpoint, you must click on the "start" button (named "continue" when the program is stopped at a breakpoint).

The "display" button accesses a window in which the axis position, force value, measuring sensor value and microswitch status are displayed in real time.

Creating a production order and a parameter set

The creation of a production order and a set of parameters is done from the control page. (see figure 8)

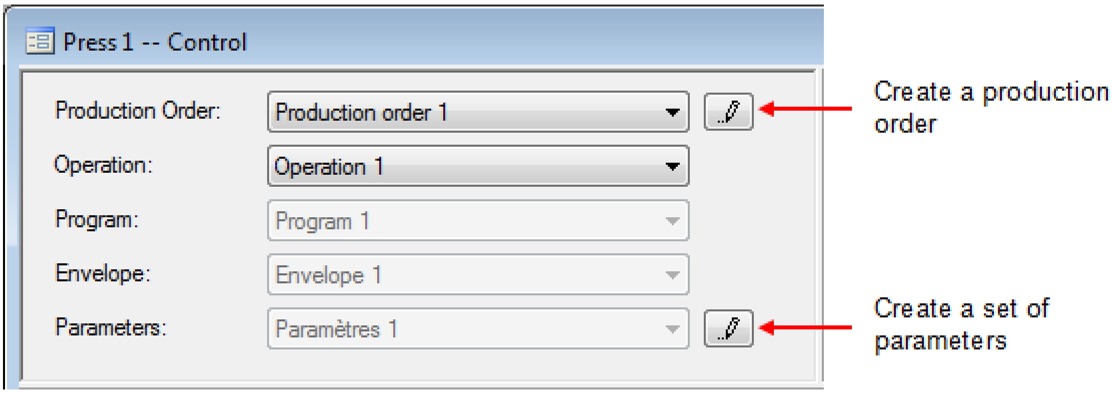


Figure 8: Creating a production order and parameter set

To create a production order, you must click on the button"…" next to the line "production order" (figure 8). The window below opens (figure 9), in which you must give a name to the production order and then add one or more operations.

These operations are necessarily composed of a program with its set of parameters and if necessary you can add an envelope.

As long as you have not performed a cycle with an operation, it is possible to modify its configuration, after having performed a cycle, you can no longer modify it.

