APICOO ROBOTICS

SusGrip Smart Gripper

USER MANUAL

March 2025

apicoorobotics.com

This page was intentionally left blank.

REVISIONS

Revision	Date	Changelog
rev_2f.1.1	17 Aug 2024	First release
rev_2f.2.1	21 Oct 2024	- Added ModbusRTU registers information; - Added examples for ModbusRTU interfacing;
rev_2f.3.1	01 Jan 2025	- Added Universal Robotics compatibility information; - Added detailed information for each register; - Adjusted graphics for Electrical Setup; - Added manual for URCap installation and programming;
rev_2f.3.2	12 Jan 2025	- Added ModbusRTU framing guide; - Added ModbusRTU register mapping; - Adjusted graphics for Product Ratings;
rev_2f.4.1	20 Feb 2025	- Changed UR compatibility information: UR16 non “e” removed from compatibility table; - Changed GPIO type: GPIO input pin 4 supports native PNP Outputs; - Added remark: PNP-to-NPN adapter for pin 4 supported. - Changed GPIO logic: GPIO input pin 4 is now pulled down by default, and activates when HIGH (Previously: Pulled up by default, activated LOW); - Improved typos, grammatical errors, general diction;
rev_2f.4.2	27 Feb 2025	- Added section name to footer of Revision and Content; - Improve clarity in several sections;
rev_2f.4.3	04 Mar 2025	- Improved typos, grammatical errors, general diction;

This page was intentionally left blank.

CONTENT

This page was intentionally left blank.

1 - INTRODUCTION

SusGrip is a smart gripper with sophisticated mechanical design that enables precise parallel gripper motion of the end-effector's fingers. The gripper is powered by one high performance BLDC motor, allowing efficient yet robust operation for a wide range of applications.

SusGrip gripper is equipped with a high precision absolute encoder for accurate position monitoring of fingers position, eliminating the need to perform pre-operation calibration precedures, thereby enhancing user's convenience and efficiency. SusGrip also offers a handful of smart features such as Object Detection and Anti Back-driving. For manual gripper adjustment during power-down, users may refers to the dedicated section which will be eleborated later in this manual.

The modular design of the SusGrip fingers allows users to easily swap and exchange different finger types, each secured with M5x08 bolts for base attachment. This flexibility enables the gripper to be tailored to specific application requirements.

SusGrip gripper adopts a human-friendly design with numerous safety features. However, Apicoo strongly recommends that users perform careful risk assessments before deploying the product into real operation.

2 - PRODUCT SPECIFICATIONS

2.1 - Mechanical dimensions

The figures below demonstrate the mechanical dimensions of the SusGrip gripper, each dimension is given in milimeters (mm).

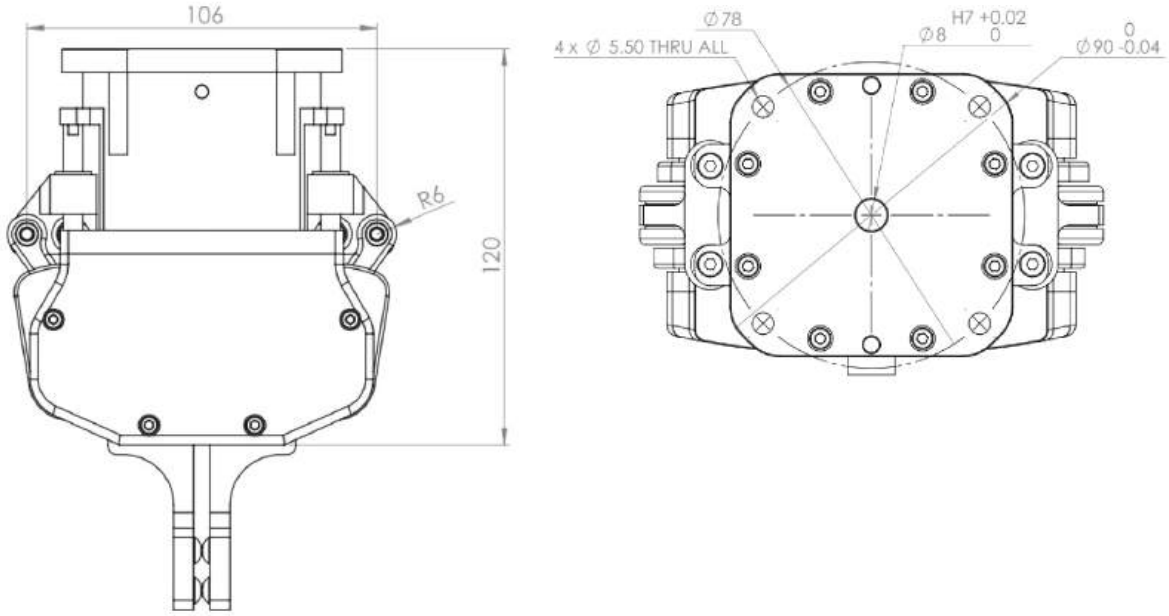


Figure 2.1 - General dimensions of the gripper:
Side view (left) and Top view (right)

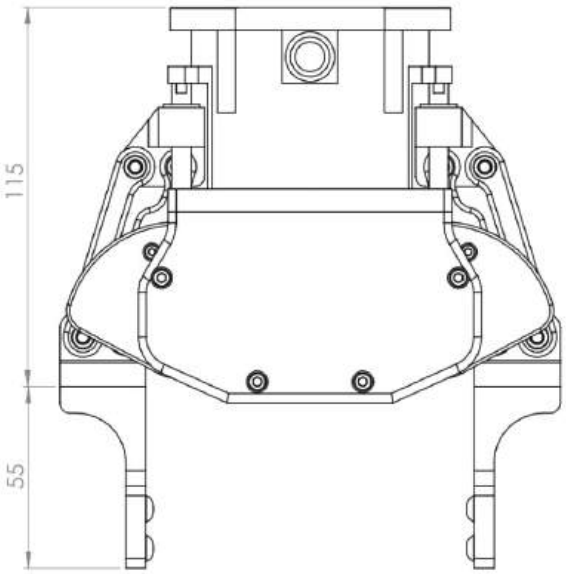
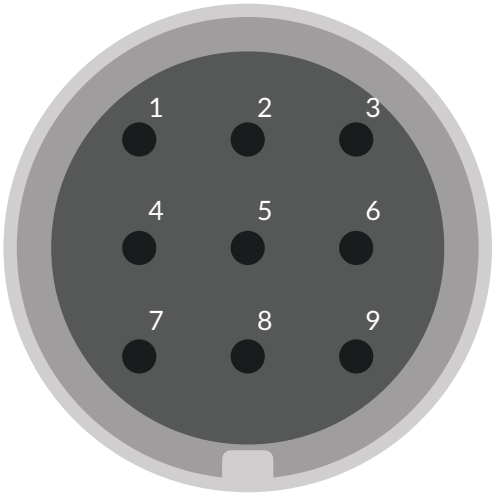


