



# INTEGRATED DRIVE ELECTRIC GRIPPERS

## MPCF SERIES



USER GUIDE

Version 2.2

Last update date: 07/04/2025



# TABLE OF CONTENTS

1. SAFETY - General product safety information
  - 1.1. General safety
2. INTRODUCTION - Introduction to this user guide
  - 2.1. About this user guide
  - 2.2. Terms and abbreviations
3. PRODUCT OVERVIEW - About Gimatic's product
  - 3.1. About Gimatic's MPCF grippers series
  - 3.2. Features
  - 3.3. MPCF system wiring
    - 3.3.1. MPCF system structure with CANbus
    - 3.3.2. MPCF system structure with RS485
  - 3.4. MPCF product range
  - 3.5. Catalogue number interpretation
4. INTERNAL SYSTEM DESIGN - Structure of a system based on MPCF grippers
  - 4.1. Introduction
  - 4.2. LEDs interpretation using middleware
  - 4.3. Communicating with the gripper using middleware
  - 4.4. Operating modes using Gimatic middleware
    - 4.4.1. Essential parameters
    - 4.4.2. List of operating modes
    - 4.4.3. Positioning operation (mode group 100)
    - 4.4.4. Operation with force control (mode group 200)
    - 4.4.5. Operation with force control and pre-positioning (mode group 300)
    - 4.4.6. Free-positioning operation (mode group 800)
    - 4.4.7. JOG operation (modes 901/902)
    - 4.4.8. Home operation (mode 900)
    - 4.4.9. Teaching and accessing registers operation (mode 000)
  - 4.5. Gripper parametrization
    - 4.5.1. General use
    - 4.5.2. For expert users only: modifying control loops parameters
  - 4.6. Operating without Gimatic middleware
5. SYSTEM COMPONENTS - Overview of system with MPCF grippers
  - 5.1. List of components
  - 5.2. Power supply
  - 5.3. USB/CAN converter
  - 5.4. USB/Serial converter
  - 5.5. Sensors
6. SPECIFICATIONS - Electrical and mechanical specifications
  - 6.1. Electrical

## 6.2. Mechanical

## 7. INSTALLATION - Mechanical and electrical configuration

### 7.1. Unpacking

### 7.2. Mechanical

#### 7.2.1. Safety loads

### 7.3. Electrical

#### 7.3.1. Electromagnetic Compatibility (EMC)

## 8. MAINTAINANCE - Suggested activities for systems based on Gimatic's MPCF grippers

### 8.1. General cautions

## 9. APPENDIX - List of accessories, illustration of MPCFApplication.exe and several examples of use

### 9.1. Accessories

### 9.2. MPCFApplication.exe

#### 9.2.1. Examples of interacting with MPCFApplication.exe

##### 9.2.1.1. Example 1 - Teaching new recipe parameters

##### 9.2.1.2. Example 2 - Make a movement from Base to Limit position using recipe 006

##### 9.2.1.3. Example 3 - Make a movement from Limit to Base Position with force control and velocity remodulation using recipe 006

##### 9.2.1.4. Example 4 - Make a movement from Base to Limit Position with force control, pre-positioning and all recipe parameters execute on-fly

##### 9.2.1.5. Example 5 - Set new zero position

##### 9.2.1.6. Example 6 - Erase the middleware

### 9.3. MPCFSerialDemo.exe

### 9.4. Examples of interacting without MPCFApplication.exe

#### 9.4.1. Examples 7 - Communication with a generic CAN master: make a grip with mode 200 and mode 220

#### 9.4.2. Examples 8 - Communication with a generic CAN master: interacting directly with CAN objects bypassing middleware

# 1. SAFETY



**NOTE:** Please read this User Guide carefully before installing the product. This document contains important notes for your personal safety. It must be read and understood by all persons who work with the product during any phase of the product lifecycle of the product life cycle. Only those documents currently available on the website are valid.



**WARNING:** Never let personnel or objects stand within the operating range of the gripper. Never operate the gripper if the machine on which it is fitted does not comply with safety laws and standards of your country.



**WARNING:** This manual and the warnings attached to the MPCF grippers only highlight hazards that can be predicted by Gimatic. Be aware they do not cover all possible hazards.  
Gimatic shall not be responsible for any accidents caused by the misuse or abuse of the device by the operator.

It is recommended that non-magnetic material be used in the system wherever possible. If magnetic material is required, ensure it is at sufficient distance from the gripper so that it is unaffected.

## 1.1. General safety

The following points must always be understood and respected:

- Equipment operators must read the User Guide carefully and make sure of the correct procedure before operating the MPCF grippers.
- Memorize the locations of the power isolator switches so that you can activate them immediately at any time if required.
- If two or more people are working together, establish signals so that they can communicate to confirm safety before proceeding to another step.
- Be aware of the closest First Aid station.
- Always make sure there are no obstacles or people near the devices during installation and or operation. Be aware of your environment and your surroundings.
- Keep the area around MPCF grippers clean and tidy.
- Take precautions to ensure that your clothing, hair or personal effects (such as jewelry) cannot become entangled in the equipment.
- Never touch any exposed wiring, connections or fittings while the equipment is in operation.
- Visually check all switches on the operator panel before operating them.
- Do not apply any mechanical force to the MPCF grippers, which may cause malfunction or failure.

- Never attempt cleaning or inspection when the machine is operating.
- Clean or inspect the equipment only after isolating all power sources.
- Only suitably qualified personnel should install, operate, repair and/or replace this equipment.
- Ensure all external wiring is clearly labeled. This will assist you and your colleagues in identifying possible electrical safety hazards.
- Install cables according to local legislation and regulations as applicable.
- Ensure there are not moving parts of the actuator while in contact with electrical connections. Movement can induce a voltage that could cause an electrical shock.

## 2. INTRODUCTION

### 2.1. About this user guide

This user guide provides the required information for planning to install, installation and servicing of MPCF grippers. It has been written specifically to meet the needs of qualified engineers, tradespersons, technicians and operators.

### 2.2. Terms and abbreviations

GND	Ground
g	Gravity $m/s^2$
V / mV	Volt / millivolt
A / mA	Ampere / milliampere
$\Omega$	Ohms
N / mN	Newton / millinewton
H / mH	Henry / millihenry
AC / DC	Alternating Current / Direct Current
Hz	Hertz
m/mm	Meter / millimeter
s / ms	Second / millisecond
inc	Increments (encoder counts)
dec	Decimal numeral system (base-ten)
hex	Hexadecimal numeral system (base-16)
AICD	Automatic Implantable Cardioverter-Defibrillator
EMC	Electromagnetic Compatibility

## 3. PRODUCT OVERVIEW

### 3.1. About Gimatic's MPCF grippers

The Gimatic's MPCF grippers have the particularity of an integrated drive. In its first edition of 2022, this new family of electric grippers consists of model MPCF3270 (size 32, total stroke 70 mm). The gripper integrates a drive with dual CANopen communication interface and RS485 Half Duplex. The drive is the result of a joint Gimatic-Ingenia project (<https://ingeniamc.com>), and the basic structure is similar to that of the standard drive model Triton Ingenia with the difference that the hardware is prepared for the integration of the STO security function and for the management of an absolute position feedback SSI.

With the CANbus interface, the electric gripper can be operated in real time as a generic device adhering to the CANopen DS-402 standard. The basic firmware of the drive allows the configuration of the motor inside the gripper, the feedback signal, the control and movement parameters of the actuator. Alternately, the gripper can be used with a middleware added by Gimatic on top of the basic firmware and it's based on predefined macros that can easily be called by an external master. The middleware is therefore a structure and a method of operation alternative to the basic standard CANopen that aims to simplify the user interaction and final applications without the need of implementation of the full CANopen standard. The choice between the two alternatives depends on the specific application and needs of the user.

The factory setting provides for the use of the middleware and in order to simplify the handling of the gripper on the application side and to allow the end user a simple and direct evaluation of the product, it is necessary to build a command-and-control interface that allows to manage the gripper as a single system. In this regard, in the appendix the Windows interface specifically developed by Gimatic ("MPCFApplication.exe") will be explained. If one instead wants to use the gripper as a generic CANopen slave for high demanding application, one must first erase the middleware by reset the macros (please see the related example in the appendix). In this case, the gripper becomes a CANopen axis CiA-301, CiA-305 and CiA-402 compliant (more details can be found here: <https://drives.novantamotion.com/emcl/canopen-protocol>) and the specific ESD file can be downloaded from Gimatic website to proper configure the controller unit (i.e. PLC).

The RS485 interface simplifies the use of the gripper in applications such as collaborative robotics allowing the direct interface of the gripper to industrial robot controllers equipped with simple serial interface. Also in this case a Windows interface ("MPCFSerialDemo.exe") has been developed to demonstrate the use of this communication channel.

The command and control of the gripper in standard operating conditions can be done indifferently through CANbus or RS485 interface.

The following document defines the specifications for the software communication process between the drive and external control unit dedicated to the configuration and testing of the gripper. The specifications explain different operation modes of the gripper managed by the macros called by an external master (either the Windows application or the software interface of a collaborative robot).

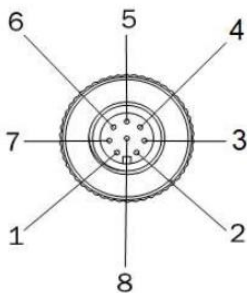
Please be sure to always use the latest versions of software, firmware, middleware, etc. (check the download area of the Gimatic website <https://www.gimatic.com/utility/> or contact your local area Gimatic's distributor).

## 3.2. Features

- Embedded drive.
- M12 – 8 pins male connector.
- Low installation and maintenance costs.
- Absolute linear encoder.
- Both serial and CANbus communication available.
- Windows applications simplify user configurations and tests (“MPCFApplication.exe” and “Gimatic MPCFSerialDemo.exe”).

## 3.3. MPCF system wiring

The MPCF gripper can operate indifferently with CANbus or serial protocol.  
MPCF grippers come with a M12 – 8 pins connector with the following pinout:



Nr. Pin	Function	Colour
1	Not used	White
2	+24V	Brown
3	CAN-H	Green
4	CAN-L	Yellow
5	RS485-	Grey
6	RS485+	Pink
7	GND (supply)	Blue
8	GND (signal)	Red

*Table 1 - MPCF pinout*

Please connect the power supply pins to an external power source and connect the pin 8 to the GND cable of the converter (either USB/CAN or USB/RS485) to guarantee a functional communication.



### 3.3.1. MPCF system structure with CANbus

A CANbus communication can be established by connecting pins 3 (green one) and 4 (yellow one) of the M12 gripper connector to a USB/CANbus converter.

Every CAN bus must be terminated with a 120  $\Omega$  resistor at each end of the bus. If not already available internally to the converter, please insert a 120  $\Omega$  resistance between CAN-H and CAN-L contacts to avoid disturbs on the communication line. If an error frames on the bus appears, please check the termination.

A full shielded cable is mandatory to establish the communication between gripper and master.

A twisted cable is recommended. As twisted pair is to be consider the CAN communication wires: CAN-H and CAN-L (and eventually the serial communication ones: RS485+ and RS485-).

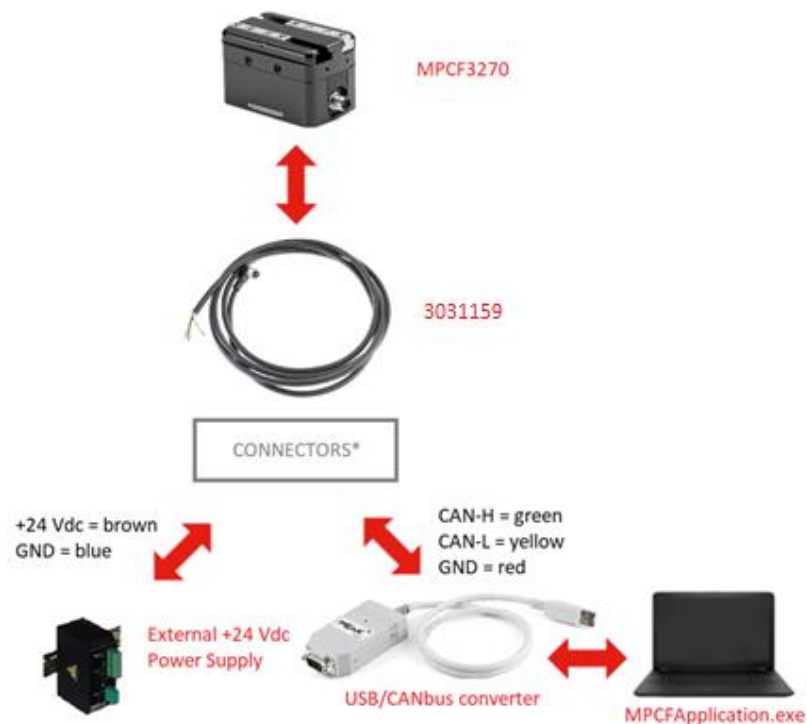


Figure 1 - System wiring with CANbus communication

\* The choice of appropriate connector is up to the user. Usually the USB/CANbus converter are provided with a D-SUB 9-pins connector. Please refer to the specific converter datasheet to correctly connect the pins. Power supply GND and signal GND (respectively pins 7 and 8 of the gripper) must be connected to ensure proper functioning.

Remark: please remember that a Peak converter is required to correctly operate with Gimatic MPCFApplication.exe.

Please refer to the “PCAN-USB – User Manual” for all the necessary information as, for example, the converter pinout:

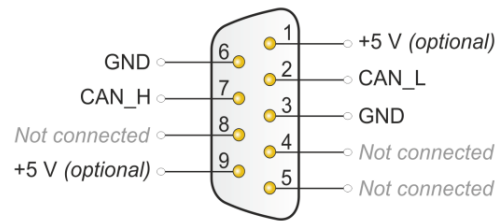


Figure 2 - View onto connector of the PCAN-USB adapter

Please remind to install the converter drivers to be able to correctly communicate with the gripper. The drivers can be found in the folder available at <https://www.gimatic.com/>.

Gimatic provides a KIT (ordering code: “MPCF-KIT-02”) composed by the M12 full shielded cable (already wired with a D-SUB connector) and the Peak USB/CANbus converter to have a ready-to-use connection. This kit includes the 120 Ohm resistor (embedded in the D-SUB connector).

Please refer to chapter 9.1 for the complete accessories list.

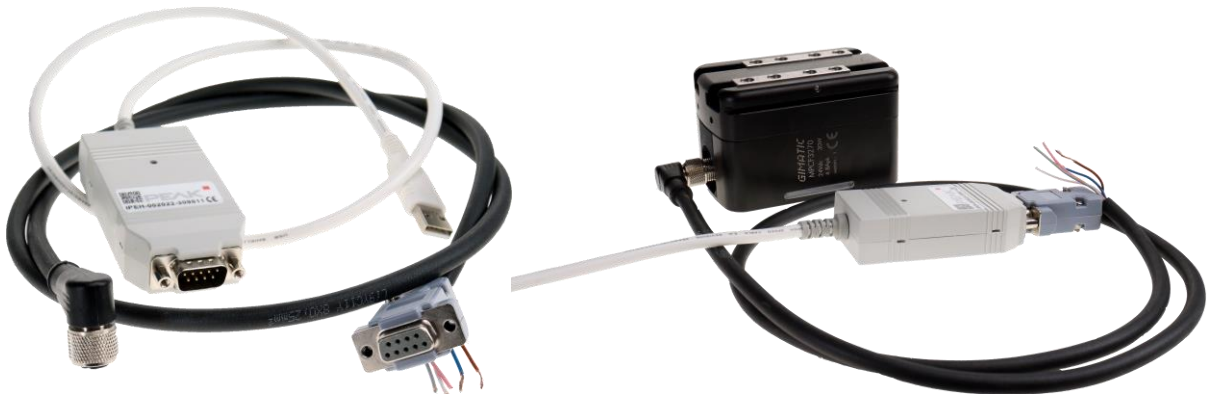


Figure 3 - MPCF-KIT-02

### 3.3.2. MPCF system structure with RS485

To configure the system using the serial protocol communication, please connect the pin number 5 (grey one) and 6 (pink one) of the M12 gripper connector to an USB/Serial converter.

A full shielded cable is mandatory to establish the communication between gripper and master.

A twisted cable is recommended. As twisted pair is to be consider the serial communication wires: RS485+ and RS485- (and eventually the CANbus communication ones: CAN-H and CAN-L).

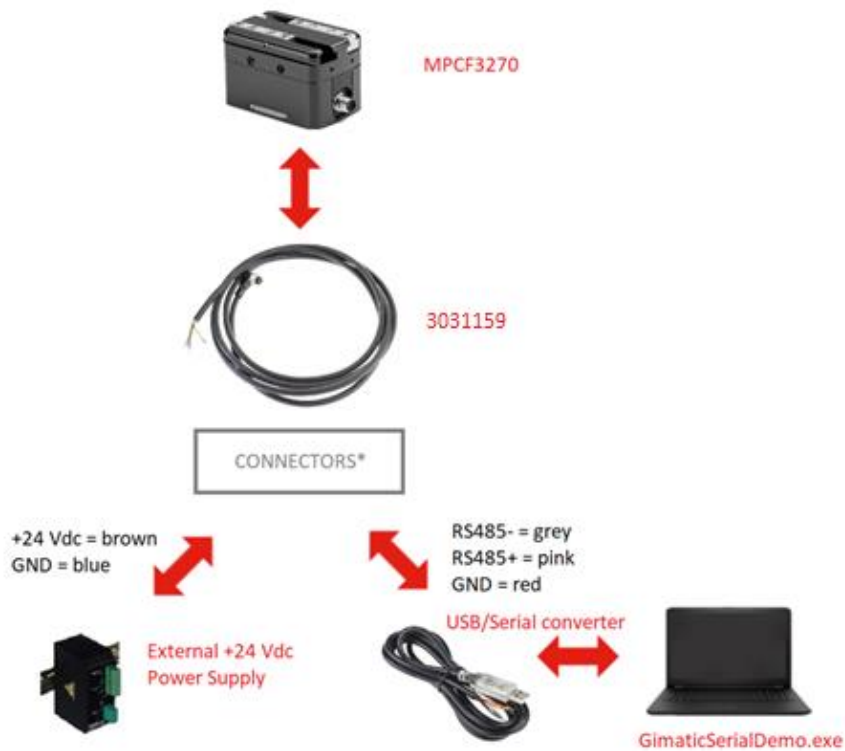


Figure 4 - System wiring with Serial communication

\* The choice of appropriate connectors is up to the user. Standard D-SUB 9-pins connectors type can be used for example to connect the RS485 lines to the USB/Serial converter. The connection choice is arbitrary, but it is mandatory to respect the one-to-one connection. In the following example pins number 2, 3 and 5 have been used.

