IST-MPLF-KIT-11

Version 1.0 | Last update: 21/12/2021

DESCRIPTION

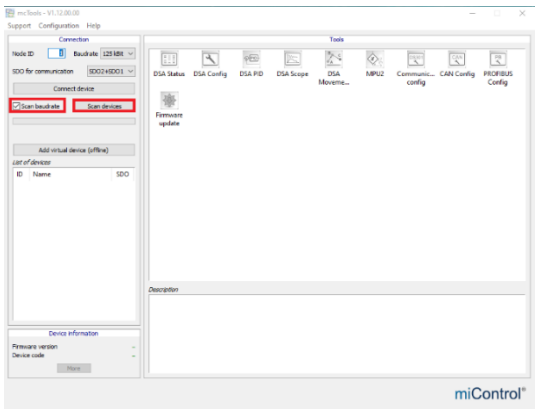
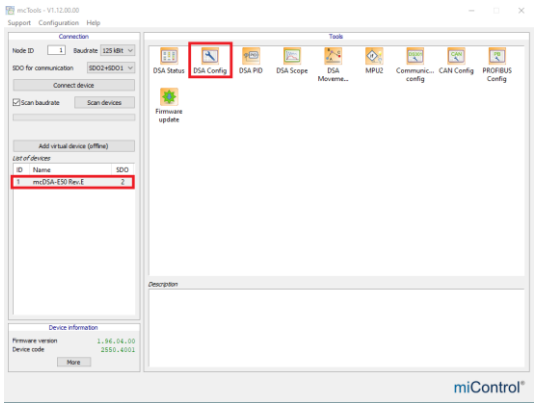
MPLF-KIT-11 is one of the evaluation kits for MPLF servo grippers.
It consists of a miControl mcDSA-E50 servo drive with its cables and connectors (Gimatic ordering code: DRV48MICTIN43IP150) and the M12 – 12 pins cable to connect the MPLF gripper to the driver (Gimatic ordering code: CFGM12901225P).

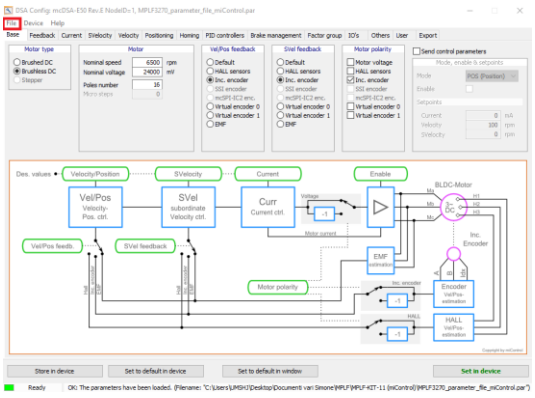
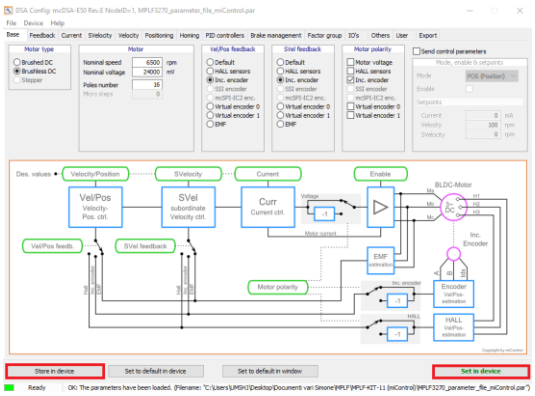
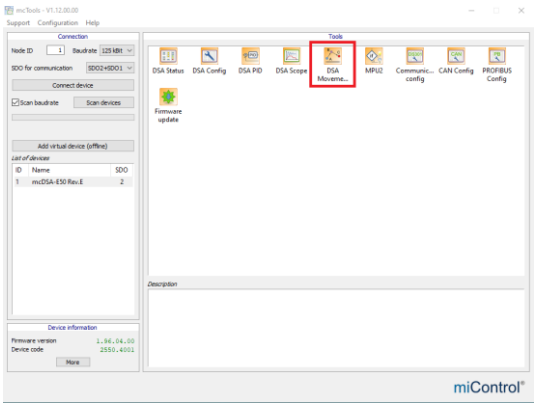
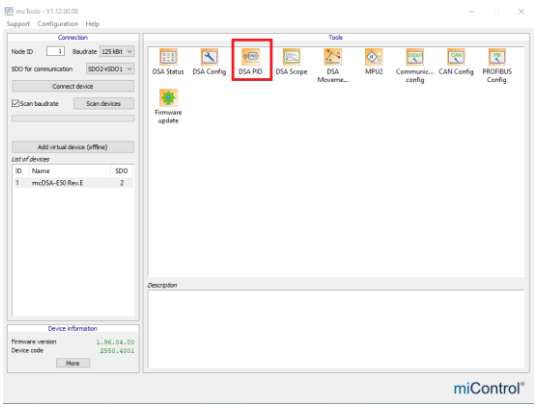
The M12 cable is provided already wired to two female connectors that must be inserted into the X5 and X6 male connectors of the drive and they are used for comunication purpose from drive to gripper and viceversa (power supply, Hall sensors, encoder signals and motor phases). Please refer to IST-MPLF for further details and gripper pinout. The X1 connector is the one reserved for the external power supply and its connection is up to the user. Connectors X2 and X4 are available for inputs and outputs signals.