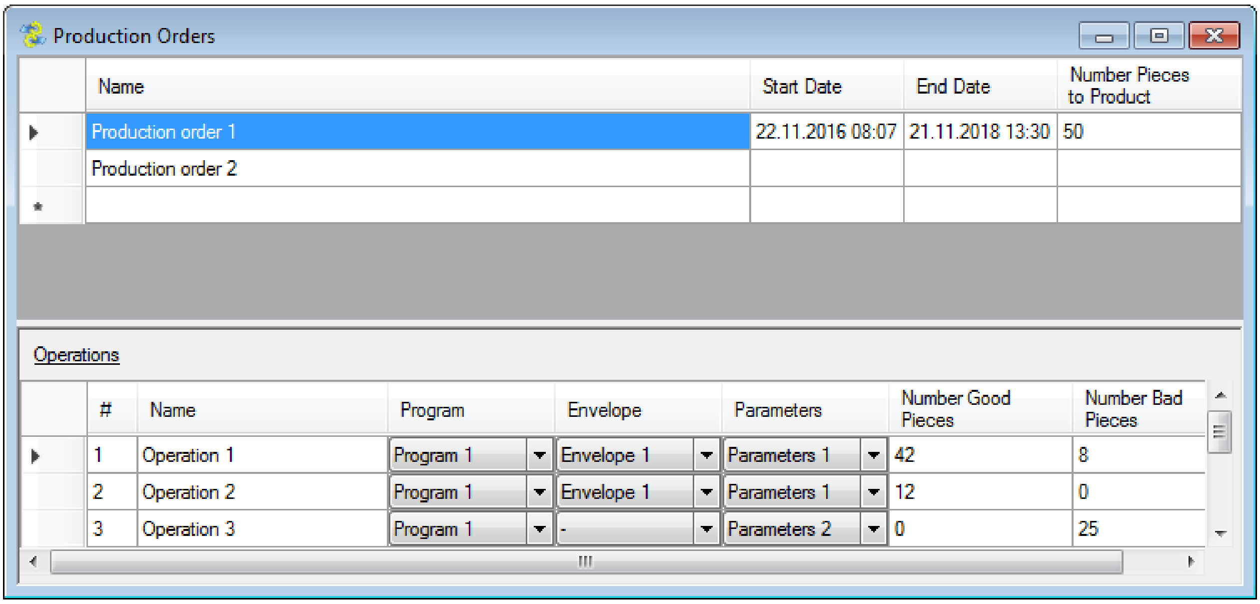


Figure 9: Creating a production order

Parameter sets allow you to save several parameter values for a single program and thus make it easier to change the values. To create them, click on the "..." button next to the "parameters" line. (figure 8)

The parameters that make up the set are located in the center of the "control" window, so you can enter the values of the parameters of each set by selecting it.

Once you have started a cycle, the selected parameter set will remain the same for the selected operation, to perform a cycle with a new parameter set, you must create a new operation.

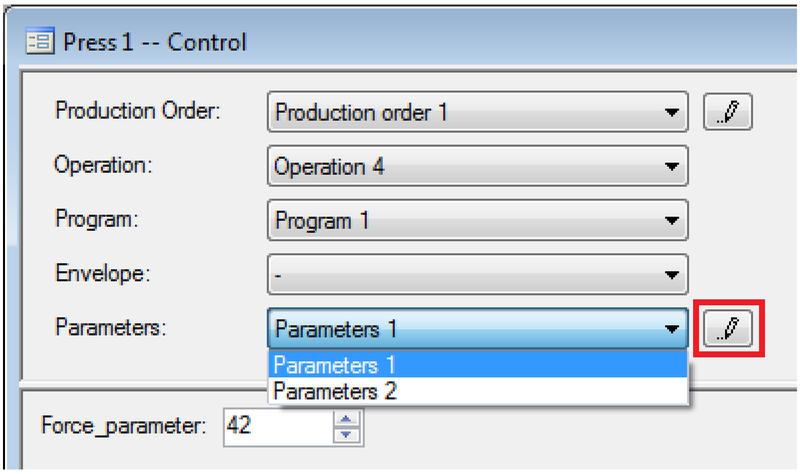


Figure 10: Creating parameter sets

Method of using the control page

Once you have created the production order, on the order page, choose the production order and the operation to be carried out.

Before starting the cycle, you must check the boxes if you want to work with or without the envelope and with or without the breakpoints. You can then start the cycle using the "Start" button.

When the program is finished, the results are automatically retrieved. (Curve, tolerance feedback, results variables and part ok/not ok) It is possible to automatically display the last result when it is received, to do so, right-click on the results table and choose "Automatically select new results".

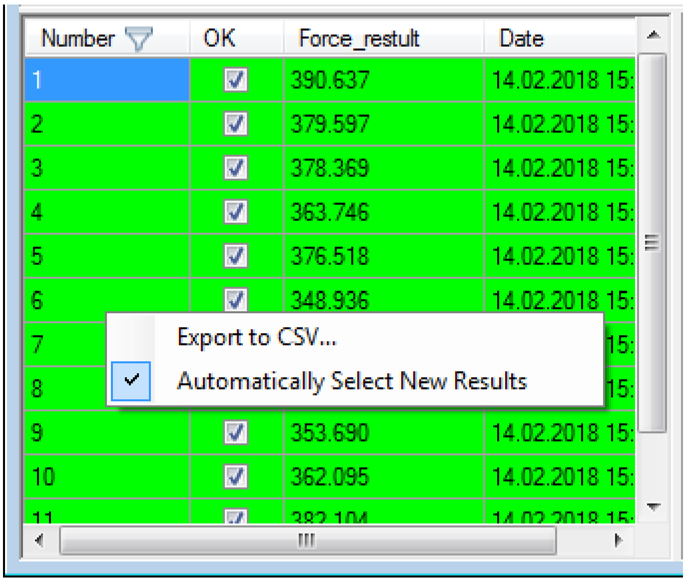


Figure 11: Automatic display of the last result

Curves

At the bottom right of the "control" window, you can view the force curve in relation to the position but also the force in time and the position in time. (see figures 12 and 13)