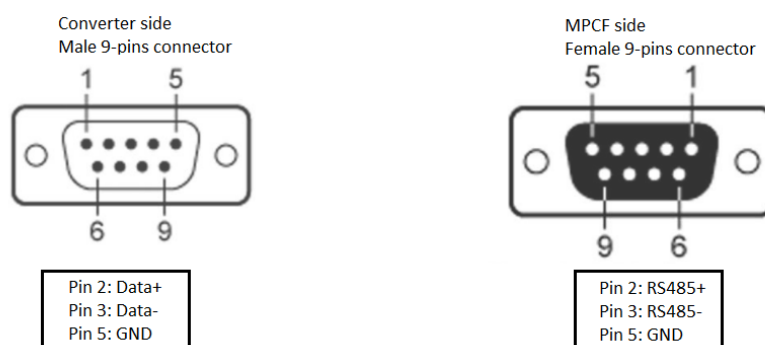


Figure 5 - RS485 arbitrary connection example

Power supply GND and signal GND (respectively pins 7 and 8 of the gripper) must be connected to ensure proper functioning.

For more details about configuring system for the serial communication, please refer to the “Gimatic MPCF gripper serial protocol.pdf” document included in the folder available at <https://www.gimatic.com/> and on to the Ingenia on-line documentation available at:

<https://drives.novantamotion.com/emcl/uart-based-rs-232-and-rs-485>.

### 3.4. MPCF product range


	
	<b>MPCF3270</b>
Stroke	70 mm (2x35 mm)
Continuous (maximum) gripping force	2x80 N (2x235 N)
Total mass	840 g
Rated current	1.58 A
Peak current	4.8 A
Supply	24 Vdc
Environmental degree	IP40
Temperature range	+5°C - 60°C
Feedback type	Absolute linear encoder on the jaws
Motor type	Brushless + Hall sensors
Max allowed length of connection cable (shielded) with CANbus	5 m
Max allowed length of connection cable (shielded) with RS485	5 m
Position resolution @ jaws	0.3 µm/pulse

Table 2 - MPCF main features

### 3.5. Catalogue number interpretation

Integrated drive electric gripper: ordering code explanation

# MPCFXXYY

**XX** → Gripper's size: 32 (large size)

**YY** → Total stroke: 70 [mm]

## 4. INTERNAL SYSTEM DESIGN

### 4.1. Introduction

The gripper can be used as generic slave device adhering to the CANopen DS-402 standard or through the use of the middleware firmware structure.

The main purpose of this document is to explain the over-structure managed by the macros (middleware) and to drive the user through its usage. Please remember that the middleware is active as standard on MPCF grippers.

An important feature of this grippers' family is that some parameters that define the movement can be stored in the non-volatile custom memory. Using of the middleware, it is possible to store into the MPCF drive a maximum eight recipes of parameters (like target position, velocity, torque, etc.) to quickly command the movements.

The execution of a movement can therefore take place in two ways: recalling one of the eight recipes previously stored or by performing the equivalent of a recipe by setting the relevant parameters in a volatile memory area. The advantage of this latter solution is to avoid having to perform multiple operations of reading and writing registers (necessary for storing recipes) and allows the customer to quickly perform a movement directly. The following also defines the possible operation modes for the gripper provided by middleware, describing the functionality, the parameters used for each movement and the value that one needs to set into main registers to control the gripper.

### 4.2. LEDs interpretation using middleware

In the bottom part of the gripper body, on both sides, there are a LEDs that can be useful to quickly interpretate the current gripper status.

The following table summarize all the possibilities.

LEDs colour	Interpretation
Green	Gripper powered on and enabled
Red	Gripper in error; a Reset is required to be able to command the gripper again
Yellow	The jaws are in the Limit Position
Azure/light blue	The jaws are in a position included in the Part Tolerance (i.e. between Teach1 and Teach2 Position)
Blue	The jaws are in a different position than the previous two

*Table 3 - Body's LEDs interpretation*

### 4.3. Communicating with the gripper using middleware

The following registers are defined to communicate with the gripper:

- **Gripper Control Register (register 0x2C00, sub-index 0x03):** register used to select the operation mode of the gripper, read and write the content of specific registers and start/stop movements. Access to this register by the external master is write-only. See the register structure in the Table 4.
- **Gripper Status Register (register 0x2C00, sub-index 0x24):** register containing the status information of the gripper. Access to this register by the external master is read-only. See the structure of the register in the Table 5.

	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Byte 3	EXMOD (Msb)	EXMOD	EXMOD	EXMOD	EXMOD	EXMOD	EXMOD	EXMOD
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Byte 2	EXMOD	EXMOD (Lsb)	-	SetPosition	Halt	-	-	-
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Byte 1	-	-	-	-	RCP_ID (Msb)	RCP_ID	RCP_ID	RCP_ID (Lsb)
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	REG_ID (Msb)	REG_ID (Lsb)	FSTP	TEACH	JOG	REER	STE	EN

*Table 4 - Gripper Control Register (0x2C00 - 0x03)*

**EN** (1 bit value, Bit0): enabling/disabling the gripper. Setting this bit involves enabling the engine that is maintained as long as this bit is not reset (unless errors result in disabling the engine, please see the Gripper Status Register).

**STE** (1 bit value, Bit1): start execution. A rise front of this bit involves the execution of the action indicated by the operation mode.

**RERR** (1 bit value, Bit2): error reset. When the gripper detects a rising edge of this bit, any errors present are reset.

**JOG** (1 bit value, Bit3): in group 900 mode of operation, the gripper performs the specified JOG movement of the mode of operation as long as this bit is set.

**TEACH** (1 bit value, Bit4): when the gripper detects a rising edge of this bit the value of Actual Position is transferred to the Value Register (teaching).

**FSTP** (1 bit value, Bit5): immediate cessation of any current movement. When the gripper detects a rise edge this bit, whatever movement is being executed is immediately stopped without deceleration ramps.

**REG\_ID** (2-bit value, Bit6...Bit7): currently unused.

**RCP\_ID** (4-bit value, Bit8...Bit11): index of the recipe to be executed (for the modalities that it previews it).

**Halt** (1 bit value, Bit19): immediate stopping of any motion with deceleration ramps.

**SetPosition** (1 bit value, Bit20): when it is high, an offset is calculated to be charged on power on to set the zero of the magnetic line; it works only once; to redo it one has to turn off and on again.

**EXMOD** (10-bit value, Bit22...Bit31): execution mode of the gripper, see execution mode.

	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Byte 3	RCP_ID (Msb)	RCP_ID	RCP_ID (Lsb)	VS4	VS3	VS2	VS1	DTCMP
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Byte 2	RCP_ACT	UD_POS	LM_POS	TH_POS	PR_POS	OF_POS	BS_POS	HM_POS
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Byte 1	-	-	-	-	-	-	-	-
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	RCP_RDY	MCMP	PositionSet	TEACH	JOG	ERR	-	EN

*Table 5 - Gripper Status Register (0x2C00 - 0x24)*

**EN** (1 bit value, Bit0): status bit, 1: gripper motor enabled, 0: gripper motor disabled.

**ERR** (1 bit value, Bit2): status bit, 1: errors occurred, 0: there are no errors.

**JOG** (1 bit value, Bit3): status bit, 1: gripper in JOG mode, 0: no running JOG movement.

**TEACH** (1 bit value, Bit4): status bit, 1: teaching completed, 0: no teaching detected (please note the TEACH bit in the Gripper Control Register).

**PositionSet** (1 bit value, Bit5): is activated after the offset to reset the position has been calculated and the only way to reset it is to turn it off and on again.

**MCMP** (1 bit value, Bit6): status bit, 1: motion execution completed, 0: movement in progress or no movement.

**RCP\_RDY** (1 bit value, Bit7): status bit, 1: recipe uploaded, 0: loading recipe.

**HM\_POS** (1 bit value, Bit16): status bit, 1: Home Position reached.

**BS\_POS** (1 bit value, Bit17): status bit, 1: Base Position reached.

**OF\_POS** (1 bit value, Bit18): status bit, 1: Offset Position reached.

**PR\_POS** (1 bit value, Bit19): status bit, 1: presence of a piece between the Offset Position and Limit Position.

**TH\_POS** (1 bit value, Bit20): status bit, 1: presence of a piece between the Teach 1 Position and Teach 2 Position.

**LM\_POS** (1 bit value, Bit21): status bit, 1: Limit Position reached.

**UD\_POS** (1 bit value, Bit22): status bit, 1: gripper stops in an undefined position.

**RCP\_ACT** (1 bit value, Bit23): status bit, 1: mode of operation from recipe active.

**VS1...VS4** (1 bit value, Bit25...Bit28): status of each of the Virtual Switches from 1 to 4 (of the active recipe or RAM register), 1: active, 0: not active.

**RCP\_ID** (3-bit value, Bit29...Bit31): index of the last recipe selected among the 8 possible.

## 4.4. Operating modes using Gimatic middleware

### 4.4.1. Essentials parameters

This section anticipates the description of the most articulated method of operation of the gripper to introduce the essential parameters that define the movements in a recipe. Figure 6 shows qualitatively the motion profile in a pre-positioning gripping mode. The velocity profile is represented in relation to the controlled position.

This control mode aims to grip a part with a nominal physical size within the range defined by the dimensions "Teach 1" and "Teach 2", starting with a velocity control movement (speed setpoint "Drive velocity") from the "Base Position" to the "Offset Position". When the "Offset Position" is reached, the control mode is switched from velocity control to current control with setpoint "Grip Force" until the gripping is completed (if the part is present) or until the "Limit Position" is reached (in case of missing part).

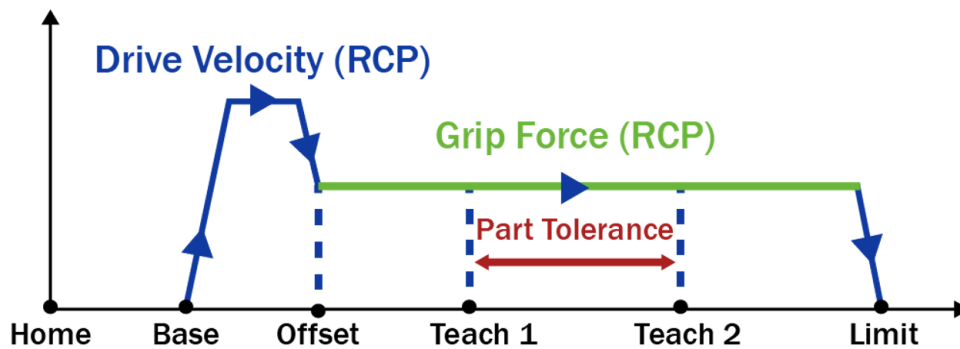


Figure 6 - Example of operating mode

With reference to Figure 6, the following parameters are defined:

- **Home Position:** zero position of the jaws assumed conventionally as reference.
- **Base Position:** starting position out of space compared to the target piece.
- **Offset position:** out of contact position but close to the part defined to perform a fast, non-contact approach movement.
- **Teach Position 1:** position corresponding to the minimum dimension of the part at which the contact with the object to be grasped could occur (dimensional tolerance). Must be smaller than Teach position 2.
- **Teach Position 2:** position corresponding to the maximum dimension of the part at which the contact with the object to be grasped could occur (dimensional tolerance). Must be greater than Teach Position 1.
- **Limit Position:** target position limit of the gripping fingers in case there is no part to be gripped or in case the gripper is simply controlled in position.
- **Drive Velocity:** velocity setpoint.
- **Grip Force:** gripping setpoint (with which the gripper grabs and holds the possible present piece).



## 4.4.2. List of operating modes

Several operating modes are available:

- Mode group 100: operating modes with position setpoint and velocity control. This group is used in case of need of simple point to point positioning, without gripping and therefore parameters like Offset Position, Grip Force, Teach1 and Teach2 are not relevant.
- Mode group 200: operating modes with position setpoint and both velocity and force control. This group is used when a gripping operation is required without switching modality at Offset Position.
- Mode group 300: operating modes with position setpoint and both velocity and force control. This group is used when a gripping operation is required with the pass through the Offset Position. The motion is therefore divided in two parts: first, a quickest movement to Offset Position with velocity control is executed, then a slowest movement with force control is performed for gripping the part (if it is present).
- Mode group 800: to execute the same functionality of previous mode groups but with on-fly parameters (values stored in RAM memory) and not from the recipes.
- Mode group 900: to execute a point to point positioning to Home position or to perform movement in JOG mode.
- Mode 000: teaching mode.

Every operating mode is identified by a number and a letter code.

**Mode name explanation, example**

### Mode 100, P2P\_SPEED\_B-L\_RCP

**Mode 100** → Identify number of the operating mode

**P2P\_SPEED** → Type of control loop:

P2P\_SPEED = Point to point movement with velocity control

P2P\_FORCE = Point to point movement with force control and speed profile

GPR\_PRP = Gripping command with force control and pre-positioning with speed profile

**B-L** → Direction of the movement:

B-L = Direction from Base Position to Limit Position; Limit Position is the position setpoint

L-B = Direction from Limit Position to Base Position; Base Position is the position setpoint

**RCP** → Parameters used for the law of movement:

RCP = All parameters taken from the relative recipe

REG\_V = All parameters taken from the relative recipe except for velocity (editable by the user in real time)

REG\_FV = All parameters taken from the relative recipe except for velocity and force (editable by the user in real time)

DIRECT = All parameters are editable directly by the user in real time

### 4.4.3. Positioning operation (mode group 100)

The gripper performs simple positioning with velocity control. If the gripper encounters an obstacle during the execution of the movement (reaching a fixed pre-set current limit in the drive), the movement stops, and an error is generated. Recovery is possible by resetting the error (flag in the Control Register) and commanding a new movement.

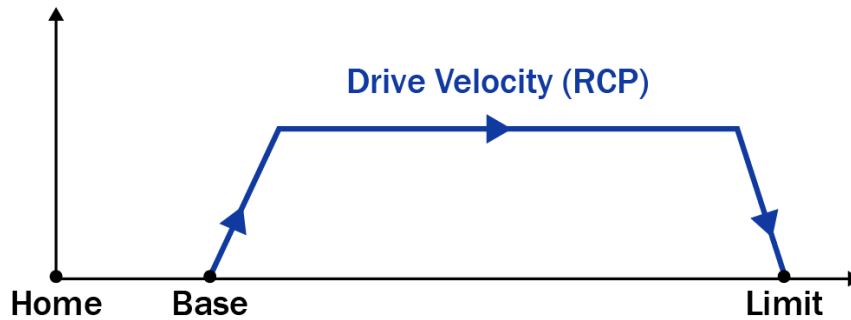


Figure 7 – Point to point positioning

**Mode 100, P2P\_SPEED\_B-L\_RCP:** Positioning with position control at Limit Position (recipe parameter) with Drive Velocity speed profile (recipe parameter).

In this mode the gripper performs a simple positioning of the jaws with control in position and with set velocity reference. The position and velocity setpoints correspond respectively to the Limit Position and Drive Velocity parameters of the specific recipe called with this command.

**Mode 101, P2P\_SPEED\_L-B\_RCP:** Positioning with position control at Base Position (recipe parameter) with Drive Velocity speed profile (recipe parameter).

In this mode the gripper performs a simple positioning of the jaws with control in position and with set velocity reference. The position and velocity setpoints correspond respectively to the Base Position and Drive Velocity parameters of the specific recipe called with this command.

**Mode 110, P2P\_SPEED\_B-L\_REG\_V:** Positioning with position control at Limit Position (recipe parameter) with Drive Velocity speed profile (register **0x2C00-0x30**).

In this mode the gripper performs a simple positioning of the jaws with control in position and with set velocity reference. The position and velocity setpoints correspond respectively to the Limit Position of the specific recipe called with this command and to the Drive Velocity of the register **0x2C00-0x30** (velocity remodulation).

**Mode 111, P2P\_SPEED\_L-B\_REG\_V:** Positioning with position control at Base Position (recipe parameter) with Drive Velocity speed profile (register **0x2C00-0x30**).

In this mode the gripper performs a simple positioning of the jaws with control in position and with set velocity reference. The position and velocity setpoints correspond respectively to the Base Position of the specific recipe called with this command and to the Drive Velocity of the register **0x2C00-0x30** (velocity remodulation).

#### 4.4.4. Operation with force control (mode group 200)

In this mode the gripper performs a movement from the Base Position to the Limit Position or vice versa applying a gripping force defined by the Grip Force parameter for any piece is grabbed inside the stroke.

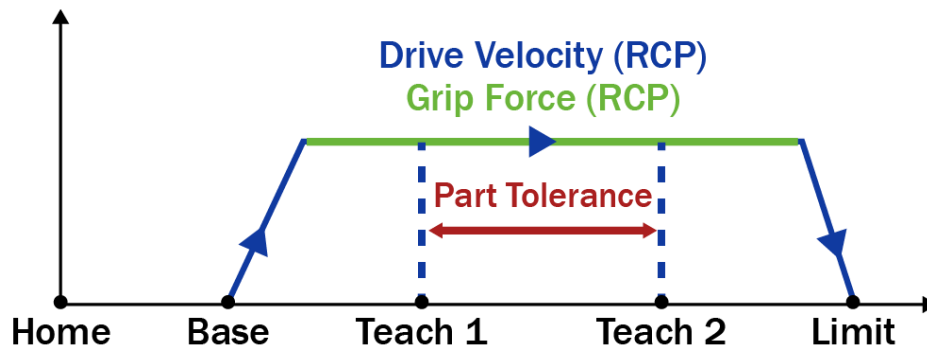


Figure 8 - Gripping with force control

**Mode 200, P2P\_FORCE\_B-L\_RCP:** Gripping command to Limit Position (recipe parameter) with setpoint of force set by the Grip Force parameter (recipe parameter) until reaching the position or gripping of the piece. In this mode, the gripper moves at a velocity set by the Drive Velocity value, interrupting and maintaining the grip force value set by Grip Force when contact is detected with any workpiece along the stroke.