Figure 2.2 - Gripper finger dimensions

2.2 - Communication port

SusGrip takes 24VDC for input power supply. The power rail and communication protocol are integrated into one single port for ease of connectivity. The following figures show the pin mapping for the port and the given cables, as well as a picture of the port:



Pin No.	Cable color	Description
1	RED	+24V input
2		None
3	YELLOW	RS485+
4	BLUE	Gripper input 1
5	BROWN	Gripper output 1
6	WHITE	GND
7	GREY	Gripper output 2
8	GREEN	RS485-
9	BLACK	GND

Figure 2.3 - Communication port pin mapping

CAUTION:
The cables color and pin mapping in this figure are given only for general applications. For URCap end-tool version, please navigate to Chapter 5.

Figure 2.4 - Communication port on the SusGrip gripper

ATTENTION:
The pin mapping illustration aligns with the gripper's fingers pointing upward.

2.3 - Product ratings

The absolute ratings of SusGrip are listed in the table below:

Table 2.1 - SusGrip absolute ratings

ENTRY	VALUE	UNIT
Grasp force	100	N
Stroke	0-128	mm
Form-fit payload	05	kg
Voltage	24	V
Max current	1.5	A
Repetition accuracy	0.1	mm
Position resolution	0.5	mm
Speed	120 (All fingers)	mm/s
Weight	1250	gr
Working temperature	-5 ~ 55	°C

2.4 - Universal Robots compatibility

SusGrip is compatible with most Universal Robots' cobots and is a part of the Polyscope ecosystem. SusGrip enables maximum ease of integration into already existing systems that utilize Universal Robots' arms by providing intuitive URCap versions and simple cable connectivity. To integrate the gripper into a UR system, users can either connect SusGrip directly to the end-tool port or connect to UR robots' controller box. The table below gives information about the URCap end-tool compatibility on UR robots:

Table 2.2 - Universal Robots URCap end-tool compatibility

IS COMPATIBLE WITH:	END-TOOL	CONTROLLER BOX
UR3	No (low amps)	Yes
UR3e	No (low amps)	Yes
UR5	No (low amps)	Yes
UR5e	Yes	Yes
UR10	Yes	Yes
UR10e	Yes	Yes
UR16e	Yes	Yes
UR20	Yes	Yes
UR30	Yes	Yes

3 - INSTALLATION GUIDE

3.1 - Mechanical setup

SusGrip can be installed onto a robot end-effector using 4 bolts of type M5x08, with 76mm PCD (Pitch Circle Diameter). The adapter flange provides flexible installation onto multiple models of industrial robots. The figure below demonstrates the setup for the gripper:

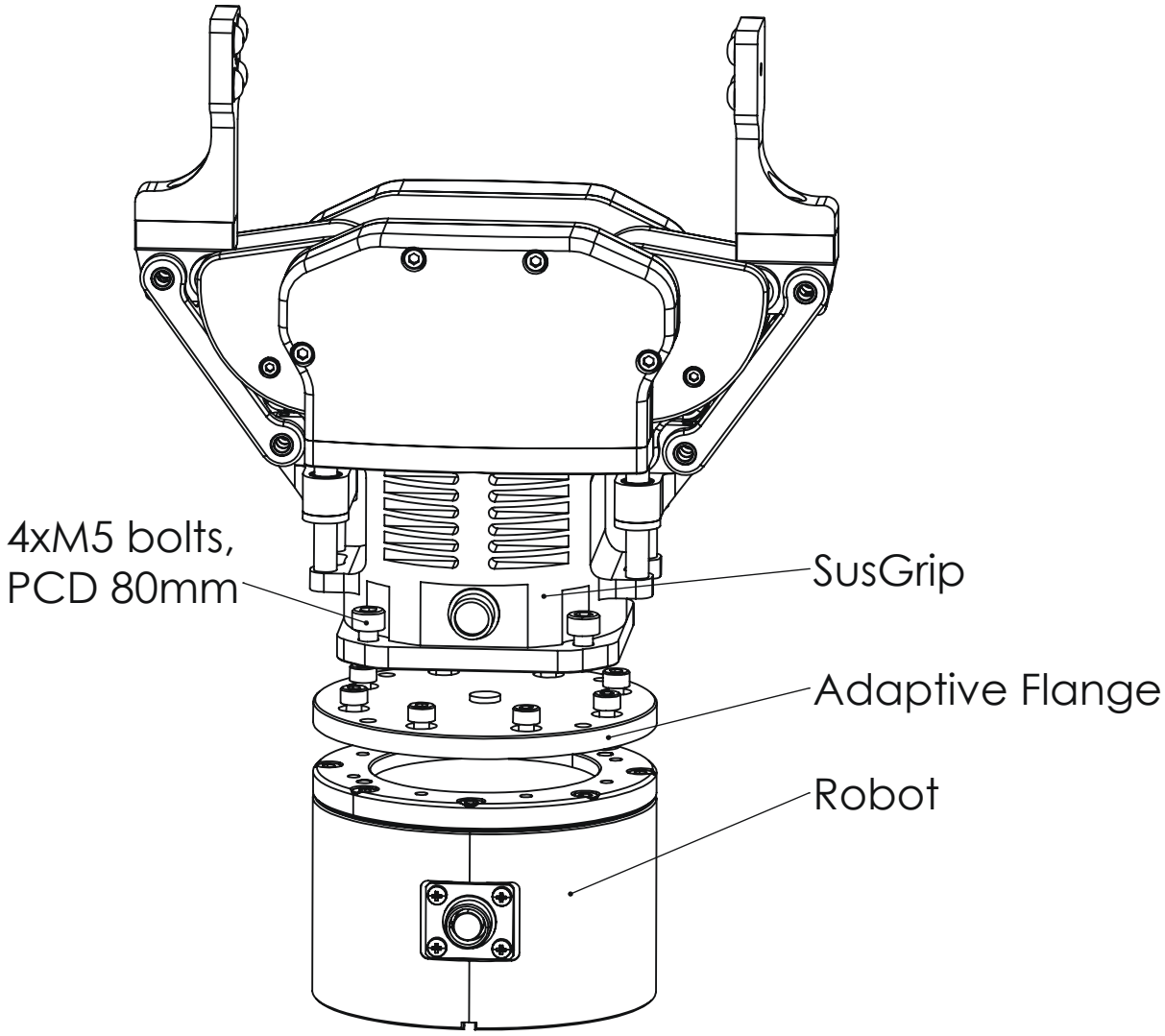


Figure 3.1 - Installing SusGrip gripper onto robotic arm

3.2 - Electrical setup

SusGrip gripper offers an easy-to-use control method using GPIO pins, along side with the RS485 Modbus RTU communication protocol. For ease of demonstration and setup, a USB-to-RS485 converter device can be used to quickly connect the gripper to a computer.