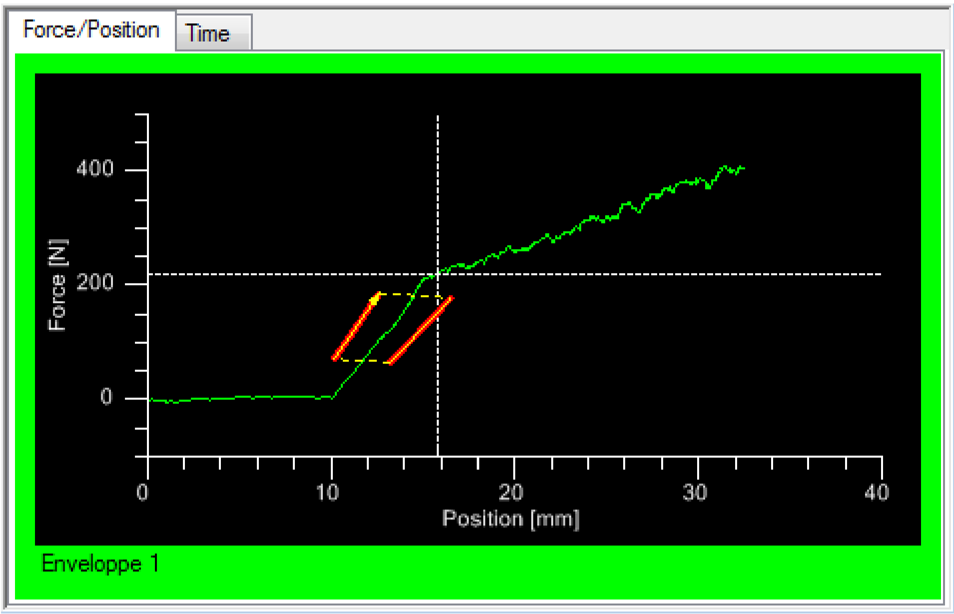


Figure 12: Force/position curve

In figure 13, the force appears in purple and the position in green. You can view the curves on separate graphs or on the same graph by clicking on the "overlapped graphs" button.

When the curves are overlapped on the same graph, you must place the cursor on one of the two for the force scale or position scale to be displayed.