MAIN FEATURES

	DRV48MICTIN43IP150-1 (miControl mcDSA-E50 servo drive)
Power supply voltage Up	24 V
Electronic supply voltage Ue	24 V
Max. output current	25 A
Continuous output current @Up = 24 V	7.5 A
Number of digital inputs	8 (Low voltage = 0 ÷ 5 V High voltage = 8 ÷ 30V)
Number of digital outputs	4 (Continuous output current = 0.3 A)
Number of analog inputs	3
Environmental degree	IP20
Dimensions	78 X 74 X 29 mm
Mass	95 g
CAN bus protocol	DS301
CAN bus device profile	DS402
CAN bus max baudrate	1 Mbit/s
Feedback type	ABZ incremental encoder

DRIVE CONFIGURATION WITH mcTools

1	Power ON the drive according to the following connection schema of the connector X1: Pins 3 and 5: GND = 0 Vdc Pin 2: +Up = 24 Vdc Pin 4: +Ue = 24 Vdc Pin 1: Fe = grounding.	
2	Download from miControl website the “mPLC Setup” configuration software along with all the technical documentation available and install it on your PC/laptop. It is recommended also to download the relevant data sheets and documentation. Please pay attention that a miControl account must be created to gain access to the download.	**https://www.micontrol.de/en** look for Drive controllers (article: mcDSA-E50, article number: 1513859)
3	Download from Gimatic website the “FILE MPLF-KIT-11” folder.	**www.gimatic.com** look for MPLF-KIT-11
4	Connect the drive to the PC using the specific USB/CAN converter (Gimatic ordering code: CONV-U2C-MICONTROL*). Launch the “mcTools” software (previously installed) and click on “Scan devices” to search the drive and establish a connection (please be sure to checking the “Scan baudrate” option). *(*) Sold separately*	
5	Once the device is found and associated, one can access the various tools. Double click on "DSA Config".	

6	Click on “File” and then “Load” in the top-left angle of the window and proceed selecting the appropriate configuration file by the size of the MPLF that one has available: - “*MPLF1630_parameter_file_miControl.par*” for MPLF1630 - “*MPLF2550_parameter_file_miControl.par*” for MPLF2550 - “*MPLF3270_parameter_ file_miControl.par*” for MPLF3270.	
7	Click to “Store in device” and then “Set in device” to store the parameters into the drive. Now, the gripper can be correctly commanded.	
8	Return in the main window and click on “DSA Movement”. With this tool is possible to command the gripper with one of the motion control modes available (position, velocity, current). Look at the miControl documentation for details.	
9	With the parameters file stored in the drive, a preliminary PID regulation is embedded. Please be sure that it is suitable for the specific application. In case of modifies needed, click on “DSA PID” tool to make the right tuning.	