The following figure demonstrates the wiring connections between the gripper and the control box of robots, or a computer using USB-to-RS485 device:

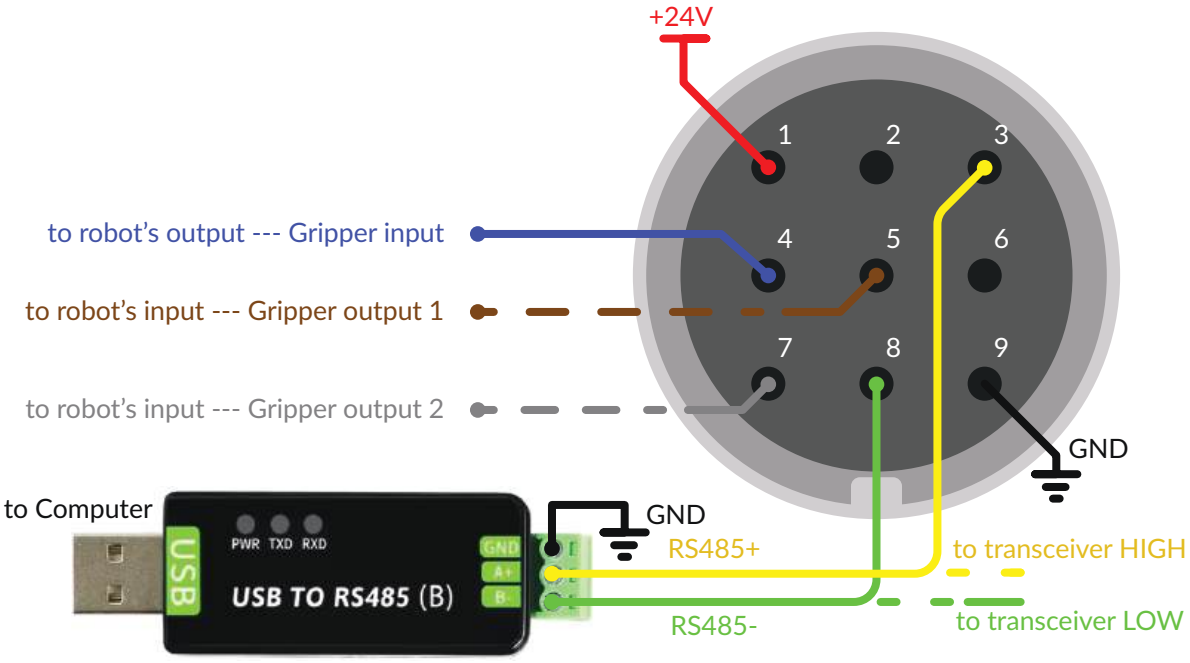


Figure 3.2 - SusGrip gripper connections to robot and computer

In the figure above:

Pin 1 is connected to 24VDC power supply,
Pin 2 is left unconnected,
Pin 3 is connected to terminal A+ on the USB-to-RS485 device,
or Positive terminal of the RS485 bus,
Pin 4 is connected to any 24V PNP digital output on the control box,
Pin 5 is connected to any 24V digital inputs on the control box,
Pin 6 is spare and can be used for GND reference,
Pin 7 is connected to any 24V digital inputs on the control box,
Pin 8 is connected to terminal B- on the USB-to-RS485 device,
or Negative terminal of the RS485 bus,
Pin 9 is connected to common GND reference.

ATTENTION:
While Pin 4 can be connected to any DO, and Pin 5 and Pin 7 to any 24V DI as desired, please make sure to configure the pinout appropriately for correct operation.

ATTENTION:
The Input Pin 4 supports native PNP Digital Output.
For NPN Digital Output, Apicoo does provide adapted cable specific for this type.
Please contact Apicoo Robotics customers support for further assistance.

3.3 - Performing preliminary tests on the SusGrip gripper

To perform tests on the gripper, users can either do wires shorting directly or send commands through the SusGrip GUI application which will be elaborated in the latter part of this section.

3.3.1 - Testing the gripper by hardware

To quickly test the gripper by hardware, first connect Pin 1 and Pin 9 with 24VDC power supply and GND reference, respectively. Upon receiving power, the gripper will close its fingers:

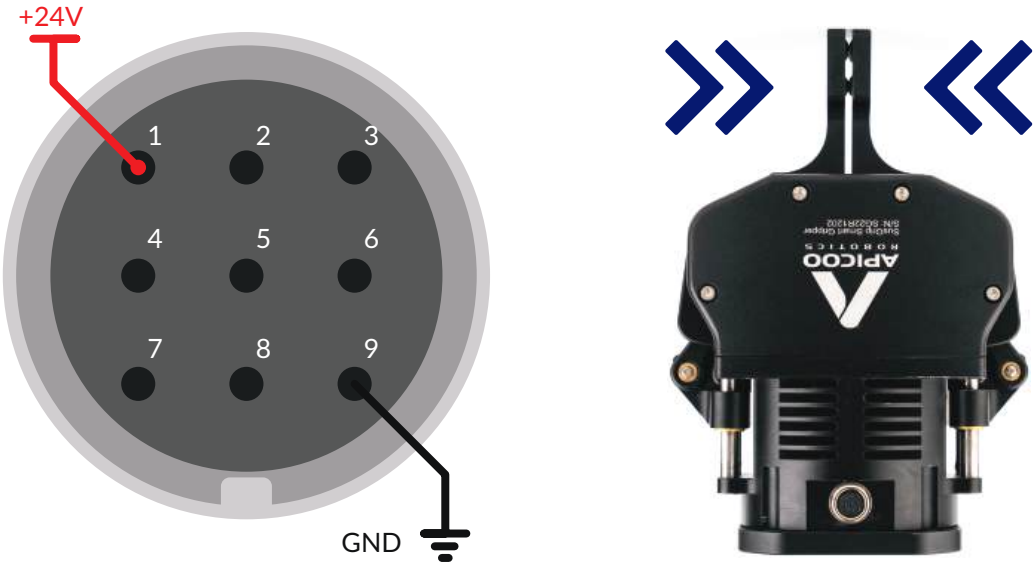


Figure 3.3 - Gripper closes its fingers when Pin 4 left open (Pulled down internally)