**Mode 201, P2P\_FORCE\_L-B\_RCP:** Gripping command to Base Position (recipe parameter) with setpoint of force set by the Grip Force parameter (recipe parameter) until reaching the position or gripping of the piece. In this mode, the gripper moves at a velocity set by the Drive Velocity value, interrupting and maintaining the grip force value set by Grip Force when contact is detected with any workpiece along the stroke.

**Mode 210, P2P\_FORCE\_B-L\_REG\_V:** Gripping command to Limit Position (recipe parameter) with Grip Force setpoint (recipe parameter) and Drive Velocity as velocity profile (register **0x2C00-0x30**). Similar to mode 200 but with velocity remodulation.

**Mode 211, P2P\_FORCE\_L-B\_REG\_V:** Gripping command to Base Position (recipe parameter) with Grip Force setpoint (recipe parameter) and Drive Velocity as velocity profile (register **0x2C00-0x30**). Similar to mode 201 but with velocity remodulation.

**Mode 220, P2P\_FORCE\_B-L\_REG\_FV:** Gripping command to Limit Position (recipe parameter) with Grip Force setpoint (register **0x2C00-0x31**) and with Drive Velocity as velocity profile (register **0x2C00-0x30**). Similar to mode 210 but with the remodulation of the gripping force.

**Mode 221, P2P\_FORCE\_L-B\_REG\_FV:** Gripping command to Base Position (recipe parameter) with Grip Force setpoint (register **0x2C00-0x31**) and with Drive Velocity as velocity profile (register **0x2C00-0x30**). Similar to mode 211 but with the remodulation of the gripping force.

#### 4.4.5. Operation with force control and pre-positioning (mode group 300)

This method of operation extends the one with the force control (mode group 200) providing an additional initial pre-positioning movement to quickly approach the object to grip. Once the Offset Position is reached, the gripper automatically switches to the force control mode without stopping the motor. The movement away from the workpiece is controlled in position at the velocity set by the parameter Drive Velocity.

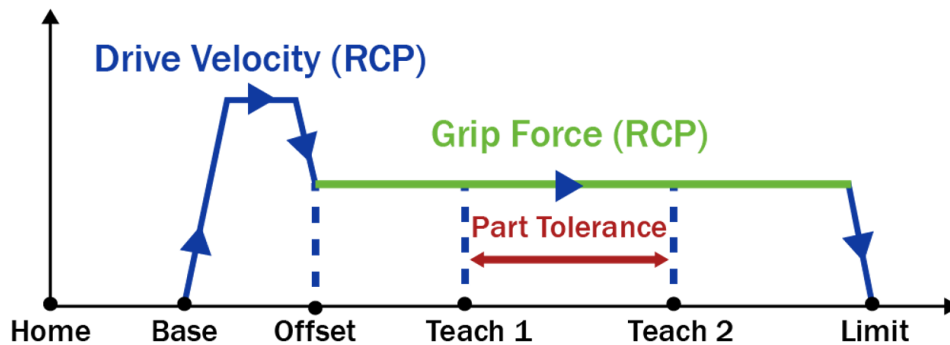


Figure 9 – Gripping with force control and pre-positioning

**Mode 300, GRP\_PRP\_B-L\_RCP:** Gripping command to Limit Position (recipe parameter) with pre-positioning approach to Offset Position (recipe parameter), setpoint of force set by the Grip Force parameter (recipe parameter) and setpoint of velocity set by the Drive Velocity parameter (recipe parameter) until reaching the position or gripping of the piece. The gripper keeps the value of gripping force set by Grip Force when the contact with any part along the stroke is detected.

**Mode 310, GRP\_PRP\_B-L\_REG\_V:** Gripping command to Limit Position (recipe parameter) with pre-positioning approach to Offset Position (recipe parameter), setpoint of force set by the Grip Force parameter (recipe parameter) and setpoint of velocity set by the Drive Velocity parameter (register **0x2C00- 0x30**). Similar to mode 300 but with velocity remodulation.

**Mode 320, GRP\_PRP\_B-L\_REG\_FV:** Gripping command to Limit Position (recipe parameter) with pre-positioning approach to Offset Position (recipe parameter), setpoint of force set by the Grip Force parameter (register **0x2C00-0x31**) and setpoint of velocity set by the Drive Velocity parameter (register **0x2C00- 0x30**). Similar to mode 310 but with force remodulation.

#### 4.4.6. Free-positioning operation (mode group 800)

These modes of operation reproduce the previous operation methods (groups 100, 200 and 300) with the possibility of modifying them all on fly.

**Mode 800, P2P\_SPEED\_B-L\_DIRECT:** similar to mode 100 but with the parameters of the registers in RAM editable on fly.

**Mode 820, P2P\_FORCE\_B-L\_DIRECT:** similar to mode 200 but with the parameters of the registers in RAM editable on fly.

**Mode 840, GRP\_PRP\_B-L\_DIRECT:** similar to mode 300 but with the parameters of the registers in RAM editable on fly.

In case of use of one of these operating modes or the ones with velocity and/or force remodulation, on follow there are the sub-index to modify on-fly the single parameters.

Index	Sub-index	Parameter
0x2C00	0x25	Base Position
0x2C00	0x26	Offset Position
0x2C00	0x27	Teach Position 1
0x2C00	0x28	Teach Position 2
0x2C00	0x29	Limit Position
0x2C00	0x2A	Tolerance in Position
0x2C00	0x2B	Virtual Switch1 position
0x2C00	0x2C	Virtual Switch2 position
0x2C00	0x2D	Virtual Switch3 position
0x2C00	0x2E	Virtual Switch4 position
0x2C00	0x2F	Tolerance Virtual Switch 1/2/3/4 Position
0x2C00	0x30	Drive Velocity
0x2C00	0x31	Grip Force
0x2C00	0x32	Acceleration
0x2C00	0x33	Deceleration

*Table 6 - Parameters editable on-fly*

Beware about measurement units: unlike parameters regarding positions, velocity and accelerations that maintain the same unit of measurement used by the CAN objects of the drive (i.e. based on encoder counts), **the Grip Force (0x2C00 – 0x31) is expressed in % of rated torque to facilitate user interaction.**

Please see the chapter 9.4.1 (example 7) reported at the end of this manual to better understand the usage of on-fly parameters.

#### 4.4.7. JOG operation (modes 901/902)

These mods of operation allow the JOG control of the gripper with pre-set velocity control as long as the JOG flag in the "Gripper Control Register" (0x2C00-0x03) is kept active.

**Mode 901, JOG\_POS:** JOG control in the direction of maximum position at set velocity as long as the JOG flag in the "Gripper Control Register" (0x2C00-0x03) is kept active.

**Mode 902, JOG\_NEG:** JOG control in the direction of minimum position at set velocity as long as the JOG flag in the "Gripper Control Register" (0x2C00-0x03) is kept active.

#### 4.4.8. Home operation (mode 900)

**Mode 900, HOME:** Position control at Home Position (recipe parameter) with Drive Velocity as velocity profile (recipe parameter).

In this mode the gripper performs a simple positioning of the jaws with position control and with set velocity reference. The position and velocity setpoints correspond respectively to the Limit Position and Drive Velocity parameters of the specific recipe called with this command.

#### 4.4.9. Teaching and accessing registers operation (mode 000)

In this mode of operation, the gripper does not perform any movement. This mode is used to allow the storage of the current position of the jaws as a new position value into a recipe or into the RAM position registers. In fact, with the gripper disabled, it is possible to manually move the jaws in desired positions and store the relative value.

## 4.5. Gripper parametrization

### 4.5.1. General use

In order to correctly operating, the gripper has to be parametrized to allow the embedded drive to communicate with it and to command it. This parametrization is made by Gimatic and already set in the gripper by default at purchase. As parametrization, is intended the download in the embedded drive of a specific file (extension .xdc).

In the parameters file is also included the parameters of the control loops (position, velocity and torque). This affects the behaviour of the gripper in the different operating modes and with different working scenarios (load on the jaws, acceleration and velocity required and so on).

By default, the gripper is configured to maximize the performance when operating with no load on the jaws (the file “MPCF3270 - No Load Vxx\_Template .xdc” has been downloaded). In addition, in the folder downloadable from Gimatic website, another parametrization file is available useful to maximize the performance when one wants to operate with 100% of recommended load (“MPCF3270 - Max Load Vxx\_Template.xdc”, with “load” is thus intended the mass of the gripping fingers designed by the customer).

Depending on how much load is mounted on the jaws, one has to download the respective file choosing between “No Load” or “Max Load”. In case of intermediate load values, the choose has to convert to the closer case also performing some practical tests.

Please, always refer to Table 9 for safety loads.

## 4.5.2. For expert users only: modifying control loops parameters

To reach the finest tuning of the control loops, the parameters can be modified directly by the user.

For doing this, just open one of the two available parametrization file (.xdc) with a common text editor program. Then search for the parameter that one wants to modify and change the “actual value” field. See the image below as example.

```
<CANOpenObject index="0x2500" name="Position control parameter set" objectType="8" SubNumber="8">
  <CANOpenSubObject subIndex="0x00" name="Number of entries" objectType="7" dataType="05" accessType="const" defaultValue="0x7" actualValue="7" />
  <CANOpenSubObject subIndex="0x01" name="Proportional constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x2710" actualValue="900" />
  <CANOpenSubObject subIndex="0x02" name="Integral constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x5" actualValue="0" />
  <CANOpenSubObject subIndex="0x03" name="Derivative constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x9C40" actualValue="170000" />
  <CANOpenSubObject subIndex="0x04" name="Integral antiwindup constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x0" actualValue="10" />
  <CANOpenSubObject subIndex="0x05" name="Velocity feedforward constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x0" actualValue="100" />
  <CANOpenSubObject subIndex="0x06" name="Acceleration feedforward constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x0" actualValue="0" />
  <CANOpenSubObject subIndex="0x07" name="Integral limit" objectType="7" dataType="07" accessType="rw" defaultValue="0x3E8" actualValue="10" />
</CANOpenObject>
<CANOpenObject index="0x2501" name="Velocity control parameter set" objectType="8" SubNumber="7">
  <CANOpenSubObject subIndex="0x00" name="Number of entries" objectType="7" dataType="05" accessType="const" defaultValue="0x6" actualValue="6" />
  <CANOpenSubObject subIndex="0x01" name="Proportional constant" objectType="7" dataType="07" accessType="rw" defaultValue="0xFA0" actualValue="4000" />
  <CANOpenSubObject subIndex="0x02" name="Integral constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x32" actualValue="50" />
  <CANOpenSubObject subIndex="0x03" name="Derivative constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x0" actualValue="0" />
  <CANOpenSubObject subIndex="0x04" name="Integral antiwindup constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x186A0" actualValue="100000" />
  <CANOpenSubObject subIndex="0x05" name="Acceleration feedforward constant" objectType="7" dataType="07" accessType="rw" defaultValue="0x0" actualValue="0" />
  <CANOpenSubObject subIndex="0x06" name="Integral limit" objectType="7" dataType="07" accessType="rw" defaultValue="0x1" actualValue="1" />
</CANOpenObject>
<CANOpenObject index="0x2502" name="Flux control parameter set" objectType="9" SubNumber="4">
  <CANOpenSubObject subIndex="0x00" name="Number of entries" objectType="7" dataType="05" accessType="const" defaultValue="0x3" actualValue="3" />
  <CANOpenSubObject subIndex="0x01" name="Proportional constant" objectType="7" dataType="06" accessType="rw" defaultValue="0xBB8" actualValue="25000" />
  <CANOpenSubObject subIndex="0x02" name="Integral constant" objectType="7" dataType="06" accessType="rw" defaultValue="0x12C" actualValue="6000" />
  <CANOpenSubObject subIndex="0x03" name="Constant scaling" objectType="7" dataType="05" accessType="rw" defaultValue="0x8" actualValue="1" />
</CANOpenObject>
<CANOpenObject index="0x2503" name="Torque control parameter set" objectType="9" SubNumber="4">
  <CANOpenSubObject subIndex="0x00" name="Number of entries" objectType="7" dataType="05" accessType="const" defaultValue="0x3" actualValue="3" />
  <CANOpenSubObject subIndex="0x01" name="Proportional constant" objectType="7" dataType="06" accessType="rw" defaultValue="0xBB8" actualValue="25000" />
  <CANOpenSubObject subIndex="0x02" name="Integral constant" objectType="7" dataType="06" accessType="rw" defaultValue="0x12C" actualValue="6000" />
  <CANOpenSubObject subIndex="0x03" name="Constant scaling" objectType="7" dataType="05" accessType="rw" defaultValue="0x8" actualValue="1" />
</CANOpenObject>
<CANOpenObject index="0x2504" name="Torque demand low pass filter" objectType="9" SubNumber="3">
  <CANOpenSubObject subIndex="0x00" name="Number of entries" objectType="7" dataType="05" accessType="const" defaultValue="0x2" actualValue="2" />
  <CANOpenSubObject subIndex="0x01" name="Filter enabled" objectType="7" dataType="05" accessType="rw" defaultValue="0x0" actualValue="1" />
  <CANOpenSubObject subIndex="0x02" name="Cutoff frequency" objectType="7" dataType="06" accessType="rw" defaultValue="0x64" actualValue="80" />
</CANOpenObject>
```

Figure 10 - Example of modifying the control loops parameters

For example, if one wants to modify the value of the “Proportional constant” of the position control loop, it has to navigate till the index “0x2500” (name “Position control parameter set”), subindex “0x01” and change the value in the “actualValue” field. The new value can be entered either in decimal or hexadecimal format (using the prefix “0x” in this last case).

**Disclaimer:** the user is only allowed to modify the parameters closely related to the control loops.

Additional change to other parameters that can lead to an uncontrollable state of the gripper is out of the Gimatic’s responsibility. In any case, it is sufficient to re-download one of the two provided files to restore the original values and be able again to operate with the gripper.

To download a parameters file in the drive using Gimatic specific interface, just see the appendix 9.2 (“MPCFApplication.exe”).

## 4.6. Operating without Gimatic middleware

As anticipated, one has the possibilities to avoid the middleware and communicate directly with the CAN objects.

Each object is simply a memory address in which a value is stored, the meaning of which will depend on the object. Examples of objects are the current position, or the target speed, or the current running through the motor.

The objects that can be access by users are reported here (CiA 402 object dictionary):

<https://drives.novantamotion.com/emcl/dictionaries#Dictionaries-CiA402objectdictionary>.

The presence of the middleware doesn't exclude the possibility of interaction with CAN objects. If there is this, the communication can be "less fast" (due to the cycle time of the macros present in the middleware) with the advantage to have two alternative ways to communicate. If one want to erase the middleware the communication is forced at low level with the only possibility to interact directly with the specified objects gaining in communication velocity, transforming in effect into a real time system.

Middleware is designed to prevent the customer from entering details at such a low level and don't worry about CAN objects and interacting only with Control and Status Registers. Therefore, it's particularly designed for the user not so used to using these technologies and it allows to avoid the implementation of the full CANopen standard.

**Every time one interacts with a CAN object must be care about measurement unit of the specific object.**

For example, the drive communicates position, velocity and acceleration data in encoder counts (increments). To pass from increments to millimeters it's sufficient to divide by 3200 (given by the resolution of absolute linear encoder embedded in the MPCF gripper). The current and torque are always referred in ‰ of the rated values (1.58 A, about 160 N).

Please see the following page for further information: <https://drives.novantamotion.com/emcl/torque-units>.

Note: objects with signed value are reported with 2's complement.



## 5. SYSTEM COMPONENTS

### 5.1. List of components

In case of usage of the gripper in **combination with Gimatic middleware**, the primary components of MPCF systems based are:

- Power supply
- USB/CAN converter (in case of CANbus communication)
- USB/Serial converter (in case of RS485 serial communication)
- M12 cable

Additionally, the following secondary components may be required depending on the application:

- External magnetic sensors

The components can be purchased separately from MPCF, and the ordering codes are shown in the Table 10 in chapter 9.1.

### 5.2. Power supply

For the power supply are required **24 Vdc and at minimum 5 A**.

For our laboratory test, we used a 24 Vdc-8.4 A power supply (Gimatic code: MPS24-8.4).



### 5.3. USB/CAN converter

For the correct functioning of our MPCFApplication.exe a PCAN-USB converter by PEAK System must be used.



Gimatic provides a kit (**MPCF-KIT-02**) that includes the M12 cable already wired with the PEAK converter to allow a sort of plug&play connection to the gripper and communicate with it through CANbus protocol. See the appendix 9.1 for further information.

Alternatively, using the gripper **as a generic device adhering to the CANopen DS-402 standard**, a generic USB/CAN converter can be used.

For example, the converter **Kvaser Leaf Light v2** by Kvaser AB can be utilized.



For the specific pinout and connections please refer to the relative user manual of each product.

## 5.4. USB/Serial converter

A generic USB/Serial converter adhering to the RS-485 Half Duplex standard can be used.