DOWNLOAD A PROGRAM INTO THE DRIVE – GIMATIC DEMO

Through the “MPU2” tool is possible to download and store a program into the drive. Using the various inputs and outputs available the user can interact with the drive and create automatic processes.

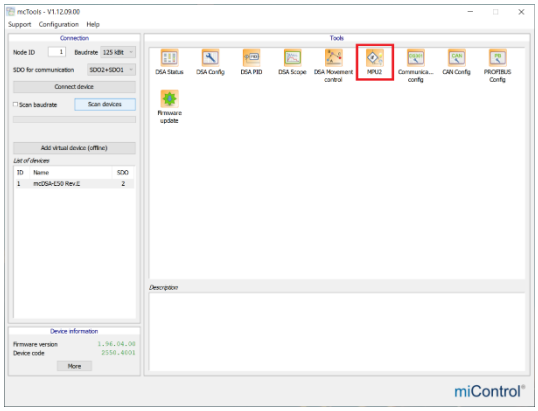
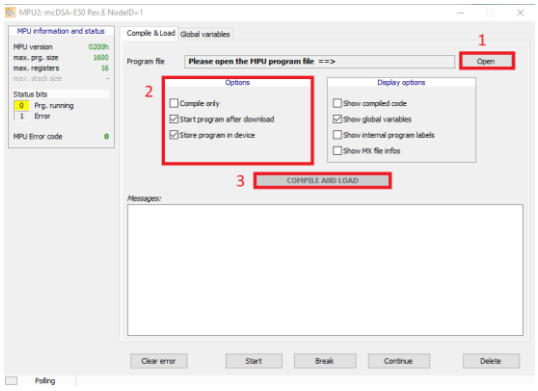
MPU stands for "Motion Process Unit" and it is based on the "Python" programming language. Please refer to the miControl documentation for all the details regarding this functionality.

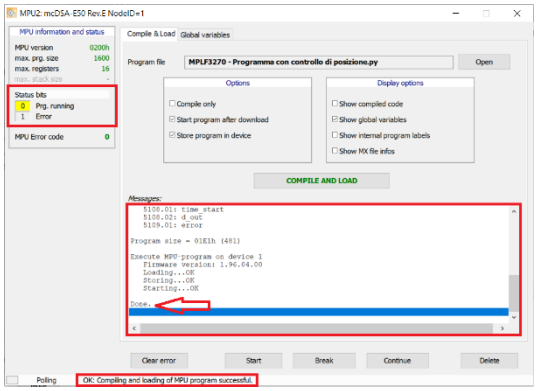
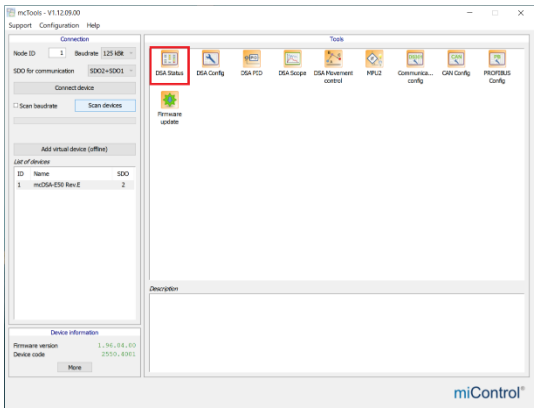
The program must be written externally and later stored in the drive memory. Doing so, the program will be available every time the device is powered without the need to be connect the drive to the PC thus making possible the usage of the MPLF system in the user’s specific application.

Just as an example, Gimatic provides a demo program that, by using three digital inputs and three digital outputs, allows the user to command opening closing movements between two predefined positions.

In the folder available on the Gimatic website, three different files are present (.py extension), one for each size of MPLF. The logic of the program is the same (explained in the next section of this document), only the configuration parameter set is different depending on gripper size.