Next, pull the input pin of the gripper to HIGH by shorting Pin 4, or the blue wire, to pin 1 or directly to 24VDC. If by doing so, the gripper opens its fingers, then the functionality of the gripper can be verified as OK:

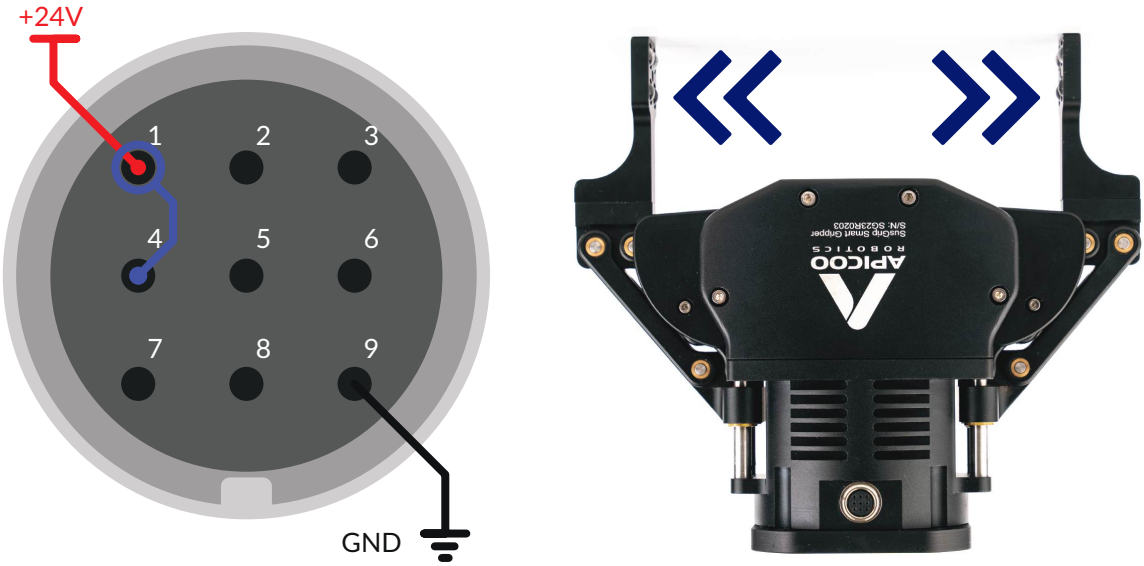


Figure 3.4 - Gripper opens its fingers when Pin 4 connected to 24VDC

3.3.2 - SusGrip GUI software installation and setup

To ensure easy setup precedure for users, Apicoo offers the SusGrip Graphical User Interface (GUI) application, which can be installed on either Microsoft Windows systems or Linux-based systems. The installation media can be found on Apicoo Robotics official website, or by inquiry via direct contact with Apicoo Robotics customers support.

To start the application, first extract the application zip file. Then in the extracted package, run the *main* executable file:

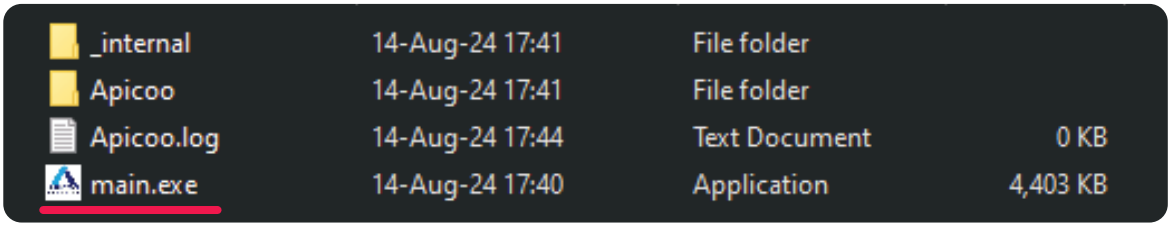


Figure 3.5 - SusGrip GUI application files

The users should be greeted with the application screen as seen in the figure below:

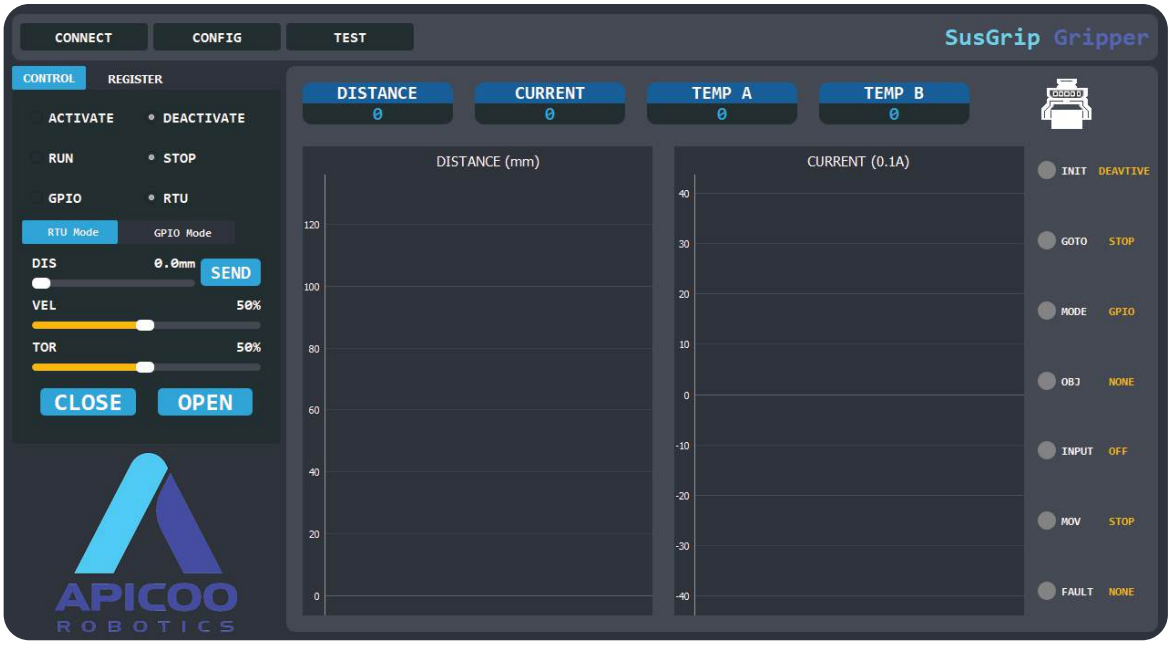


Figure 3.6 - SusGrip GUI application

ATTENTION:
If the application cannot execute, please run with Administrator's or Super User's priviledge!
If the problem cannot be resolved, please contact Apicoo Robotics customers support.