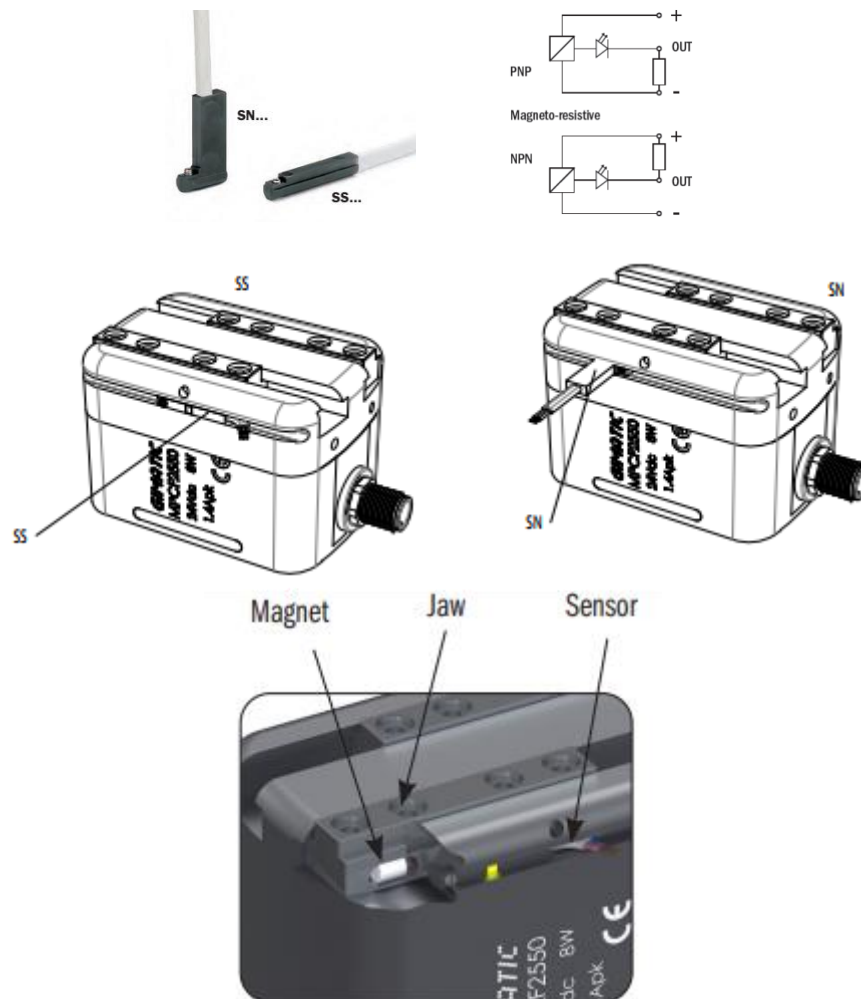
## 5.5. Sensors

MPCF grippers, like all the grippers manufactured by Gimatic, are provided with two slots (one per side) for the use of magnetic sensors.

Please note that the use of these sensors is recommended in case of need of redundant limit switch signals, since the gripper is able to communicate its currently position through the absolute encoder.


Gimatic can provide these magnetic sensors, please contact your local area Gimatic's distributor.

Please refer to the appendix for models and ordering codes of all the accessories.



## 6. SPECIFICATIONS

### 6.1. Electrical

	
	<b>MPCF3270</b>
Stroke	70 mm (2x35 mm)
Continuous (maximum) gripping force	2x80 N (2x235 N)
Minimum jaw closing time (full stroke)	0.31 s
Rated current	1.58 A
Peak current	4.8 A
Supply	24 Vdc
Torque constant	35.5 mNm/A
Feedback type	Absolute linear encoder on the jaws
Position resolution @Jaws	0.3 µm/pulse
Motor type	Brushless Motor + Hall sensor
Max allowed length of connection cable (shielded) with CANbus	5 m
Max allowed length of connection cable (shielded) with RS485	5 m

*Table 7 - Electrical specifications*

## 6.2. Mechanical

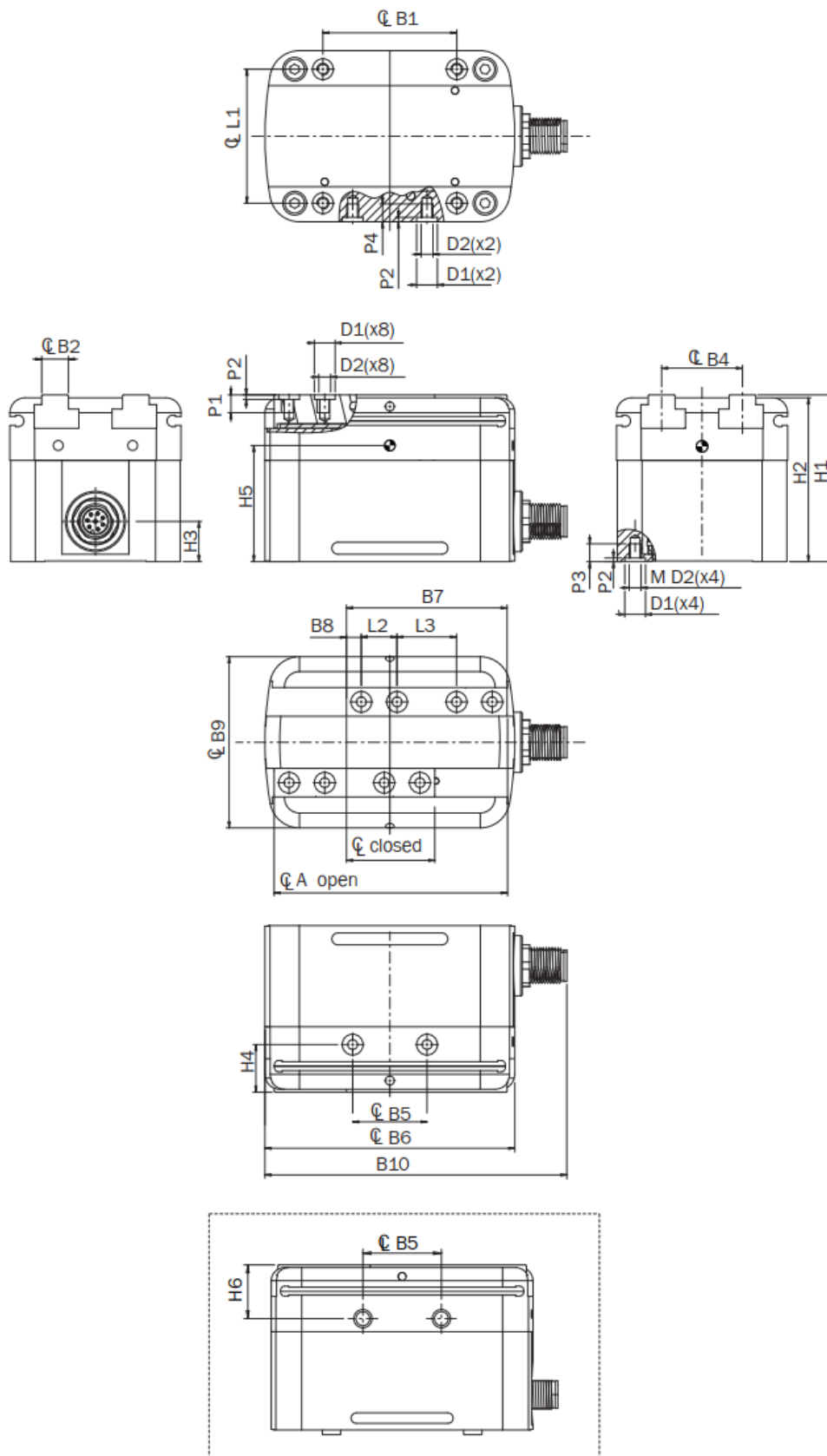


Figure 11 - MPCF3270 mechanical drawings

	MPCF3270
A	9.5
B1	54
B2	10
B3	/
B4	30,58
B5	/
B6	100
B7	60
B8	6,5
B9	62,5
B10	110
B11	30
C	25
D1	7 H8
D2	m5
H1	63
H2	62
H3	13,5
H4	/
H5	44,25
H6	20.5
L1	54
L2	14
L3	19
P1	7,2
P2	1,5
P3	8
P4	8

*Table 8 – Geometric dimensions of the MPCF gripper*

## 7. INSTALLATION

### 7.1. Unpacking

Before unpacking commences, wait for the gripper to reach room temperature to prevent condensation. Once at room temperature, remove protective packaging from the device. During installation, ensure that the actuator is kept on a clean surface away from any other magnetic and ferrous materials.

All personnel involved in transporting, storing, installing and/or maintenance of the actuator must be made aware of the potential hazards involved.

### 7.2. Mechanical



**WARNING:** Elevated surface temperatures can be present during operation of the product. Allow the gripper to cool before touching it.

#### 7.2.1. Safety loads

Check the Table 9 for the maximum permitted loads. Excessive forces or torques can damage the gripper, cause operation problems and endanger the safety of the operator.  $F_s$ ,  $M_x s$ ,  $M_y s$ ,  $M_z s$ , are the maximum permitted loads under static conditions, that is with motionless jaws.  $F_d$ ,  $M_x d$ ,  $M_y d$ ,  $M_z d$ , are the maximum permitted loads under dynamic conditions, that is with running jaws. The following table also shows the maximum permitted load (m) on each gripping tool when the gripper operates at peak performance. The Figure 12 also shows the jaw maximum backlash. The dynamic load represents the limit load at which actuators are tested with 1 million full opening and closing cycles.

	MPCF3270
$F_s$	180 N
$M_x s$	20 Nm
$M_y s$	20 Nm
$M_z s$	20 Nm
$F_d$	2 N
$M_x d$	20 Ncm
$M_y d$	20 Ncm
$M_z d$	20 Ncm
m	200 g

Table 9 - Safety loads and backlashes

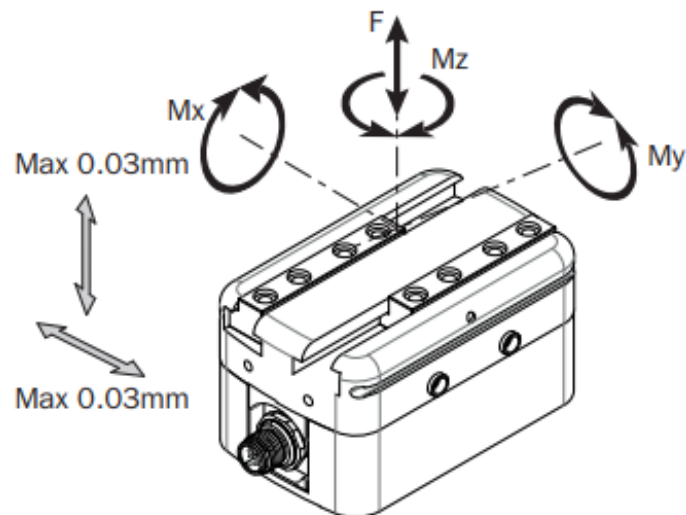


Figure 12 - MPCF3270 maximum backlashes

## 7.3. Electrical



**DANGER:** Ensure power has been completely disconnected before touching electrical connections. Electrical shock can cause serious or fatal injury.

### 7.3.1. Electromagnetic Compatibility (EMC)

While the ultimate responsibility for a system's EMC compliance lies with the system builder, the MPCF design provides good EMC performance as a system component.

The following are general recommendations when using the MPCF gripper to minimize Electromagnetic Interference (EMI) in the system.

- Keep all cable routing as short and direct as possible.
- Separate low voltage signal cables from power cables and noisy components.
- Ensure cable shielding is terminated correctly.
- Other sources of EMI in the system, such as servo drives, must also be considered for EMC, refer to component documentation for further information.

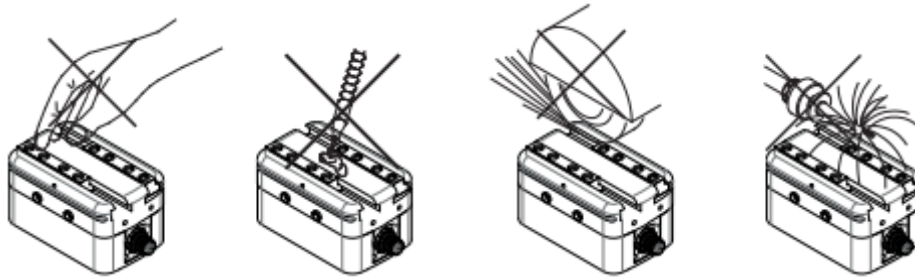


## 8. MAINTAINANCE

### 8.1. General cautions








MPCF grippers require very little maintenance. However, the following activities are recommended for periodic maintenance:


- Never let the gripper come into contact with corrosive substances, soldering splashes or abrasive powders.
- Clean any accumulated debris from the jaws.
- Ensure all parts are secured.
- Check cables for signs of wear or damage.
- After 10 million cycles, please contact Gimatic for preventive maintenance.



## 9. APPENDIX

### 9.1. Accessories

Part Number	Description	Image
3031159	2.5 m shielded cable with M12-8 pins female connector.	
MPS24-8.4 MPS24-13	Power supply unit series SP with universal AC input and 24 Vdc output (available with 8.4 A and 13 A as nominal current) by MeanWell®.	
SN4N225-G SN4M225-G SS4N225-G SS4M225-G	Magnetic proximity switches (N for PNP and M for NPN output) with 2.5 m pig tailed cable output.	
SN3N203-G SN3M203-G SS3N203-G SS3M203-G	Magnetic proximity switches (N for PNP and M for NPN output) with 0.3 m cable output length and M8 connector.	
PRO-SN4N225-G PRO-SN4M225-G PRO-SS4N225-G PRO-SS4M225-G	Programmable magnetic sensor (N for PNP and M for NPN), with 3 digital programmable outputs and 2.5 m cable.	
PRO-SN3N215-G PRO-SN3M215-G PRO-SS3M215-G PRO-SS3N215-G	Programmable magnetic sensor (N for PNP and M for NPN), with 3 digital programmable outputs and 0.3 m cable output length and M8 connector.	
PRO-SSR4N225-G PRO-SSR4M225-G	Programmable magnetic sensor (N for PNP and M for NPN), with 3 digital outputs (2 of them programmable) and 2.5 m cable.	
PRO-SSR3N215-G PRO-SSR3M215-G	Programmable magnetic sensor (N for PNP and M for NPN), with 2 digital outputs (2 of them programmable) and 0.3 m cable output length and M8 connector.	
KIT-GMPLM3240	Sterilisable silicone cover for a high standard of cleanliness and hygiene. The systems meet the most stringent GMP (Good Manufacturing Practice) cleanliness standards for Grade A and B.	

MPCF-KIT-01	Couple of jaws for MPCF3270.	
MPCF-KIT-02	KIT for feeding and communicating via CANbus with the MPCF gripper. <b>The KIT includes the cable M12 - 8 poles, wired with the D-SUB 9 pin female connector, and the Peak USB/ CAN converter. 120 Ohm resistor is already embedded in the D-SUB connector.</b>	
CONV-USB2CAN-01	USB to CANopen converter by Peak System, necessary for the correct use of MPCFApplication.exe.	
CA800120 CA803000	D-SUB 9 pin female connector (connector and cover).	

*Table 10 - List of components and relative ordering codes*

For a complete and up to date list of accessories, please refer to Gimatic's website [www.gimatic.com](http://www.gimatic.com) and Gimatic's shop <https://shop.gimatic.com/en>.

## 9.2. MPCFApplication.exe

The Windows based application “MPCFApplication.exe” developed by Gimatic allows the user to test the CANbus communication of the MPCF and to execute different movements using at maximum 8 recipes.

This application can be used to change the recipes parameters (i.e. positions and velocity, values of acceleration, deceleration, virtual limits, etc...), check the gripper status and make eventual updates made available by Gimatic.

The main purpose of the MPCFApplication.exe is to setup the gripper and make possible its testing before its integration in an automatic system.

The proper functioning of this interface is guaranteed by the middleware provided by Gimatic and default installed in the drive.

The following picture shows the user interface of MPCFApplication.exe.