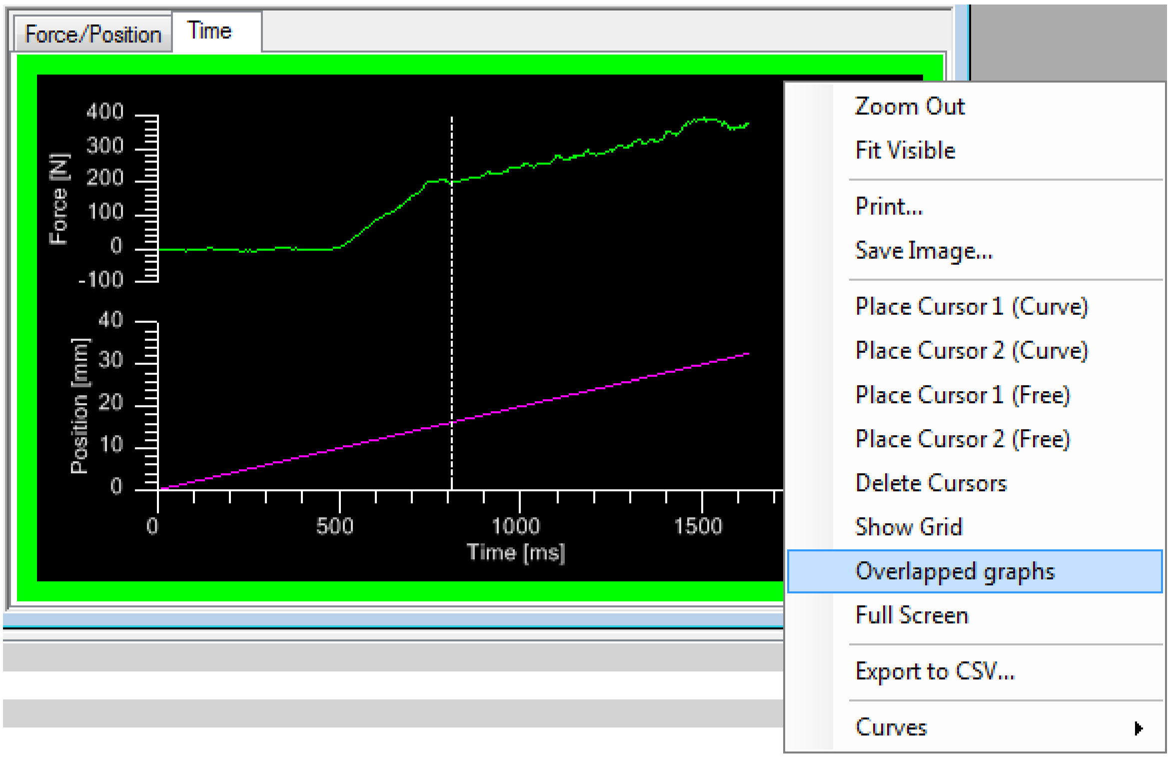


Figure 13: Force/time and position/time curves

Data base

If you want to delete a production order and its results, click on "file" in the menu bar and then "database". In this window you can delete production orders that you no longer need.

From this window (figure 14), you can export the database in ".sql" format to be used on another workstation for example. To do this, click on the "export" button to save the database in a".sql" file format and then transfer the file to the new workstation. Afterwards, from MecaMotion, open the ".sql" file using the "import" button, all the production orders of the project will then be added in the "results management" field.

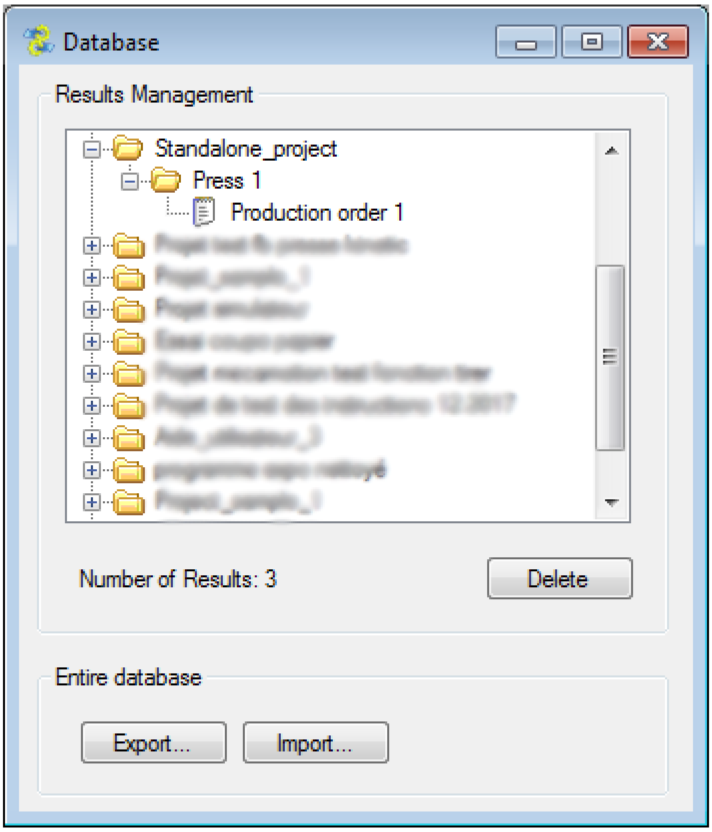


Figure 14: Database

Press control in manual mode

The press is controlled in manual mode from the control page by clicking on the "manual mode" button.