The following table shows the steps necessary to store program and parameters into the drive.

1	Click on “MPU2” tool from the main window.	
2	Using the appropriate buttons, select the desired program (for example "*MPLF3270 – Gimatic demo.py*") and click on “COMPILE AND LOAD” (please be sure to select the “Start program after download” and “Store program in device” options).	

3	In case of successful operation, the word “Done.” Appears at the bottom of the screen. Now the program is correctly stored into the drive.	
4	At this point the user can operate accordingly to the selected program by disconnecting the converter (the program is indeed stored into the memory of the drive, no communication between PC and drive is needed). In the initial phase of testing, it is still recommended to leave the converter connected to monitor the status of the gripper through the "DSA Status" tool.	

GIMATIC DEMO – PINOUT

INPUT	
Digital Input 0	Open
Digital Input 1	Close
Digital Input 2	Homing
OUTPUT	
Digital Output 0	Homing completed
Digital Output 1	Opening completed
Digital Output 2	Pre-positioning completed

GIMATIC DEMO – EXPLANATION

The demo program consists of an opening and closing motion sequence of the gripper between two predefined positions set as variables in the Python program.

The three programs share the same logic but with some differences depending on the gripper size (like configuration parameters and position values used for the movements). For this example, the "*MPLF3270 – Gimatic demo*" will be shown.

After the power up and initialization of the gripper, the **first operation to be done** (the only one allowed) is the **Homing** procedure that can be started by activating digital input 2 (i.e. set to +24 Vdc the digital input). The motion is triggered with the rising edge of this signal.
The jaws will move toward a **mechanical limit** (jaws completely open) and to this position will be assigned the reference position of null stroke (equivalent to 0 encoder count).

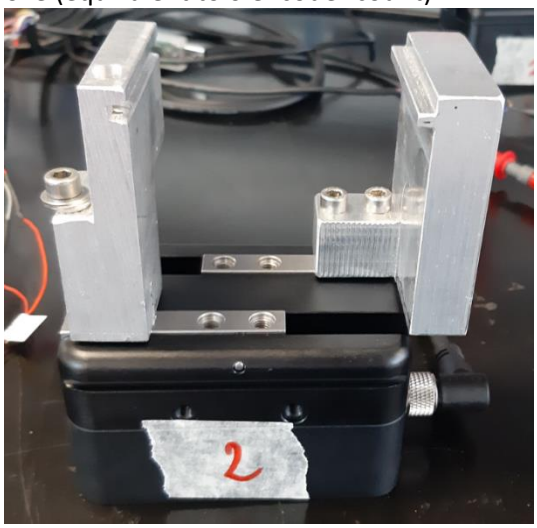


Figure 1 - Home position: jaws in mechanical limit

The digital output 0 will then be automatically activated to indicate the correct execution of the procedure.
At this point, it will be possible to control the closing (through digital input 1) and the opening (through digital input 0) of the jaws. Every movement will be triggered by the rising edge of the corresponding digital signal. In case of rising edge on both digital inputs, the last command received by the drive will be processed.

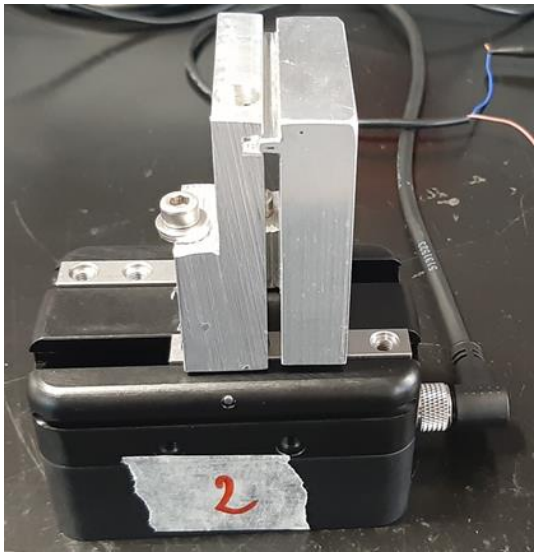


Figure 2 - Clamp/Close: the jaws move to take the piece, alternatively they close completely (current control)