**MPCF - COBOT GRIPPERS USER INTERFACE**

**General Data**

**Device Information**

Node Number: 32  
 Model: MPCF3270  
 Year: 2020  
 Batch#: 5779  
 Serial#: 1  
 Revision: 0  
 FW Version: 2.7.5

**Operation Information**

Working Hours: 00:00:06:25

**Gripper Status**

Macro State: 10 Cycle Time: 499 [ms]  
 Recipe: 0 Error Code: 0x00000000

☐ ENABLE ☐ HOME ☐ RCP\_ACT  
☐ BASE ☒ LIMIT ☒ RCP\_RDY  
☐ ERR ☐ OFFSET ☐ VS1  
☐ JOG ☐ PR\_POS ☐ VS2  
☐ TEACH ☐ TH\_POS ☒ VS3  
☐ MCMP ☐ UDP\_POS ☐ VS4

☐ POSITION SET

Act. Position: 58.15 [mm]  
 Act. Velocity: 0.00 [mm/s]  
 Act. Torque: 0.00 [%]  
 Temperature: 39.40 [°C]

Pos Limit: 70.00 - -70.00 [mm]  
 Vel Limit: 235.00 - 0.00 [mm/s]  
 Acc Limit: 1562.50 - 0.00 [mm/s^2]  
 Dec Limit: 1562.50 - 0.00 [mm/s^2]  
 Torque Limit: 100 - 0 [%]

**Gripper Control Register**

0-TEACH MODE  
 Recipe ID: 000

Enable Move Stop  
 Jog Emergency Reset

Teach Base Copy values to recipes

Base: 5.00 [mm] Virtual Sw1: 10.00 [mm]  
 Offset: 15.00 [mm] Virtual Sw2: 5.00 [mm]  
 Teach1: 12.00 [mm] Virtual Sw3: 58.00 [mm]  
 Teach2: 15.00 [mm] Virtual Sw4: 50.00 [mm]  
 Limit: 58.00 [mm] Toll.Pos Swi.: 1.00 [mm]  
 Grip Force: 90.00 [%] Toll. In Pos: 1.00 [mm]  
 Drive Velocity: 150.00 [mm/s]

Home: 30.00 [mm]  
 Limit High: 70.00 [mm] Limit Low: -70.00 [mm]  
 Acceleration: 200.00 [mm/s^2] Deceleration: 200.00 [mm/s^2]

PCAN\_ERROR\_OK SetPosition Report

Figure 13 - MPCFApplication main page

The application is structured into three divisions:

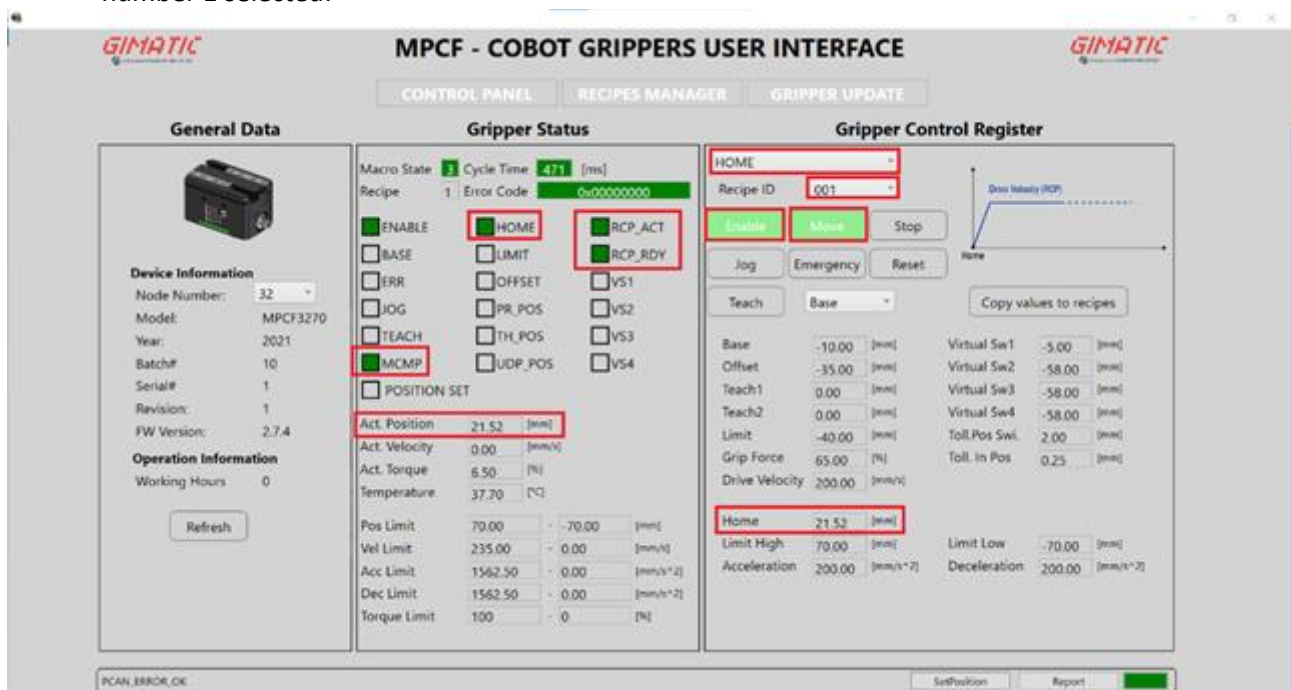
- Control panel: where is possible to check the gripper status and command the gripper.
- Recipes manager: where is possible to change, save or load the recipes parameters.
- Gripper update: where is possible to update the drive firmware, the middleware (the macros that allow the correct use of the MPCF with this application) and the control loops parameters.

## Control panel

The control panel is ideally divided into four sections:

- General data: represent the device and operation information of the connected gripper like the model, year, batch serial number, etc... this fields are not editable but are automatically read by the application once the gripper is correctly connected; click on “Refresh” button to force a newly reading operation of these values.
- Gripper status: represent the real time status of the connected gripper; through the help of different “virtual LEDs”, it is possible to know the real time value of the main bits of the “Gripper Status Register” and so in which position the gripper is stopped, if it has finished the desired movement and so on. In the bottom area there are some fields that represent the values of position, velocity, torque etc...
- Gripper control register: in this section is possible to command the gripper by selecting the desired movement and the recipe parameters; with the appropriate buttons, the user can command operations like:
  - Select a recipe
  - Select an operation mode
  - Enable the gripper (“Enable”)
  - Start the selected movement (“Move”)
  - Stop the current movement (“Stop”)
  - Make an emergency stop (“Emergency”), different from the previous one because the movement is stopped without deceleration ramp
  - Move the gripper in Jog mode (“JOG”)
  - Reset the errors occurred (“Reset”)
  - Assign the actual position value to a specific position (“Teach”, only in teaching mode)

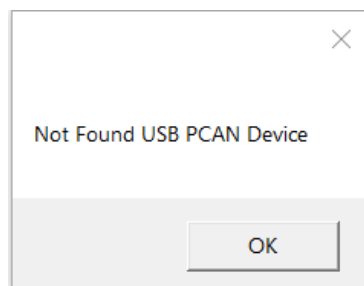
The following picture shows an example of homing movement with the parameters of the recipe number 1 selected:



Please note that if one is not in teaching mode (i.e. a movement is selected, like the picture before) the fields are coloured by grey and so are not editable; in addition, when an operation mode is selected a chart will appear to better explain what that specific movement will do.

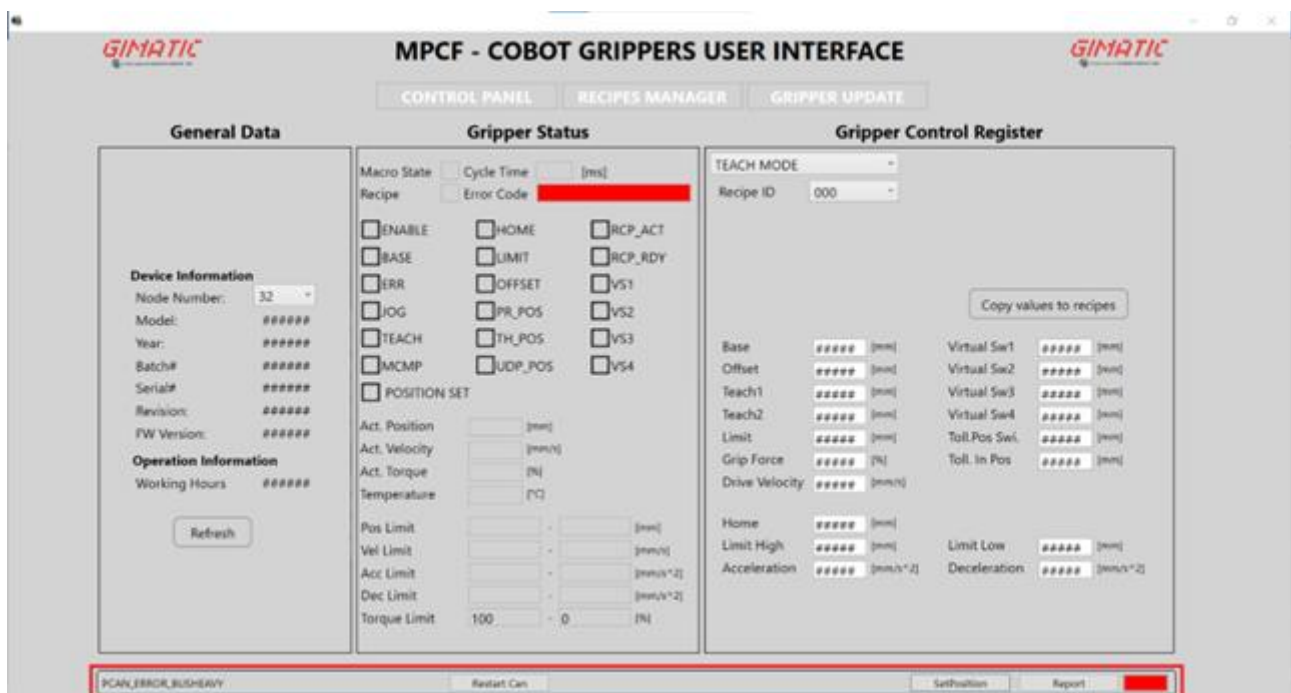
- Status bar: in the bottom of the main window one can find a log section declaring if the CAN connection is correctly established (a green LED blinks continuously and “PCAN\_ERROR\_OK” string appears), a button to set the actual position as zero position (“SetPosition”) and a button useful to generate a text file with the errors occurred during the usage of the application and the main information of the gripper (“Report”). In case of reset of the zero position, the following password must be inserted to allow the reference change: “zeroposition”. After the reference change has been complete, the application will ask to turn off and on the power supply to make the modification actual.

In case of execution of the MPCFApplication.exe with an absent or not properly CAN communication, the following error appears:



Please check the connections and try to execution newly the application.

Instead, if the application is executed without power supply, the main windows appear like below:



In the bottom status bar, the LED is coloured by red and the “PCAN\_ERROR\_BUSHEAVY” string appears.

Please power on the gripper and click on “Restart Can” button. Doing this the connection will be establish, if necessary, click on “Refresh” button to correctly load all the values.

Please note that, like previously explained, the values highlighted in the picture below are common for all the 8 recipes:

TEACH MODE

Recipe ID: 000

Buttons: Enable, Move, Stop, Jog, Emergency, Reset

Teach: Base (dropdown), Copy values to recipes

Base	-1.50 [mm]	Virtual Sw1	-10.00 [mm]
Offset	-35.00 [mm]	Virtual Sw2	-70.00 [mm]
Teach1	-58.00 [mm]	Virtual Sw3	-58.00 [mm]
Teach2	-10.00 [mm]	Virtual Sw4	-5.00 [mm]
Limit	-60.00 [mm]	Toll.Pos Swi.	0.25 [mm]
Grip Force	90.00 [%]	Toll. In Pos	0.25 [mm]
Drive Velocity	200.00 [mm/s]		
Home	-30.00 [mm]		
Limit High	70.00 [mm]	Limit Low	-70.00 [mm]
Acceleration	200.00 [mm/s^2]	Deceleration	200.00 [mm/s^2]

### Recipes manager

The picture below shows how this section appears:

MPCF - COBOT GRIPPERS USER INTERFACE

CONTROL PANEL RECIPES MANAGER GRIPPER UPDATE

Recipe

Recipe	Idx	Name	Unit	U/S	Bytes	Value
0	20	Base Position	[mm]	✓	4	0
0	21	Offset Position	[mm]	✓	4	0
0	22	Teach1 Position	[mm]	✓	4	0
0	23	Teach2 Position	[mm]	✓	4	0
0	24	Limit Position	[mm]	✓	4	0
0	25	Tolerance in Position	[mm]	✓	4	0
0	26	Virtual Switch 1 Position	[mm]	✓	4	0
0	27	Virtual Switch 2 Position	[mm]	✓	4	0
0	28	Virtual Switch 3 Position	[mm]	✓	4	0
0	29	Virtual Switch 4 Position	[mm]	✓	4	0
0	30	Tolerance Virtual Switch 1/2/3/4 Position	[mm]	✓	4	0
0	31	Drive Velocity	[mm/s]	✓	4	0
0	32	Grip Force	[%]	✓	4	0
1	36	Base Position	[mm]	✓	4	0
1	37	Offset Position	[mm]	✓	4	0
1	38	Teach1 Position	[mm]	✓	4	0
1	39	Teach2 Position	[mm]	✓	4	0

Read All Store All Save To File Load From File

PCAN\_ERROR\_OK SetPosition Report

Here is possible to find and modify all the parameters of all the 8 recipes.

With the properly buttons it is possible to:

- Read the recipe parameters actually stored in the drive memory ("Read All").
- Store the recipe parameters actually showed in the recipes manager into the drive memory ("Store All").



- Save the recipe parameters actually showed in the recipes manager to a text file (“Save To File”) or vice versa load the parameters from a file (“Load From File”).

In case of reading or writing operations, a progress bar indicates the operation progress, and a final message appears to indicate if the operation has finished without any error.

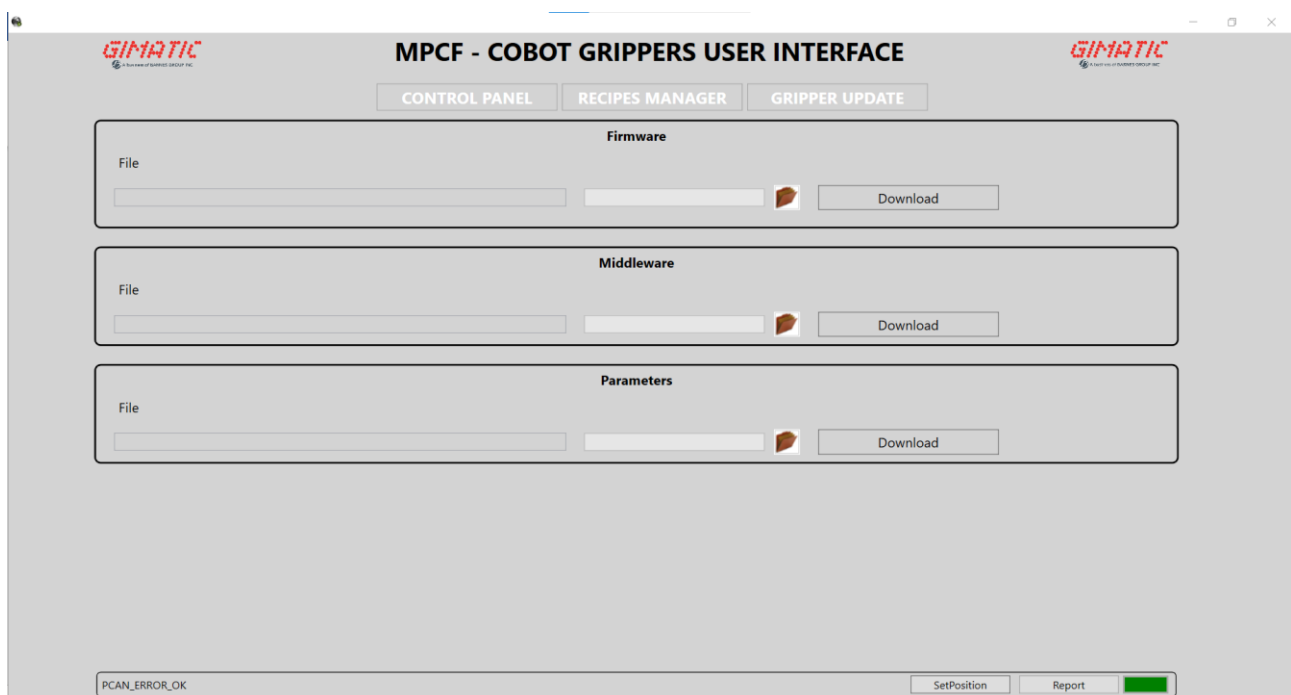
Also in this section it is possible to find the status bar in the bottom of the window.

### Gripper update

In this section, the updated files can be selected and downloaded on the gripper. Again, there are a progress bar and a final message.

The following update can be done:

- Firmware: file format .hexenc, represent the file regarding the drive firmware developed by Ingenia specifically for the Gimatic MPCF grippers and must be released by Gimatic. A properly password must be entered to allow this download to start.
- Middleware: file format .mlm, represent the macro file useful for the correct functioning of the electric gripper with the application; it must be released by Gimatic but there is the possibility to download an empty file useful to reset the macros and use the gripper in real time as a generic device adhering to the CANopen DS-402 standard.
- Parameters: file format .xdc, represent the parameters of the control loops, the generic settings about motor and encoder, virtual limits, maximum profiles for the laws of movement and so on set by default in Gimatic HQ. It is strongly recommended to use only parameters shared by Gimatic. However, if one wants to modify some values, must do it through a common text editor (changing the needed parameters from the XML file) and starting from the Gimatic original file (see chapter 4.5.2). Download the updated file on the drive to make the changes.



In order to avoid accidental errors that can damage the correct functioning of the MPCFApplication.exe or gripper, the password useful to update the drive FW will be shared by Gimatic (or by the customer distributor) only when an update is available and necessary. If a firmware update is required, a successive update of both middleware and parameters is mandatory (in this order).

Please note that if one wants make use of the MPCF as a generic CANOpen slave, it is necessary to reset the macros by update the middleware with a blank file provided by Gimatic (please refer to example 6).



### 9.2.1. Examples of interacting with MPCFApplication.exe

On the following, will be show some examples of use of the gripper in association with the MPCFApplication.exe. All major interface features will be touched, like teaching a new position (example 1), make movements using the recipe parameters (example 2), make movements with remodulation of velocity and force (example 3), make movements using all on-fly parameters (example 4), set a new zero position (example 5) and finally erase the middleware to operate the MPCF as a generic CAN slave.

#### 9.2.1.1.Example 1 - Teaching new recipe parameters

To modify the recipes parameters, one must go to the recipes manager section. In this section, the customer can modify all the recipes parameters. Alternately, it is possible to first make changes directly from the control panel by selecting the “TEACH” operation mode and then go in the recipes manager only to save the parameters on the drive.

After selecting the Teach mode, the values fields turn white as back colour, and they become editable.

In this example will be shown how to change the Home Position (common for all the recipes) and the Base and Limit Position of the recipe number 006.

Position	Old value [mm]	New value [mm]
Home	-30	21.52
Base	-10	8
Limit	-50	55

The customer can manually write the new values by clicking the proper field or, in case of a change regarding a position, move the jaws in the desired position, select by the appropriate drop-down menu the position that one want modify and then click on the “Teach” button. With this latter method, the value shown in the “Actual Position” will be store in the specified position.