3.3.3- Testing the gripper with SusGrip GUI

Using the USB-TO-RS485 device provided in the product package, connect gripper Pin 3 and Pin 8 to device terminals A+ and B-, respectively, as well as common GND reference, as depicted in section 3.2, figure 3.2.

Then, connect the USB-TO-RS485 device to any USB port available on a computer. Upon successful connection, a blue LED on the gripper body turns on then remains idle blue:

Figure 3.7 - The blue LED remains idle

On the SusGrip GUI application, click on “CONFIG”, enter the correct ID (defaults to 86) and select the appropriate COM port:

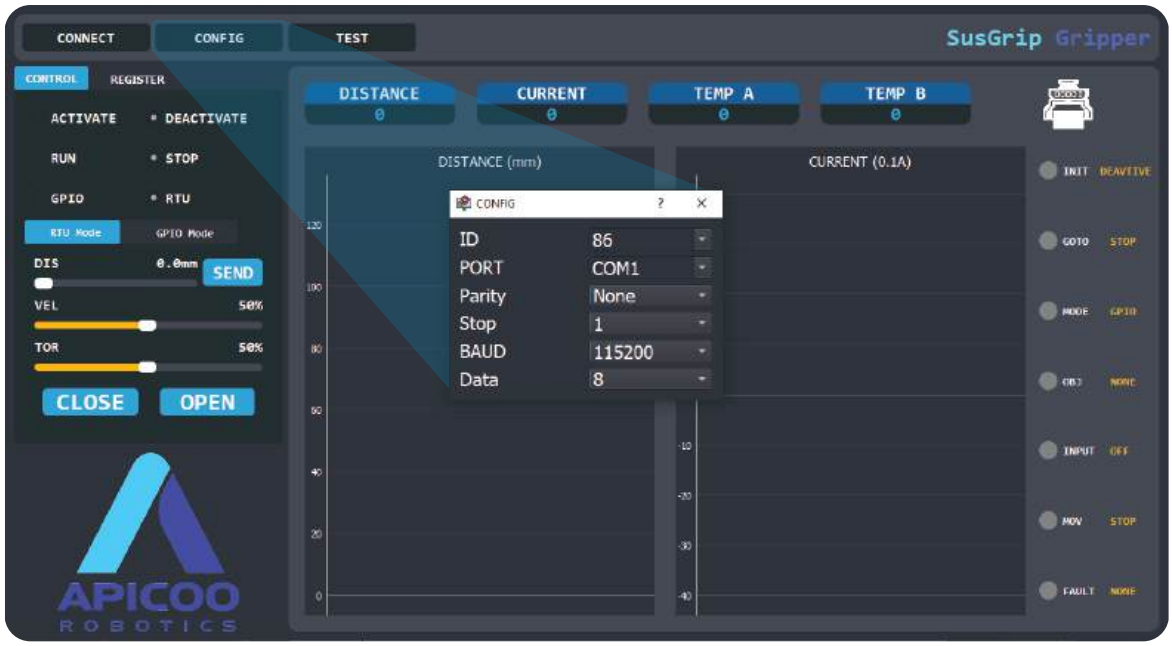


Figure 3.8 - On the SusGrip GUI application, click on “CONFIG”

Then, click on “CONNECT”:

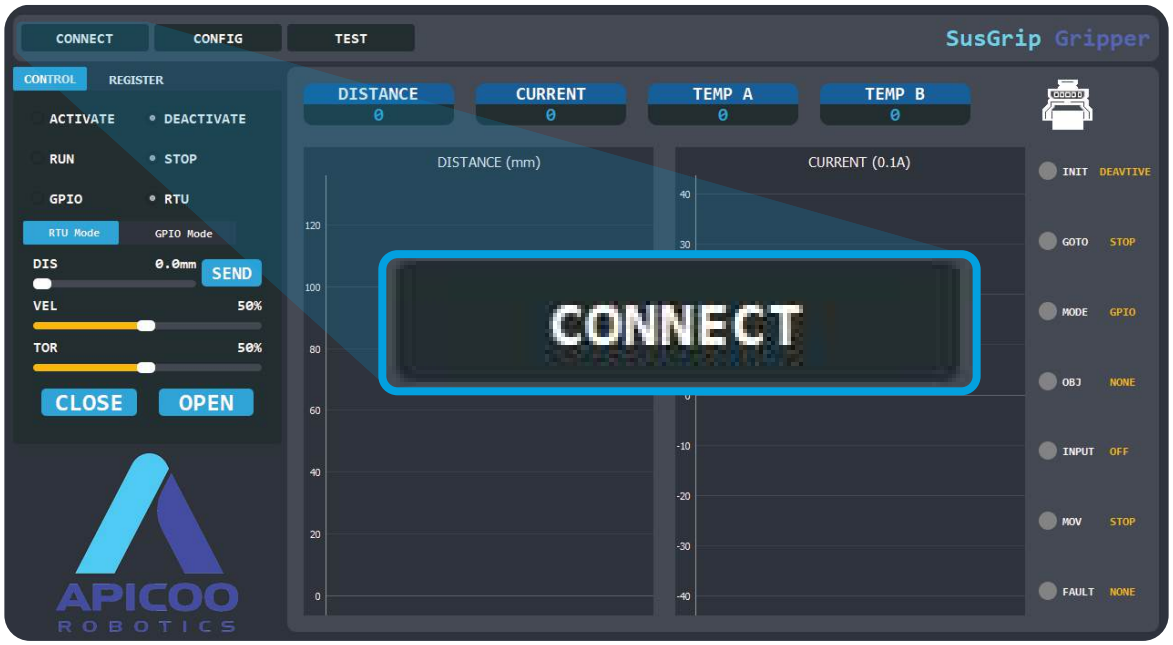


Figure 3.9 - Click on “CONNECT” to start controlling the gripper

Once succeeded, with the sliders on the left-hand side tab, users can send commands to manipulate the gripper position (distance between 2 fingers), velocity and torque.