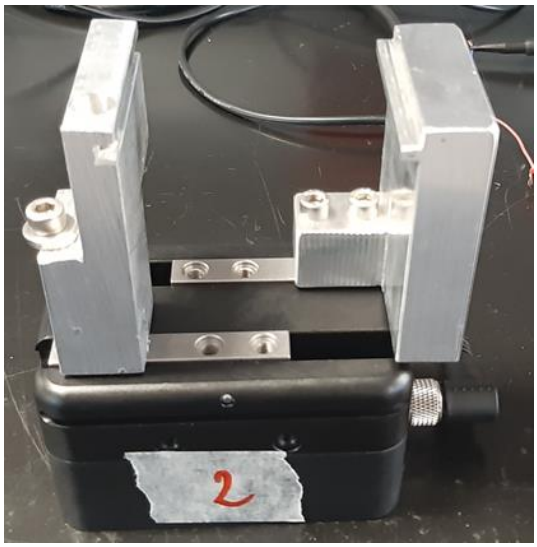


Figure 3 - Open: jaws in open state (not completely in mechanical limit, position control)

The **opened gripper** configuration is reached by controlling the gripper with a **position control** loop (position parametrized by the variable "**TARGET_POS_OPEN**", 50 counts as default value).

The **closing (or gripping) task** is performed by initially executing a positioning motion to a predefined position and then switching to current control mode to adapt to target dimensional tolerances and hold the target with a specified force (proportional to the current setpoint).
The full sequence is parametrized in the program using 3 variables:

- "**TARGET_POS_CLOSE**": position where the part is expected to be taken (950 counts as default value)
- "**POS_CLOSE_OFFFSET**": offset from the "TARGET_POS_CLOSE" position used to calculate pre-positioning (400 counts as default value)
- "**TARGET_CURR_CLAMP**": current setpoint proportional to the gripping force (1100 mA as default value). **Be careful not to exceed the nominal current of 1500 mA**).

If the piece is present, the gripper holds it with current "TARGET_CURR_CLAMP", if it is not present, the jaws close completely.

WARNING: The dimensions of the target must allow for the gripping to happen in current control mode to avoid fatal damage to the motor. Please check dimensions and values of the parameters accordingly.

Please also note that the gripper remains enabled if there is at least one digital input signal active, otherwise the gripper is disabled (losing any eventual gripped target).

Example: if the user commands a closing operation, the gripper performs the pre-positioning (in position control mode) and then the gripping (in current control mode). **If the digital input 1 is active, the gripper holds the target** (if present) with current equal to "TARGET_CURR_CLAMP", when the digital input 1 is released, the gripper is disabled, and the target is no-longer hold.

The picture shows a snapshot of the demo program source code with the most relevant parameters to set accordingly to the specific target dimensions and desired gripping force.

```
62  # On following it is possible to change the positions, velocity and current targets ------------------------------------------------------
63  TARGET_POS_OPEN             = 50     # [inc] Open position setpoint
64  TARGET_POS_CLOSE            = 950    # [inc] Position where it is expected to clamp the piece
65  POS_CLOSE_OFFSET            = 400    # [inc] Absolute offset value relative to the closing position (TAREGT_POS_CLOSE), used for pre-positioning
66  TARGET_POS_OFFSET           = TARGET_POS_CLOSE - POS_CLOSE_OFFSET
67  TARGET_VEL                  = 1000   # [rpm] Velocity setpoint  (maximum acceptable value: 5000 rpm)
68  TARGET_CURR_POS             = -300   # [mA] Current setpoint used for the homing (maximum acceptable value: 1500 mA)
69  TARGET_CURR_CLAMP           = 1100   # [mA] Current setpoint used for the clamping (maximum acceptable value: 1500 mA)
70
71  WAIT_TIME                   = 1500   # [ms] Time to allow jaws to move into Home position
```