Macro State 4 Cycle Time 468 [ms]  
 Recipe 6 Error Code 0x00000000

☐ ENABLE   ☐ HOME   ☐ RCP\_ACT  
☐ BASE   ☐ LIMIT   ☒ RCP\_RDY  
☐ ERR   ☐ OFFSET   ☐ VS1  
☐ JOG   ☐ PR\_POS   ☐ VS2  
☐ TEACH   ☐ TH\_POS   ☐ VS3  
☐ MCMP   ☒ UDP\_POS   ☐ VS4  
☐ POSITION SET

Act. Position 21.52 [mm]  
 Act. Velocity 0.00 [mm/s]  
 Act. Torque 0.00 [%]  
 Temperature 37.70 [°C]

Pos Limit 70.00 - -70.00 [mm]  
 Vel Limit 235.00 - 0.00 [mm/s]  
 Acc Limit 1562.50 - 0.00 [mm/s^2]

TEACH MODE  
 Recipe ID 006

Enable Move Stop  
 Jog Emergency Reset

Teach Home Copy values to recipes

Base -10.00 [mm] Virtual Sw1 0.00 [mm]  
 Offset 0.00 [mm] Virtual Sw2 -58.00 [mm]  
 Teach1 -58.00 [mm] Virtual Sw3 0.00 [mm]  
 Teach2 -5.00 [mm] Virtual Sw4 0.00 [mm]  
 Limit -50.00 [mm] Toll.Pos Swi. 1342172 [mm]  
 Grip Force 100.00 [%] Toll. In Pos 0.50 [mm]  
 Drive Velocity 150.00 [mm/s]

Home -30.00 [mm]  
 Limit High 70.00 [mm] Limit Low -70.00 [mm]  
 Acceleration 200.00 [mm/s^2] Deceleration 200.00 [mm/s^2]

Macro State <b>5</b> Cycle Time <b>479</b> [ms]	<b>TEACH MODE</b>	
Recipe <b>6</b> Error Code <b>0x00000000</b>	Recipe ID <b>006</b>	
<input type="checkbox"/> ENABLE <input checked="" type="checkbox"/> HOME <input type="checkbox"/> RCP_ACT <input type="checkbox"/> BASE <input type="checkbox"/> LIMIT <input checked="" type="checkbox"/> RCP_RDY <input type="checkbox"/> ERR <input type="checkbox"/> OFFSET <input type="checkbox"/> VS1 <input type="checkbox"/> JOG <input type="checkbox"/> PR_POS <input type="checkbox"/> VS2 <input checked="" type="checkbox"/> TEACH <input type="checkbox"/> TH_POS <input type="checkbox"/> VS3 <input type="checkbox"/> MCMP <input type="checkbox"/> UDP_POS <input type="checkbox"/> VS4 <input type="checkbox"/> POSITION SET	Enable Move Stop Jog Emergency Reset <input checked="" type="button"/> Teach <input type="button"/> Home <input type="button"/> Copy values to recipes	
Act. Position <b>21.52</b> [mm] Act. Velocity <b>0.00</b> [mm/s] Act. Torque <b>0.00</b> [%] Temperature <b>39.40</b> [°C]	Base <b>-10.00</b> [mm] Virtual Sw1 <b>0.00</b> [mm] Offset <b>0.00</b> [mm] Virtual Sw2 <b>-58.00</b> [mm] Teach1 <b>-58.00</b> [mm] Virtual Sw3 <b>0.00</b> [mm] Teach2 <b>-5.00</b> [mm] Virtual Sw4 <b>0.00</b> [mm] Limit <b>-50.00</b> [mm] Toll.Pos Swi. <b>1342172</b> [mm] Grip Force <b>100.00</b> [%] Toll. In Pos <b>0.50</b> [mm] Drive Velocity <b>150.00</b> [mm/s]	
Pos Limit <b>70.00</b> - <b>-70.00</b> [mm] Vel Limit <b>235.00</b> - <b>0.00</b> [mm/s] Acc Limit <b>1562.50</b> - <b>0.00</b> [mm/s^2]	<input checked="" type="button"/> Home <b>21.52</b> [mm] Limit High <b>70.00</b> [mm] Limit Low <b>-70.00</b> [mm] Acceleration <b>200.00</b> [mm/s^2] Deceleration <b>200.00</b> [mm/s^2]	

Similarly for Base and Limit Position.

Macro State <b>5</b> Cycle Time <b>468</b> [ms]	<b>TEACH MODE</b>	
Recipe <b>6</b> Error Code <b>0x00000000</b>	Recipe ID <b>006</b>	
<input type="checkbox"/> ENABLE <input type="checkbox"/> HOME <input type="checkbox"/> RCP_ACT <input type="checkbox"/> BASE <input type="checkbox"/> LIMIT <input checked="" type="checkbox"/> RCP_RDY <input type="checkbox"/> ERR <input type="checkbox"/> OFFSET <input type="checkbox"/> VS1 <input type="checkbox"/> JOG <input type="checkbox"/> PR_POS <input type="checkbox"/> VS2 <input type="checkbox"/> TEACH <input type="checkbox"/> TH_POS <input type="checkbox"/> VS3 <input type="checkbox"/> MCMP <input checked="" type="checkbox"/> UDP_POS <input type="checkbox"/> VS4 <input type="checkbox"/> POSITION SET	Enable Move Stop Jog Emergency Reset <input type="button"/> Teach <input type="button"/> Home <input type="button"/> Copy values to recipes	
Act. Position <b>64.89</b> [mm] Act. Velocity <b>0.00</b> [mm/s] Act. Torque <b>0.00</b> [%] Temperature <b>40.40</b> [°C]	Base <b>8</b> [mm] Virtual Sw1 <b>6</b> [mm] Offset <b>40</b> [mm] Virtual Sw2 <b>38</b> [mm] Teach1 <b>45</b> [mm] Virtual Sw3 <b>0.00</b> [mm] Teach2 <b>50</b> [mm] Virtual Sw4 <b>0.00</b> [mm] Limit <b>55</b> [mm] Toll.Pos Swi. <b>0.50</b> [mm] Grip Force <b>90</b> [%] Toll. In Pos <b>0.50</b> [mm] Drive Velocity <b>150.00</b> [mm/s]	
Pos Limit <b>70.00</b> - <b>-70.00</b> [mm] Vel Limit <b>235.00</b> - <b>0.00</b> [mm/s] Acc Limit <b>1562.50</b> - <b>0.00</b> [mm/s^2]	Home <b>21.52</b> [mm] Limit High <b>70.00</b> [mm] Limit Low <b>-70.00</b> [mm] Acceleration <b>200.00</b> [mm/s^2] Deceleration <b>200.00</b> [mm/s^2]	

Please make sure that one is modifying the values regarding the desired recipe.

To definitively store the new values in the drive, one must click on “Copy values to recipes”, go into the recipes manager (that it will show the values like just modified) and click on “Store All”.

Recipe	Idx	Name	Unit	U/S	Bytes	Value
5	110	Tolerance Virtual Switch 1/2/3/4 Position	[mm]	<input type="checkbox"/>	4	1.00
5	111	Drive Velocity	[mm/s]	<input type="checkbox"/>	4	100.00
5	112	Grip Force	[%]	<input type="checkbox"/>	4	80.00
6	116	Base Position	[mm]	<input checked="" type="checkbox"/>	4	8
6	117	Offset Position	[mm]	<input checked="" type="checkbox"/>	4	40
6	118	Teach1 Position	[mm]	<input checked="" type="checkbox"/>	4	45
6	119	Teach2 Position	[mm]	<input checked="" type="checkbox"/>	4	50
6	120	Limit Position	[mm]	<input checked="" type="checkbox"/>	4	55
6	121	Tolerance in Position	[mm]	<input type="checkbox"/>	4	0.5
6	122	Virtual Switch 1 Position	[mm]	<input checked="" type="checkbox"/>	4	6
6	123	Virtual Switch 2 Position	[mm]	<input checked="" type="checkbox"/>	4	38
6	124	Virtual Switch 3 Position	[mm]	<input checked="" type="checkbox"/>	4	0
6	125	Virtual Switch 4 Position	[mm]	<input checked="" type="checkbox"/>	4	0
6	126	Tolerance Virtual Switch 1/2/3/4 Position	[mm]	<input type="checkbox"/>	4	0.5
6	127	Drive Velocity	[mm/s]	<input type="checkbox"/>	4	150
6	128	Grip Force	[%]	<input type="checkbox"/>	4	90
7	132	Base Position	[mm]	<input checked="" type="checkbox"/>	4	-50.00

Read All Store All Save To File Load From File

Note that the parameters that have been changed are coloured by white before to perform a writing procedure.

### 9.2.1.2.Example 2 - Make a movement from Base to Limit position using recipe 006

The following example shows how to command a movement from Base Position to Limit Position of the gripper using recipe 006. It is not expected the gripping of a piece. In fact, in case of an obstacle along the stroke, an error can occur. The operation mode corresponding to this movement is the 100.

First, one must select the recipe and the movement desired.

Macro State **9** Cycle Time **463** [ms]  
Recipe **6** Error Code **0x00000000**

☐ ENABLE ☐ HOME ☒ RCP\_ACT  
☒ BASE ☐ LIMIT ☒ RCP\_RDY  
☐ ERR ☐ OFFSET ☐ VS1  
☐ JOG ☐ PR\_POS ☐ VS2  
☐ TEACH ☐ TH\_POS ☐ VS3  
☒ MCMP ☐ UDP\_POS ☐ VS4  
☐ POSITION SET

Act. Position **8.03** [mm]  
Act. Velocity **0.00** [mm/s]  
Act. Torque **-1.00** [%]  
Temperature **44.00** [°C]  
  
Pos Limit **70.00** - **-70.00** [mm]  
Vel Limit **235.00** - **0.00** [mm/s]  
Acc Limit **1562.50** - **0.00** [mm/s^2]  
Dec Limit **1562.50** - **0.00** [mm/s^2]  
Torque Limit **100** - **0** [%]

P2P\_SPEED\_B - L\_RCP  
Recipe ID **006**

Enable Move Stop  
Jog Emergency Reset

Teach Home Copy values to recipes

Base **8.00** [mm] Virtual Sw1 **6.00** [mm]  
Offset **40.00** [mm] Virtual Sw2 **38.00** [mm]  
Teach1 **45.00** [mm] Virtual Sw3 **0.00** [mm]  
Teach2 **50.00** [mm] Virtual Sw4 **0.00** [mm]  
Limit **55.00** [mm] Toll.Pos Swi. **0.50** [mm]  
Grip Force **100.00** [%] Toll. In Pos **0.50** [mm]  
Drive Velocity **150.00** [mm/s]

Home **21.52** [mm] Limit High **70.00** [mm] Limit Low **-70.00** [mm]  
Acceleration **200.00** [mm/s^2] Deceleration **200.00** [mm/s^2]

If everything goes right, the LEDs corresponding to “RCP\_ACT” and “RCP\_RDY” must turn on indicating that the recipe and the operation mode are correctly uploaded.

Note the chart describing the operation mode just selected in the top-right angle of the window. This operation mode is a simple point to point motion with a velocity setpoint equals to recipe parameter “Drive Velocity”. The “Grip Force” parameter is not considered in this case.

The next step is to enable the gripper and start the movement.

For doing this, it is sufficient to click on “Enable” button and then on “Move” button.

The screenshot displays the control interface for a Gimatic MPCF integrated drive gripper. The interface is split into two main sections.

**Left Panel (Status and Controls):**

- Macro State:** 6
- Cycle Time:** 433 [ms]
- Recipe:** 6
- Error Code:** 0x00000000
- Control Buttons:**
  - ☒ ENABLE (highlighted with a red box)
  - ☐ HOME
  - ☒ LIMIT (highlighted with a red box)
  - ☐ RCP\_ACT
  - ☐ BASE
  - ☐ RCP\_RDY
  - ☐ ERR
  - ☐ OFFSET
  - ☐ VS1
  - ☐ JOG
  - ☒ PR\_POS
  - ☐ VS2
  - ☐ TEACH
  - ☐ TH\_POS
  - ☐ VS3
  - ☒ MCMP (highlighted with a red box)
  - ☐ UDP\_POS
  - ☐ VS4
  - ☐ POSITION SET
- Act. Position:** 54.93 [mm] (highlighted with a red box)
- Act. Velocity:** 0.00 [mm/s]
- Act. Torque:** -2.10 [%]
- Temperature:** 44.00 [°C]
- Position Limits:**
  - Pos Limit: 70.00 - -70.00 [mm]
  - Vel Limit: 235.00 - 0.00 [mm/s]
  - Acc Limit: 1562.50 - 0.00 [mm/s^2]
  - Dec Limit: 1562.50 - 0.00 [mm/s^2]
  - Torque Limit: 100 - 0 [%]

**Right Panel (Recipe Settings and Graph):**

- P2P\_SPEED\_B - L\_RCP** (dropdown menu)
- Recipe ID:** 006
- Buttons:** Enable (highlighted with a red box), Move (highlighted with a red box), Stop, Jog, Emergency, Reset.
- Teach:** Home (dropdown menu)
- Copy values to recipes** (button)
- Position Settings:**
  - Base: 8.00 [mm]
  - Offset: 40.00 [mm]
  - Teach1: 45.00 [mm]
  - Teach2: 50.00 [mm]
  - Limit: 55.00 [mm] (highlighted with a red box)
  - Grip Force: 100.00 [%]
  - Drive Velocity: 150.00 [mm/s]
- Virtual Switches:**
  - Virtual Sw1: 6.00 [mm]
  - Virtual Sw2: 38.00 [mm]
  - Virtual Sw3: 0.00 [mm]
  - Virtual Sw4: 0.00 [mm]
  - Toll.Pos Swi.: 0.50 [mm]
  - Toll. In Pos: 0.50 [mm]
- Home and Limit Settings:**
  - Home: 21.52 [mm]
  - Limit High: 70.00 [mm]
  - Acceleration: 200.00 [mm/s^2]
  - Limit Low: -70.00 [mm]
  - Deceleration: 200.00 [mm/s^2]

**Graph:** A line graph titled "Drive Velocity (RCP)" showing a trapezoidal profile. The x-axis is labeled with "Home", "Base", and "Limit". The y-axis represents velocity.

Please note that the gripper will be in enable mode as long as the “Enable” button is pressed, while the movement will start on the pressing of the “Move” button independently if the button remains clicked or not.

The movement will stop when the setpoint position will be reach or in case of click on “Stop” (or “Emergency”) button.

### 9.2.1.3.Example 3 - Make a movement from Limit to Base Position with force control and velocity remodulation using recipe 006

In this example, it will be shown the opposite movement respect to the previous example. However, in this case, it is performed a force control. That meaning that if the jaws meet an object during their stroke, the gripper will hold the piece.

In addition, this operation mode (211) includes the possibility to change the velocity setpoint and so using a different value respect to the one stored in the recipe (in this case the 006 again).

Like usual, the first thing to do is to select the desired movement and the recipe.

Macro State 9 Cycle Time 465 [ms]  
 Recipe 6 Error Code 0x00000000

☐ ENABLE ☐ HOME ☒ RCP\_ACT  
☐ BASE ☒ LIMIT ☒ RCP\_RDY  
☐ ERR ☐ OFFSET ☐ VS1  
☐ JOG ☒ PR\_POS ☐ VS2  
☐ TEACH ☐ TH\_POS ☐ VS3  
☒ MCMP ☐ UDP\_POS ☐ VS4  
☐ POSITION SET

Act. Position 54.93 [mm]  
 Act. Velocity 0.00 [mm/s]  
 Act. Torque 1.10 [%]  
 Temperature 36.80 [°C]

Pos Limit 70.00 - -70.00 [mm]  
 Vel Limit 235.00 - 0.00 [mm/s]  
 Acc Limit 1562.50 - 0.00 [mm/s^2]  
 Dec Limit 1562.50 - 0.00 [mm/s^2]  
 Torque Limit 100 - 0 [%]

P2P\_FORCE L - B REG V  
 Recipe ID 006

Enable Move Stop  
 Jog Emergency Reset

Teach Base Copy values to recipes

Base 8.00 [mm]  
 Offset 40.00 [mm]  
 Teach1 45.00 [mm]  
 Teach2 50.00 [mm]  
 Limit 55.00 [mm]  
 Grip Force 90.00 [%]  
 Drive Velocity 150.00 [mm/s]

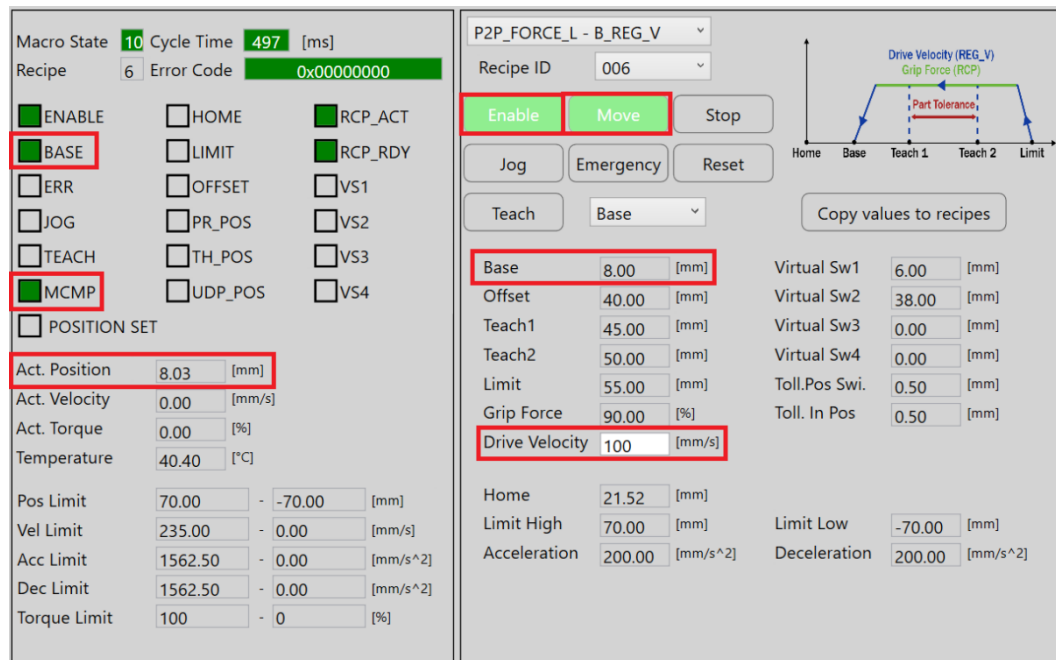
Virtual Sw1 6.00 [mm]  
 Virtual Sw2 38.00 [mm]  
 Virtual Sw3 0.00 [mm]  
 Virtual Sw4 0.00 [mm]  
 Toll.Pos Swi. 0.50 [mm]  
 Toll. In Pos 0.50 [mm]

Home 21.52 [mm]  
 Limit High 70.00 [mm]  
 Acceleration 200.00 [mm/s^2]  
 Limit Low -70.00 [mm]  
 Deceleration 200.00 [mm/s^2]

Note the two LEDs (“RCP\_ACT” and “RCP\_RDY”) turn on and especially the “Drive Velocity” parameter that become editable (white as back colour).