The SusGrip GUI application also provides graphs and data for monitoring the gripper fingers distance, as well as temperature and current consumption:

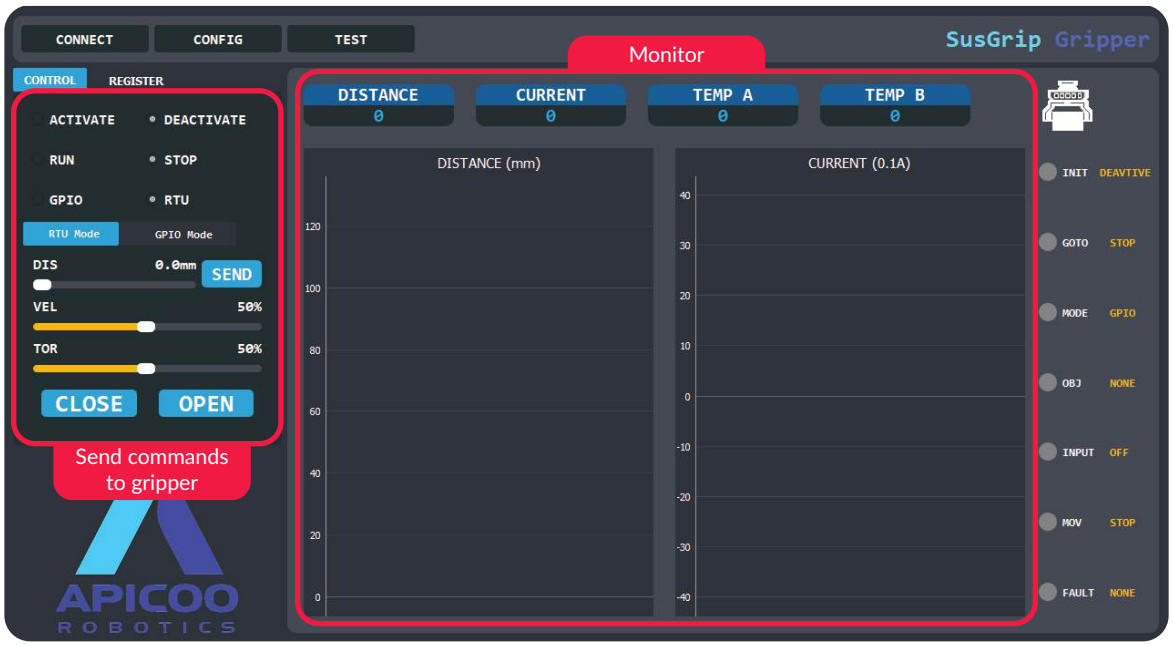


Figure 3.10 - Controlling and monitoring panels

ATTENTION:
After adjusting the sliders, be sure to click on
the “SEND” button to confirm command and send to the gripper.

4 - OPERATION GUIDE

SusGrip gripper offers in total 2 modes of control to users: GPIO mode and Modbus mode. Upon powering up, the gripper defaults to GPIO control mode for ease of operation. In this section, Apicoo provides users with a guide on operating the SusGrip gripper with both modes.

4.1 - Operating SusGrip gripper in GPIO mode

The GPIO mode is the default control mode of the SusGrip gripper. In this mode, the GPIO Pin 4 is used to control the grasping and releasing behavior of the gripper.

As described in section 3.3.1, Pin 4 is pulled down to LOW logic level by default. The gripper closes its fingers when Pin 4 is written LOW, and opens its fingers when Pin 4 is written HIGH.

The distance at which the gripper closes or opens its fingers in this mode is determined by preset parameters. The default parameters are:

Open Distance: **120mm**

Close Distance: **10mm**

To adjust these preset parameters, users can use the SusGrip GUI application. In the SusGrip GUI app, navigate to the “REGISTER” tab next to the “CONTROL” tab on the upper left-hand side panel:

Figure 3.11 - Register panel in SusGrip GUI

ATTENTION:
Detailed explanations on manipulating registers will be elaborated in the latter section. For now, Apicoo suggests that users follow this guide as is.

The “REGISTER” panel has 6 rows and 2 columns. The “COMMAND” column that contains a series of 200x registers acts as control registers for sending commands to the gripper, and the “STATUS” column that contains a series of 100x registers acts as status registers, as its name implies, for monitoring the gripper status.

In GPIO mode the registers values are:

[2000]: 7 (Mode of operation - **STRICTLY**)
[2001]: Don’t care
[2002]: 1 - 255 (% max velocity)
[2003]: 1-255 (% max torque)
[2004]: 0-255 (Close distance, divided by 2, constrained 0 to 132mm)
[2005]: 0-255 (Open distance, divided by 2, constrained 0 to 132mm)

For example, to change the behavior of the gripper in GPIO mode as close fingers to 20mm, open fingers to 100mm, velocity to 50%, and torque to 50%:

[2000]: 7 (Mode of operation - **STRICTLY**)
[2001]: _
[2002]: 127 (50% max velocity)
[2003]: 127 (50% max torque)
[2004]: 40 (Close distance at 20mm)
[2005]: 200 (Open distance at 100mm)

Finally, click on “SAVE CONFIG” to confirm and apply changes to gripper:.

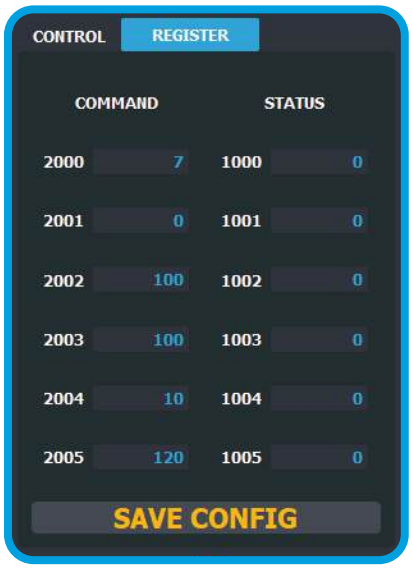


Figure 3.12 - Click on SAVE CONFIG to apply changes to preset parameters

ATTENTION:
Register [2000] value MUST be strictly 7 in GPIO mode and 3 in Modbus mode.
Velocity and Torque values of register [2002] and [2003] are set as percentage of max value.

4.2 - Operating SusGrip gripper in ModbusRTU mode

This section provides detailed elaboration of the gripper's Modbus registers, and is dedicated to the assistance in operating the gripper in RTU mode using the **REGISTER** tab in the SusGrip GUI application. To learn about manually sending RTU commands without the application, please navigate to section 4.3.

The Modbus RTU protocol of the SusGrip gripper has the following configuration:

Baudrate:	**115200**
Data bit/Parity/Stop bit:	**8/N/1**
Modbus functions:	**FC04, FC06, FC16** (FC stands for Function Code)
Slave ID:	**86**

As seen in figure 3.12 and figure 3.13, there are 2 sets of registers, the 200x series are control registers and the 100x series are status registers. These are all 16-bit registers, and can be manipulated bitwise. Below, the detailed information about the 200x registers will be elaborated, then the 100x ones.