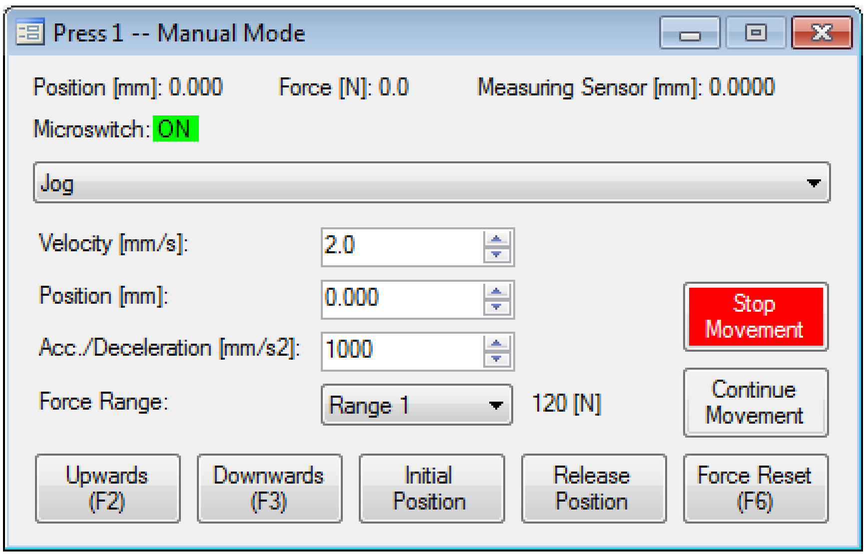


Figure 15 : Manual mode

This manual mode allows you to move the axis in run on sight and perform movements by giving relative or absolute position setpoints.

If you perform manual positioning, you must choose the speed, position and ramp (acceleration/deceleration). When a positioning is in progress, you can stop the movement at any time and then restart using the "stop movement" and "continue movement" buttons.

At the top of the window are displayed, the axis position, force value, measuring sensor value and micro switch status in real time.

From this window, it is also possible to move the axis to the initial or release position, select the range of the force sensor and reset the force sensor.

Errors

When an error occurs, the "Warning" triangle at the bottom right of the software will blink. If you click on it, the list of errors present is displayed, you can acknowledge them using the "ACK" button. You will find the list of errors in the PROFINET error list.

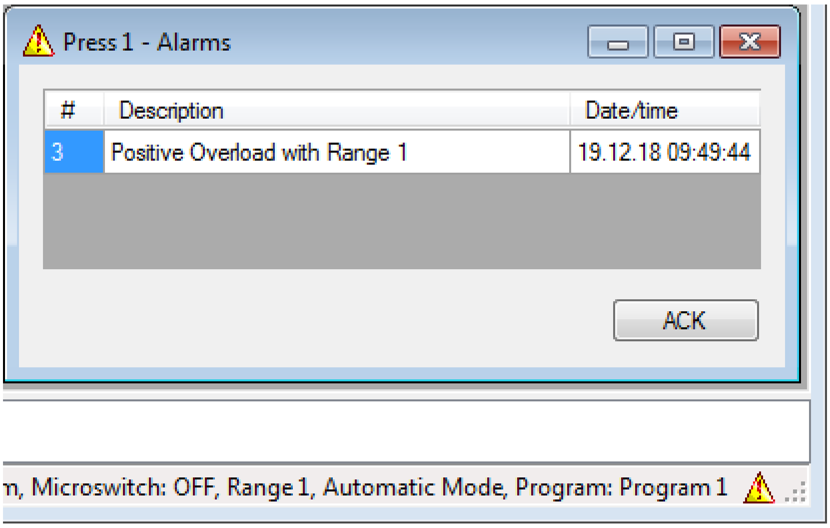


Figure 16 : Errors

Production mode

In the production mode you will find all the functions of the "control" window with a simplified layout to make it easier for the operator to use.

You can protect access to the edit mode and production mode with passwords.