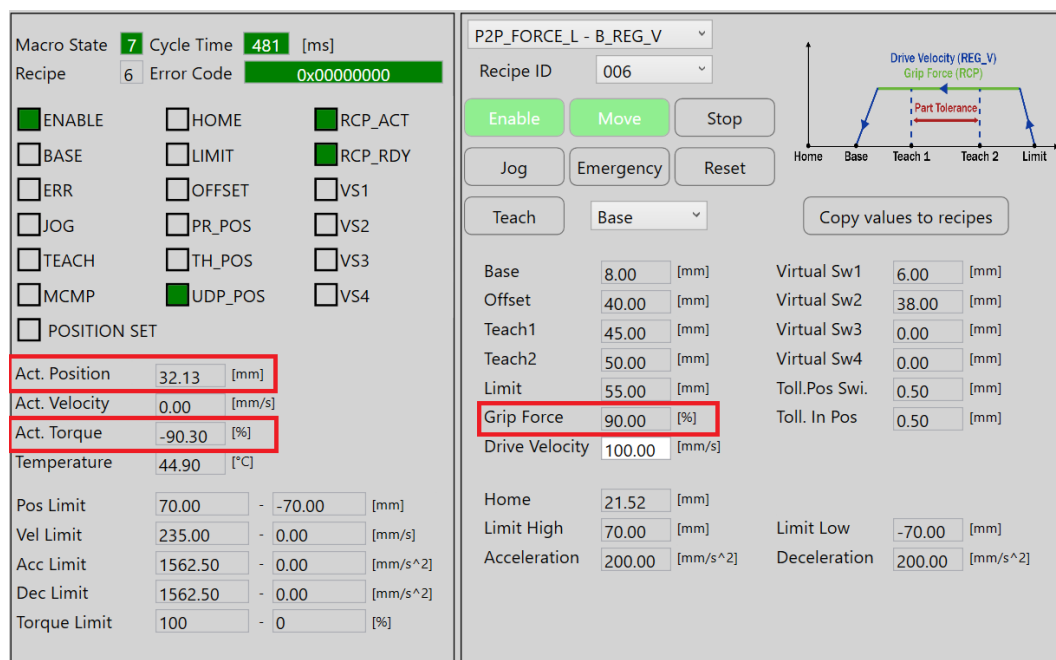
Insert the desired velocity setpoint value, in this example equals to 100 mm/s, and click on “Enable” button first and “Move” button then to start the movement.

Base	8.00	[mm]	Virtual Sw1	6.00	[mm]
Offset	40.00	[mm]	Virtual Sw2	38.00	[mm]
Teach1	45.00	[mm]	Virtual Sw3	0.00	[mm]
Teach2	50.00	[mm]	Virtual Sw4	0.00	[mm]
Limit	55.00	[mm]	Toll.Pos Swi.	0.50	[mm]
Grip Force	90.00	[%]	Toll. In Pos	0.50	[mm]
Drive Velocity	100	[mm/s]			



In our case, no obstacles were detected and therefore the movement ended at Limit Position.

If we repeat the same steps but with a piece to grip along the gripper stroke, the result is the following:



Note that the torque developed by the gripper is quite equals to the force setpoint (90%).

If one change newly the operation mode to one that expect only recipe parameters, the “Drive Velocity” value returns equals to the stored one (150 mm/s).

### 9.2.1.4.Example 4 - Make a movement from Base to Limit Position with force control, pre-positioning and all recipe parameters execute on-fly

In this example, it will be shown the operation mode 840. This mode consists in a movement with force control and pre-positioning (at the Offset Position) and in addition all the parameters can be editable on-fly (i.e. in real time), allowing the user to completely customize the movement to be carried out.

After the recipe and the desired movement have been selected, one can modify every value of the recipe (it is intuitive from the white colour as back colour of the edit box).

Click on “Enable” and “Move” button to start the movement.

Like the previous example, if there is no object along the gripper stroke, the movement will complete at the Limit Position (in this case modified at 20 mm).

Macro State	4	Cycle Time	548 [ms]
Recipe	6	Error Code	0x00000000

<input checked="" type="checkbox"/> ENABLE	<input type="checkbox"/> HOME	<input type="checkbox"/> RCP_ACT
<input type="checkbox"/> BASE	<input checked="" type="checkbox"/> LIMIT	<input checked="" type="checkbox"/> RCP_RDY
<input type="checkbox"/> ERR	<input type="checkbox"/> OFFSET	<input type="checkbox"/> VS1
<input type="checkbox"/> JOG	<input type="checkbox"/> PR_POS	<input type="checkbox"/> VS2
<input type="checkbox"/> TEACH	<input type="checkbox"/> TH_POS	<input type="checkbox"/> VS3
<input checked="" type="checkbox"/> MCMP	<input type="checkbox"/> UDP_POS	<input type="checkbox"/> VS4
<input type="checkbox"/> POSITION SET		

Act. Position	20.24 [mm]
Act. Velocity	-32.13 [mm/s]
Act. Torque	-8.60 [%]
Temperature	45.80 [°C]

Pos Limit	70.00	-70.00 [mm]
Vel Limit	235.00	0.00 [mm/s]
Acc Limit	1562.50	0.00 [mm/s^2]
Dec Limit	1562.50	0.00 [mm/s^2]
Torque Limit	100	0 [%]

GPR\_PRP\_B-L\_DIRECT  
Recipe ID 006  
    
    
Teach Base

Base	50 [mm]	Virtual Sw1	50 [mm]
Offset	38 [mm]	Virtual Sw2	38.00 [mm]
Teach1	30 [mm]	Virtual Sw3	30 [mm]
Teach2	40 [mm]	Virtual Sw4	40 [mm]
Limit	20 [mm]	Toll.Pos Swi.	0.50 [mm]
Grip Force	80 [%]	Toll. In Pos	0.50 [mm]
Drive Velocity	130 [mm/s]		

Home	21.52 [mm]	Limit Low	-70.00 [mm]
Limit High	70.00 [mm]	Deceleration	200.00 [mm/s^2]
Acceleration	200.00 [mm/s^2]		

If instead a piece is clamped, then the jaws will keep the position with a torque equivalent to the one inserted as setpoint (80%).

Macro State	8	Cycle Time	527 [ms]
Recipe	6	Error Code	0x00000000

<input checked="" type="checkbox"/> ENABLE	<input type="checkbox"/> HOME	<input type="checkbox"/> RCP_ACT
<input type="checkbox"/> BASE	<input type="checkbox"/> LIMIT	<input checked="" type="checkbox"/> RCP_RDY
<input type="checkbox"/> ERR	<input type="checkbox"/> OFFSET	<input type="checkbox"/> VS1
<input type="checkbox"/> JOG	<input type="checkbox"/> PR_POS	<input type="checkbox"/> VS2
<input type="checkbox"/> TEACH	<input checked="" type="checkbox"/> TH_POS	<input type="checkbox"/> VS3
<input type="checkbox"/> MCMP	<input type="checkbox"/> UDP_POS	<input type="checkbox"/> VS4
<input type="checkbox"/> POSITION SET		

Act. Position	32.45 [mm]
Act. Velocity	0.00 [mm/s]
Act. Torque	-79.50 [%]
Temperature	45.80 [°C]

Pos Limit	70.00	-70.00 [mm]
Vel Limit	235.00	0.00 [mm/s]
Acc Limit	1562.50	0.00 [mm/s^2]
Dec Limit	1562.50	0.00 [mm/s^2]
Torque Limit	100	0 [%]

GPR\_PRP\_B-L\_DIRECT  
Recipe ID 006  
    
    
Teach Base

Base	50 [mm]	Virtual Sw1	50 [mm]
Offset	38 [mm]	Virtual Sw2	38 [mm]
Teach1	30 [mm]	Virtual Sw3	30 [mm]
Teach2	40 [mm]	Virtual Sw4	40 [mm]
Limit	20 [mm]	Toll.Pos Swi.	0.50 [mm]
Grip Force	80.00 [%]	Toll. In Pos	0.50 [mm]
Drive Velocity	130.00 [mm/s]		

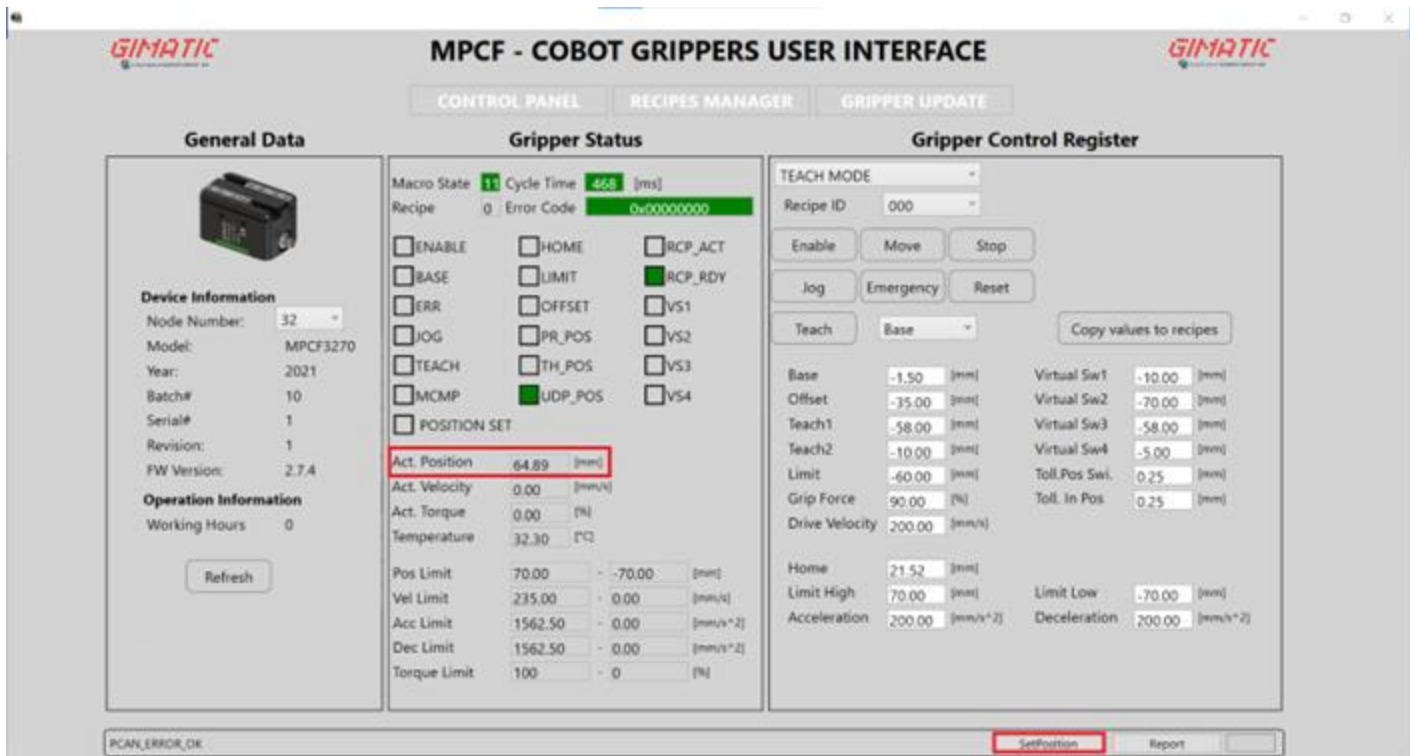
Home	21.52 [mm]	Limit Low	-70.00 [mm]
Limit High	70.00 [mm]	Deceleration	200.00 [mm/s^2]
Acceleration	200.00 [mm/s^2]		

Note also that the “TH\_POS” LED turns on because the jaws stopped in a position between Teach 1 and Teach 2 Position.

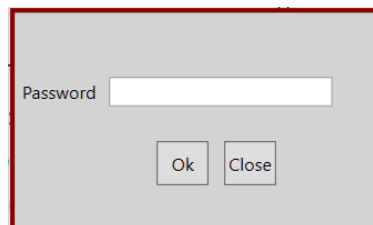


### 9.2.1.5.Example 5 - Set new zero position

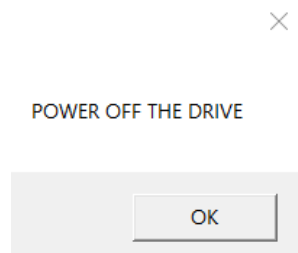
For setting a new zero position, it is sufficient to move the jaws in the desired position and click on the “SetPosition” button in the bottom-right angle of the window.



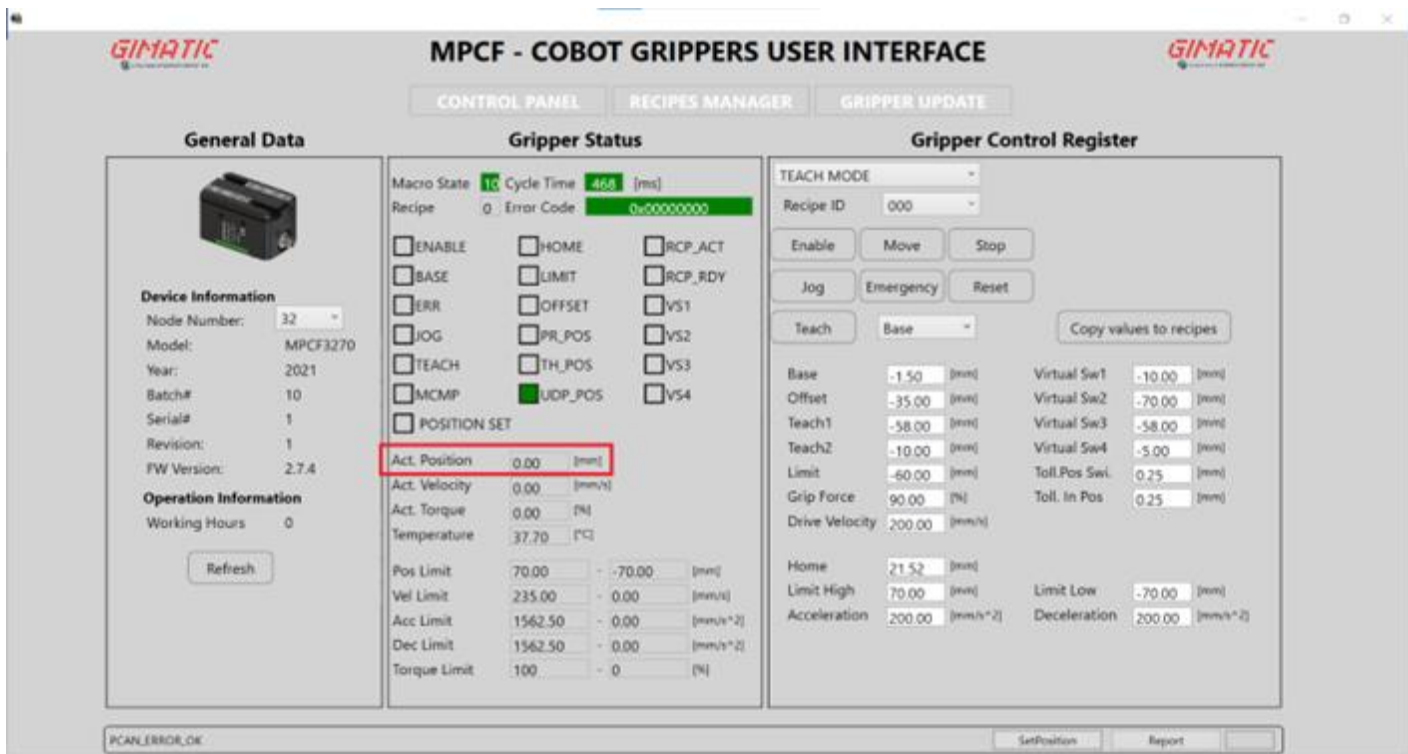
After clicking the button, a message will be displayed asking to insert a valid password (just to be sure to avoid unintended changes by the user). Please insert the following password: “zeroposition” and click “Ok”.



If the password just inserted is correct, a new message will be displayed asking to power off the drive to make the modify available.



After turning off the drive, please wait a few seconds to allow the fully shutdown of the gripper. Then, turn it on again and click the “OK” button and verify that the new zero position has been effectively modified.