Register 2000 controls the gripper status and mode of operation:

15 - 3	2	1	0
x	MODE	GTO	ACT

Bit 0 - ACT activates the gripper, must be 1 for other bits to be valid:
- 0: Deactivate gripper
- 1: Activate gripper

Bit 1 - GTO enables gripper fingers movement:
- 0: Disable fingers movement, fixed in place
- 1: Enable fingers movement

Bit 2 - MODE decides the operation mode for the gripper:
- 0: Modbus RTU mode with RS485
- 1: GPIO mode

Bit 3-15 are reserved, DON'T CARE.

Register 2001 sets the target position (distance between fingers) of the gripper:

15 - 0
Value / 2 = Target position in mm

Bit 0-15 set the values of distance in milimeters, divided by 2,
the actual setable value is constrained from 0 to 132 mm.

Registers 2002 and 2003 set the fingers speed and torque, respectively:

15 - 8	7 - 0
x	Value / 2.55 = Percentage speed/torque

Bit 0-7 take the values of speed in or current amplitude in % max value
from 0-255, for example: 0000 0000b = 0%, 0111 1111b = 50%
Bit 8-15 are reserved, DON'T CARE.

Registers 2004 and 2005 set the preset values for closing distance and opening distance in GPIO mode, respectively:

15 - 0
Value / 2 = Preset opening/closing distance in mm

Bit 0-15 take the values of distance in milimeters,
the actual setable value is constrained from 0 to 132 mm

Register 1000 shows the gripper status:

15 - 8	7	6	5	4	3	2	1	0
gFAULT	gMOVE	gIN	gOBJ		gMODE		gGTO	gACT

Bit 0 - gACT holds the activation status of the gripper:
- 0: Gripper deactivated
- 1: Gripper activated

Bit 1 - gGTO holds the information on whether fingers movement is allowed:
- 0: Fingers movement is constrained, locked in place
- 1: Fingers movement is allowed

Bit 2-3 - gMODE hold the information on the operation mode of the gripper:
- 00: Modbus RTU mode with RS485
- 01: GPIO mode

Bit 4-5 - gOBJ hold the information on whether the gripper detected an object on its trajectory:
- 00: No object detected
- 01: Object detected while opening fingers
- 10: Object detected while closing fingers
- 11: Object dropped, fell out of grasp

Bit 6 - gIN holds the status of gripper input pin (Pin 4) accordingly

Bit 7 - gMOVE holds the information on whether the fingers are moving:
- 0: Fingers are stationary
- 1: Fingers are on their way moving

Bit 8-15 - gFAULT hold the information about error codes during operation process of the gripper:
- 0000 0000: No error
- 0000 0001: Motor driver current limit exceeded (DRV OCTW)
- 0000 0010: Motor driver error (DRV FAULT)
- 0000 0011: Gripper over-current
- 0000 0100: Gripper overheated
- 0000 0101: Target object out of gripper's grasp range

Registers 1001 echoes the target position set by users to register 2001:

15 - 0
Value / 2 = Target position set by user in mm

Registers 1002 shows the actual position read by the encoder:

15 - 0
Value / 2 = Actual position in mm

Register 1003 shows the current magnitude of the motor:

15 - 0
Value / 10 = Current magnitude in Amps

Register 1004 shows the temperature of the gripper read by 2 thermometers:

15 - 8	7 - 0
Temperature A	Temperature B

Register 1005 is reserved.

Using this guideline, users can set the desired values for each parameter via the provided SusGrip GUI application. In the next page, some examples are provided on how to set values to the registers using this scheme.

If users wish to implement their own Modbus RTU client communicating with SusGrip RTU server without using SusGrip GUI application, the next section provides the guideline for RTU framing in communication with the gripper.

To operate SusGrip in Modbus RTU mode using the provided SusGrip GUI application, navigate to the “REGISTER” tab next to the “CONTROL” tab on the upper left-hand side panel:

Figure 3.11 (Repeated) - Register panel in SusGrip GUI

In this tab, users can input directly values for the registers elaborated previously to manipulate the gripper to their desire. Below are some examples on how to calculate the values according to specific requirements and desire:

Example: Manipulating registers in Modbus RTU mode for: finger distance 120mm, speed 100%, torque (current) 50%:

- RTU mode -> [2000]: 0000 0011 = (dec) 3
- Position 120mm -> [2001]: 1111 0000 = (dec) 240 = 2x120
- Speed 100% -> [2002]: 1111 1111 = (dec) 255 = 100%x255
- Torque 50% -> [2003]: 0111 1111 = (dec) 127 = 50%x255

Example: Manipulating registers in Modbus RTU mode for: finger distance 16mm, speed 20%, torque (current) 80%:

- RTU mode -> [2000]: 0000 0011 = (dec) 3
- Position 16mm -> [2001]: 0010 0000 = (dec) 32 = 2x16
- Speed 20% -> [2002]: 0011 0011 = (dec) 51 = 20%x255
- Torque 80% -> [2003]: 1100 1100 = (dec) 204 = 80%x255

Example: Manipulating registers in GPIO mode for: speed 20%, torque (current) 50%, closing distance 20mm, opening distance 80mm:

- GPIO mode -> [2000]: 0000 0111 = (dec) 7
- Speed 20% -> [2002]: 0011 0011 = (dec) 51 = 20%x255
- Torque 50% -> [2003]: 0111 1111 = (dec) 127 = 50%x255
- Close 20mm -> [2004]: 0010 1000 = (dec) 40 = 2x20
- Open 80mm -> [2005]: 1010 0000 = (dec) 80 = 2x40

4.3 - Modbus RTU framing guide

The previous sector provided detailed information about the gripper's Modbus registers and quick examples on how to calculate register values. This sector provides a guide on manually sending request to SusGrip RTU server via Modbus RTU.

(Repeated) The Modbus RTU protocol of the SusGrip gripper is configured as follow:

Baudrate: **115200**
Data bit/Parity/Stop bit: **8/N/1**
Modbus functions: **FC04, FC06, FC16** (FC stands for Function Code)
Slave ID: **86**

To communicate with SusGrip gripper RTU server, users need to send RS485 signals to the corresponding pin 3 and pin 8, as specified in section 3.2. To send commands or requests to SusGrip RTU server, the serial package should align with the following frame:

[ServAddr]_[FuncCode]_[Reg][Reg]_[Data][Data]_[CRC][CRC] (8 Bytes);

The gripper's RTU server will repond with the corresponding frame:

[ServAddr]_[FuncCode]_[Reg][Reg]_[Data][Data]_[CRC][CRC] (8 Bytes);

or when requested to read holding registers:

[ServAddr]_[FuncCode]_[BytesNum]_[Data][Data]_[CRC][CRC] (7 Bytes).