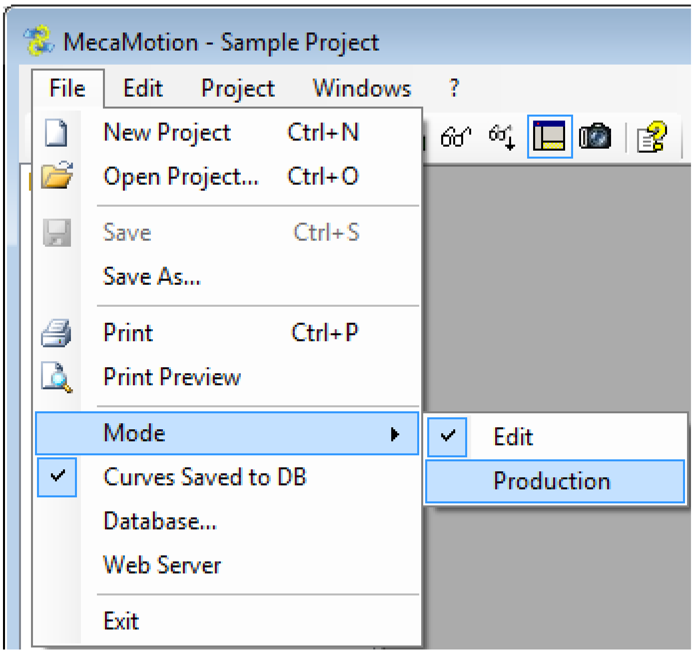


Figure 17: Access to production mode

If you want to return to edit mode, click on the production tab at the top left and then on the parameters tab at the right. On this window is the "project editor" button to return to edit mode. (Figure 18)

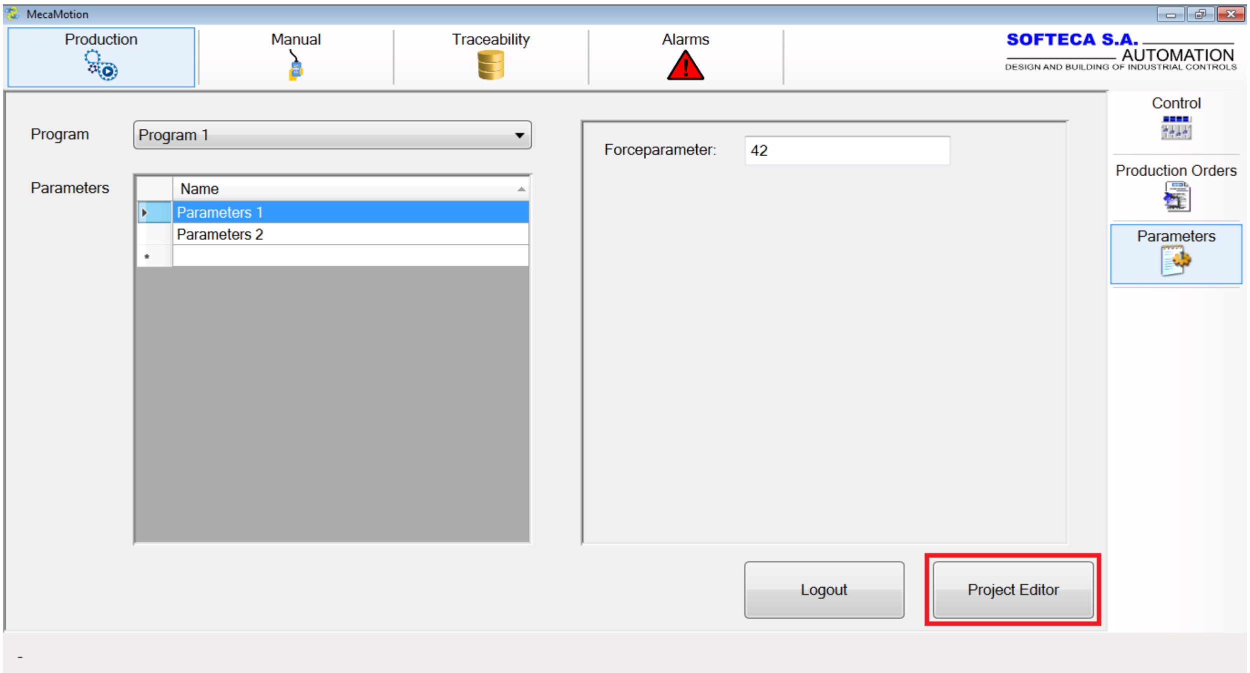


Figure 18: Return to editing mode