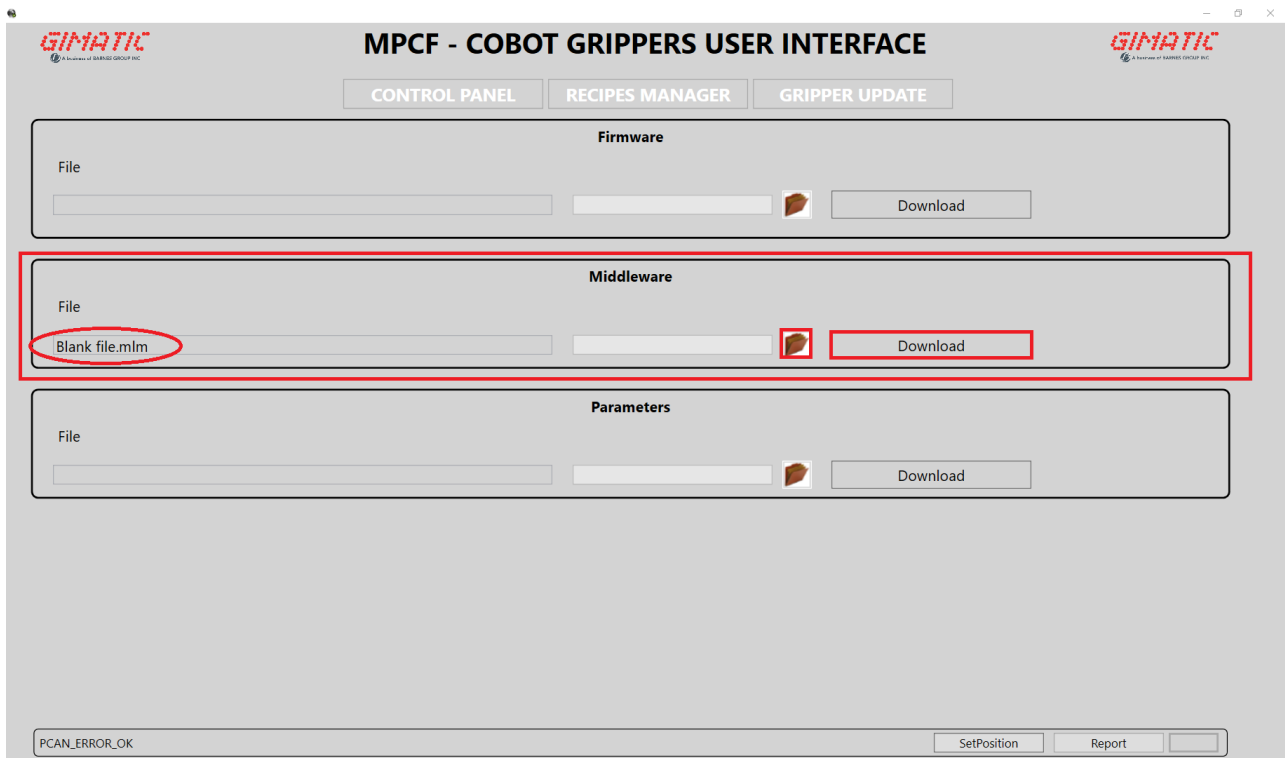


In case of problems an error will be displayed in the appropriate area of the window ("Error code").

### 9.2.1.6.Example 6 - Erase the middleware

For resetting the macros and be able to use the gripper as a generic CANOpen slave, a blank file must be uploaded as middleware.

Please choose the path of the blank file, select it and start the download.



A successful message will appear indicating that the operation went well.



### 9.3. MPCFSerialDemo.exe

An additional Windows based application is introduced (“MPCFSerialDemo.exe”) as a tool to test and to familiarize with the serial communication protocol.

This interface, even this developed by Gimatic, allows to execute basic movements using recipes already set through the MPCFApplication.exe application and to view the raw data generated by the reading and writing of registers for the case of the most used commands. Additional tests can also be performed by enabling the expert mode’s grating that guarantees a full access to the Control register.

Only one application must be active at a time and it is not possible to change the recipes parameters through the MPCFSerialDemo.exe application (use the MPCFApplication.exe instead).

The main purpose of the MPCFSerialDemo.exe is to make possible the testing of the gripper before its integration into an automatic system (like collaborative robots, for example).

The following picture shows the user interface of MPCFSerialDemo.exe:

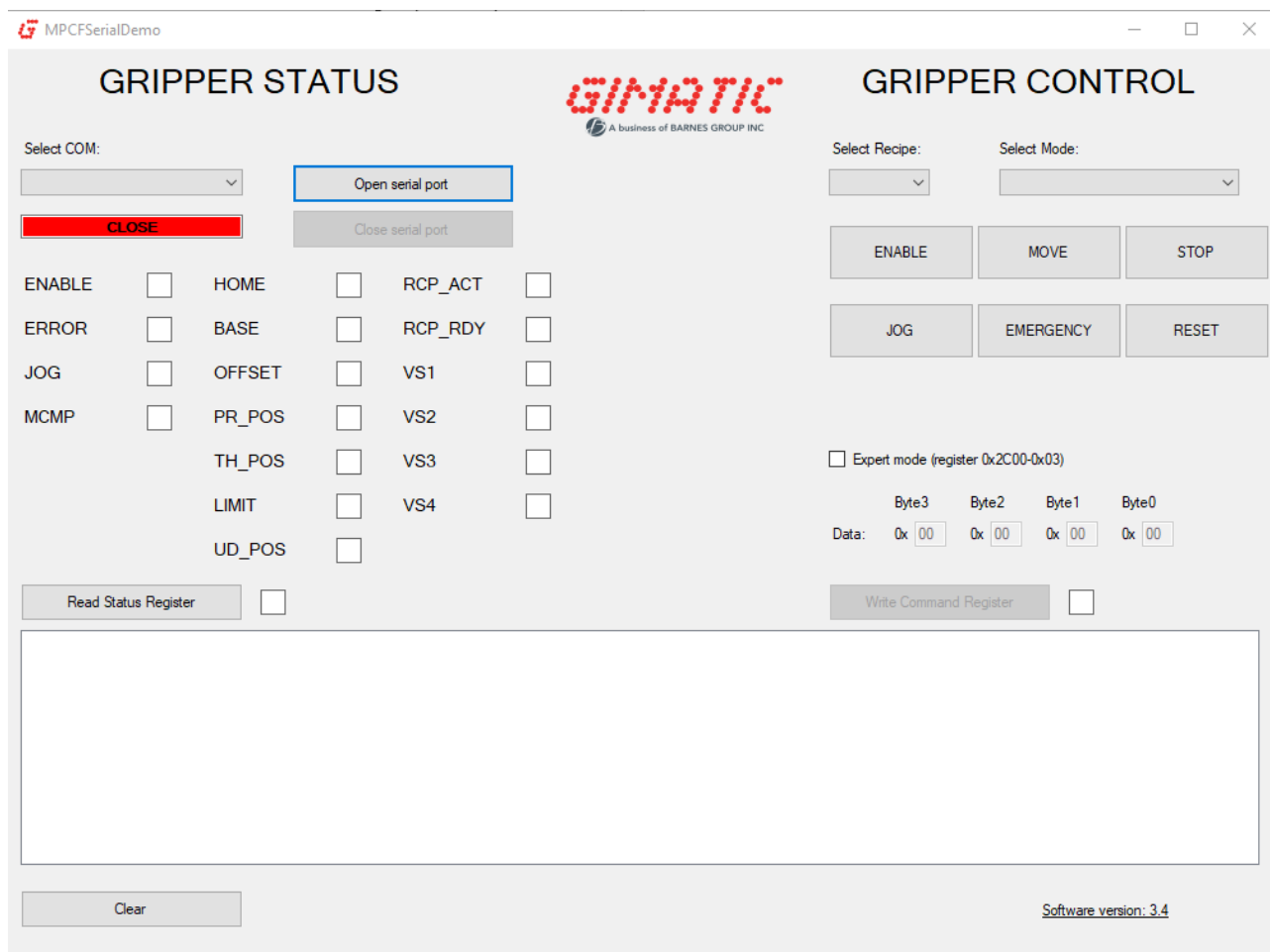


Figure 14 - MPCFSerialDemo interface

As for the MPCFApplication.exe, the proper functioning of this interface is guaranteed by the middleware provided by Gimatic and default installed in the drive.

For details about the MPCFSerialDemo.exe and the serial communication, please refer to the “Gimatic MPCF gripper serial protocol.pdf” document included in the folder available at <https://www.gimatic.com/>.

## 9.4. Examples of interacting without MPCFApplication.exe

Following examples aim to show the structure of CANopen messages useful to perform movements like the ones illustrated before in case of use of a generic CAN master instead of MPCFApplication.exe. With CAN messages it's possible to interact with the Gimatic middleware, and therefore with the Gripper Control and Status Registers like made by MPCFApplication.exe, or directly with the CAN objects bypassing the middleware.

**Please remember that the access to the objects is possible with or without the presence of the middleware.** However, erasing the middleware makes earning velocity in data transmission making the gripper act like a generic device adhering to the CANopen DS-402 standard. The objects that can be access by users are reported here (CiA 402 object dictionary):

<https://drives.novantamotion.com/emcl/dictionaries#Dictionaries-CiA402objectdictionary>.

The structure of **CANopen SDO frames** is shown below.

For command/query to the gripper:

ID	DLC	Payload							
		Byte 0					Byte 1-2	Byte 3	Byte 4-7
		Bits 5-7	Bit 4	Bits 2-3	Bit 1	Bit 0			
0x600 + node ID	8	CSS	0 (reserved)	n	e	s	Index	Sub-index	Data

Where:

- **ID** is the CAN identifier (hexadecimal format) of the message. At the value 0x600 must be added the Node ID of the gripper, for example: 0x600 + 0x20 = 0x620.
- **DLC** specifies the Data Length Code of the CAN message, which is the number of data bytes of the message (0 to 8).
- **CSS** is the Client Command Specifier; it is 1 for download (command the gripper) and 2 for upload (query to the gripper).
- **n** is the Number of bytes in the data part which do not contain data (valid if **e** & **s** are set).
- If set, **e** indicates an "expedited transfer" (all data is in a single CAN frame).
- If set, **s** indicates that data size is shown in **n**.
- **Index** is the object dictionary index of the data to be accessed (**represented in little-endian format**).
- **Sub-index** is the subindex of the object dictionary variable.
- **Data** contains the data to transfer to the gripper (**represented in little-endian format**).

The response from the gripper is as below:

ID	DLC	Payload			
		Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x580 + node ID	8	CMD	Index	Sub-index	Data

Where:

- **ID** is the CAN identifier (hexadecimal format) of the message. At the value 0x580 must be added the Node ID of the gripper, for example: 0x580 + 0x20 = 0x5A0.
- **DLC** specifies the Data Length Code of the CAN message, which is the number of data bytes of the message (0 to 8).
- **CMD** can be equal to: 0x4F (for 1 byte data length response), 0x4B (for 2 bytes data length response), 0x47 (for 3 bytes data length response) or 0x43 (for 4 bytes data length response) in case of successful upload of data required by the master; 0x60 in case of successful download required by the master; 0x80 in case of abort due to invalid command or specific error (in this case the error code is reported in data field).
- **Index** is the object dictionary index of the data to be accessed.
- **Sub-index** is the subindex of the object dictionary variable.
- **Data** contains the data to be uploaded.

For more details, please consult the relative Ingenia documentation at <https://drives.novantamotion.com/emcl/canopen-protocol> and the CANopen standard.

### 9.4.1. Examples 7 - Communication with a generic CAN master: make a grip with mode 200 and mode 220

This example aims to show two different things:

- Specific CAN SDO messages' structure.
- Difference between an operating mode with velocity and force remodulation (in this case mode 220) and its corresponding standard operating mode (in this case mode 200).

#### **Mode 200, P2P FORCE B-L RCP:**

First, one has to send a message to set the desired operating mode (200) and the recipe (for instance recipe number 004). For convenience, in a single message we also set the gripper in enabled mode. As usual, with the middleware the object useful to command the gripper is the Gripper Control Register, index 0x2C00, sub-index 0x03.

Message to send has ID equals to 0x620 (0x600 + 0x20) and is composed as follow:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	00	2C	03	01	04	00	32

Where:

- Type of command is equal to 0x22 because it is performed a download command in expedited transfer.
- Data bytes are correctly found setting the bits of the Gripper Control Register (please refer to Table 4). Remember also that index e data bytes are in little-endian format.

The response by the gripper will start with byte 0 equals to 0x60 indicating the successful response to command. The data bytes are all set to zero since no data was requested.

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
5A0	8	60	00	2C	03	00	00	00	00

Now the gripper is enabled and ready to start the movement. It is sufficient to activate the relative bit in the Control Register to move the jaws. The message is so composed:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	00	2C	03	03	04	00	32

The gripper has performed the desired movement with all parameters taken from recipe 004, where Drive Velocity is equal to 200 mm/s (640000 inc/s) and Grip Force equal to 100% of rated force. Even in this case, the gripper will respond with a successful message like made before.

The gripper will start the movement and, if a piece is encountered, will perform a grip operation. To check if the gripping force is equal to the one set in the recipe, one has to read the relative object (index 0x6077, sub-index 0x00). The corresponding message is:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	40	77	60	00	00	00	00	00

**Note the type of command byte value (0x40) and the data bytes (set to zero) given that this message is a query.**

In this case, the response by the gripper is different because it has to pass the requested value:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
5A0	8	4B	77	60	00	E8	03	00	00

Byte 0 indicates that the data length is 2 bytes. From the data values received is possible to extrapolate the actual torque value: 0x03E8, equals to 1000 in dec. Therefore, the result is 1000‰ of the rated torque, i.e. 100% of rated torque: the gripper is applying the force as reported in the recipe.

**Mode 220, P2P FORCE B-L REG FV:**

Let's see how to perform the same movement but with a modulation of velocity to 25 mm/s (80000 inc/s) and of the gripping force to 50% calling the on-fly parameters (i.e. changing their value in real time).

As made before, one has to set the operating mode (in this case 220), recipe number (004) and enabling the gripper.

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	00	2C	03	01	04	00	37

The new velocity and gripping force must be set by referring to Table 6.

**Please beware about unit measurement.**

The on-fly Drive Velocity value as index 0x2C00, sub-index 0x30 and value 0x013880:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	00	2C	30	80	38	01	00

The on-fly Grip Force value has instead index 0x2C00, sub-index 0x31 and value 0x32.

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	00	2C	31	32	00	00	00

In both cases gripper responds confirming the successful result of download operation.

Now the gripper is enabled and ready to start the movement. As made before, it is sufficient to activate the relative bit in the Control Register to move the jaws:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	00	2C	03	03	04	00	37



Even in this case, the gripper will respond with a successful message like made before and it will apparently perform the same movement, but this time the movement velocity should be decreased a lot and the gripping force should be halved. Let's verify it reading the actual torque value:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	40	77	60	00	00	00	00	00

Gripper response:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
5A0	8	4B	77	60	00	FA	01	00	00

It's easy to extrapolate the actual torque value: 0x01FA, equals to 506 in dec. Therefore, the result is 506‰ of the rated torque, i.e. about 50% of rated torque: the gripper is applying the force set on-fly and not the one reported in the recipe.

Since we have modified velocity and force by on-fly, if now one would recharge recipe number 4 and execute a movement, this would be carried out with the original values.

### 9.4.2. Examples 8 - Communication with a generic CAN master: interacting directly with CAN objects bypassing middleware

On follow there are reported some operations as examples of interacting directly with the CAN objects bypassing the Gimatic middleware.

#### 0x6040 – Controlword, set the gripper in Switch on mode:

It's important to send this command every time the gripper is turned on to correctly communicate with it.  
Message sent to the gripper:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	40	60	00	07	00	00	00

#### 0x6040 – Controlword, enable operation (enabling the gripper):

Message sent to the gripper:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	40	60	00	0F	00	00	00

#### 0x6064 – Read position actual value (value returned in increments):

Message sent to the gripper:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	40	64	60	00	00	00	00	00

Response from the gripper:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub-index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
5A0	8	43	64	60	00	88	88	00	00

The value received is: 0x8888, equals to 34952 inc that means  $34952/3200 = 10.92$  mm.

**0x606C – Read velocity actual value (value returned in increments/second):**

Message sent to the gripper:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	40	6C	60	00	00	00	00	00

Response from the gripper:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
5A0	8	43	6C	60	00	90	91	01	00

The value received is: 0x019190, equals to 102800 inc/s that means  $34952/3200 = 32.13$  mm/s.

**0x60FF – Store target velocity (value in increments/second):**

Desired value: 50 mm/s, equals to 160000 inc/s and therefore 0x027100 (hex).

Message Sent to the gripper:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	FF	60	00	00	71	02	00

**0x60FF – Store target velocity (value in increments/second):**

Desired value: -50 mm/s, equals to -160000 inc/s and therefore 0x018F00 (hex, 2's complement).

Message Sent to the gripper:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	FF	60	00	00	8F	01	00

Please note the 2's complement for the negative value to be write.

**0x6060 – Set the modes of operation:**

Desired mode: profile position, value: 0x01.

Message Sent to the gripper:

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	60	60	00	01	00	00	00

**Movement example (gripping with profile velocity):**

In this example, it will be showed how to perform a gripping action using the profile velocity mode.

Every operation mode is explained in detail in the online official documentation (for instance: <https://drives.novantamotion.com/emcl/profile-velocity-mode>). Profile velocity means that the jaws start moving in a direction specified by the sign of the target velocity value and, once encountered an obstacle/object to grip, they will keep the maximum torque set. Since it is a profiled movement, the dynamics' parameters used are profile acceleration and profile deceleration.

The steps to perform are the follow:

- Switch on the gripper

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	40	60	00	07	00	00	00

- Set the operation mode to profile velocity. Object with index 0x6060, sub-index 0x00, value: 0x03.

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	60	60	00	03	00	00	00

- Set maximum torque to 700% of rated torque. Object with index 0x6072, sub-index 0x00, value: 0x02BC.

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	72	60	00	BC	02	00	00

- Set other dynamic's parameters (both profile acceleration and profile deceleration to 600 mm/s<sup>2</sup>, equals to 1920000 inc/s<sup>2</sup>). Object 0x6083, sub-index 0x00, value: 0x1D4C00.

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	83	60	00	1D	4C	00	00

Object 0x6084, sub-index 0x00, value: 0x1D4C00.

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	84	60	00	1D	4C	00	00

- Set target velocity to 100 mm/s (equals to 320000 inc/s). Object 0x60FF, sub-index 0x00, value: 0x04E200. **The sign determines the movement direction.**

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	FF	60	00	00	E2	04	00

- Enable the gripper trough the controlword. Object 0x6040, sub-index 0x00, value: 0x0F.

ID	DLC	Byte 0 (type of command)	Byte 1 (index LSB)	Byte 2 (index MSB)	Byte 3 (sub- index)	Byte 4 (data LSB)	Byte 5	Byte 6	Byte 7 (data MSB)
620	8	22	40	60	00	0F	00	00	00

The gripper starts moving at the desired velocity and if it encounters a piece (or reaches the mechanical limit in case of missing part), it will hold with the max torque set.

The actual torque value can be verified reading the relative object as made before.





GIMATIC S.R.L.  
Via Enzo Ferrari 2/4 25030 Roncadelle (Bs) Italy  
Tel: +39 030 2584655 Fax: +39 030 2583886  
[info@gimatic.com](mailto:info@gimatic.com)  
[www.gimatic.com](http://www.gimatic.com)