4.3.1 - Registers mapping

Function	Address	Description	Read/Write	Range	Unit
Config	07D0h	Enable, disable or choose mode of operation	W		
Set Position	07D1h	Set target position to the gripper	W	0000h - 0100h	mm
Set Velocity	07D2h	Set target velocity to the gripper	W	0000h - 00FFh	%
Set Torque	07D3h	Set target torque to the gripper	W	0000h - 00FFh	%
Set GPIO Close Position	07D4h	Set closing position for GPIO mode	W	0000h - 0100h	mm
Set GPIO Open Position	07D5h	Set opening position for GPIO mode	W	0000h - 0100h	mm
Status	03E8h	Get the instantaneous status of the gripper	R	0000h - 0100h	mm
Target Position	03E9h	Get the target position value set by the user	R	0000h - 0100h	mm
Actual Position	03EAh	Get the actual position read by encoder	R	0000h - 0100h	mm
Current Magnitude	03EBh	Get the current magnitude of the motor	R	0000h - 00FFh	A
Gripper Temperature	03ECh	Get the temperature of the motor	R	0000h - FFFFh	C

4.3.2 - Configuration register (0x07D0)

The **Configuration** register sets the state of the gripper, as well as selects the operation mode: GPIO or Modbus RTU. By power on, the gripper automatically defaults to GPIO mode, activates and enables movement.

- Address: 07D0h
- Write Only
- Default: 0007h

15 - 3	2	1	0
x	MODE	GTO	ACT

This register's value is bit addressable, as described in the previous section.

Example RTU frame for initializing the gripper in GPIO mode:

SEND: **56 06 07 D0 00 07 C5 62**
RESPONSE: **56 06 07 D0 00 07 C5 62**
where 56h is gripper RTU server address (86);
06h is Modbus function 06;
07 D0h is register address;
00 07h is data, refer to the previous section for detailed infomation;
62h and C5h are error checking bytes calculated by the client.

Example RTU frame for initializing gripper in ModbusRTU mode:

SEND: **56 06 07 D0 00 03 C4 A1**
RESPONSE: **56 06 07 D0 00 03 C4 A1**

4.3.3 - Position registers (0x07D1, 0x03E9, 0x03EA)

The **Set Position** register commands the target value for the gripper. Please note that the position value must be doubled before sending. For instance, to command a position of 100mm, users should set a value of 100 x 2 = 200 to this register.

- Address: 07D1h
- Write Only
- Value half-ed

15 - 0
Value / 2 = Target position in mm

Example command target position of 100mm:

SEND: **56 06 70 D1 00 C8 D4 F6** (00 C8h = 200 = 100mm x2)
RESPONSE: **56 06 70 D1 00 C8 D4 F6**

The **Target Position** register stores the value that the user set to the above register.

- Address: 03E9h
- Read Only

This register provides re-confirmation of the position setpoint, should the users need it. To read from this register, the users may send:

SEND: **56 04 03 E9 00 01 ED 9D**

then the value returned by SusGrip RTU server is:

RESPONSE: **56 04 02 xx xx CRC1 CRC2**

where <xx xx> is 2 bytes of data of the value stored in the register,
<CRC1 CRC2> is 2 bytes of CRC calculated by the server.

The **Actual Position** register stores the value of the instantaneous position of the gripper, read by gripper's encoder.

- Address: 03EAh
- Read Only

To access the content of this register, the user can send for example:

SEND: **56 04 03 EA 00 01 1D 9D**

then the value returned by SusGrip RTU server is:

RESPONSE: **56 04 02 xx xx CRC1 CRC2**

where <xx xx> is 2 bytes of data of the value stored in the register,
<CRC1 CRC2> is 2 bytes of CRC calculated by the server.

4.3.4 - Set Velocity register (0x07D2)

The **Set Velocity** register commands the maximum velocity for the gripper on its trajectory. The value is constrained from 0 to 255 (0000h - 00FFh), and is set as the percentage of maximum value.

- Address: 07D2h
- Write Only
- Set as percentage, with 0000h = 0% and 00FFh = 100%

15 - 8	7 - 0
x	Value / 2.55 = Percentage speed/torque

Example command gripper's velocity to 50%:

SEND: **56 06 07 D2 00 7F 64 80**

RESPONSE: **56 06 07 D2 00 7F 64 80**

where <00 7F> is 50% of 00FFh.

4.3.5 - Set Torque register (0x07D3)

The **Set Torque** register commands the force with which the gripper reaches its target position. The value is constrained from 0 to 255 (0000h - 00FFh), and is set as the percentage of maximum value, similar to the Set Velocity register above.

- Address: 07D3h
- Write Only
- Set as percentage, with 0000h = 0% and 00FFh = 100%

Example command gripper's torque to 50%:

SEND: **56 06 07 D3 00 7F 35 40**

RESPONSE: **56 06 07 D3 00 7F 35 40**

4.3.6 - Status register (0x03E8)

The **Status** register stores the states of the gripper, its operation mode, traveling flag, object-detection flag, as well as error and warning codes. The detailed meaning of each bit was elaborated in the previous section 4.2, register 1000.

- Address: 03E8h
- Read Only

15 - 8	7	6	5	4	3	2	1	0
gFAULT	gMOVE	gIN	gOBJ		gMODE		gGTO	gACT

To read from this register, the users may send:

SEND: **56 04 03 E8 00 01 BC 5D**

then the value returned by SusGrip RTU server is:

RESPONSE: **56 04 02 xx xx CRC1 CRC2**

where <xx xx> is 2 bytes of data of the value stored in the register,
<CRC1 CRC2> is 2 bytes of CRC calculated by the server.

4.3.7 - Current Magnitude register (0x03EB)

The **Current Magnitude** register stores the value of the strength of electric current being set to the gripper. This value is directly proportional to the strength of the torque of the gripper. To obtain the current magnitude in Amps, the value read from this register must be divided by 10.

- Address: 03EBh
- Read Only
- Value / 10 = Amperes

15 - 0
Value / 10 = Current magnitude in Amps

To read from this register, the users may send:

SEND: **56 04 03 EB 00 01 4C 5D**

then the value returned by SusGrip RTU server is:

RESPONSE: **56 04 02 00 08 CD 3A**

where <00 08> is current magnitude, here it is 8/10 = 0.8 Amps,
<CD 3A> is 2 bytes of CRC calculated by the server.

4.3.8 - Gripper Temperature register (0x03EC)

The **Gripper Temperature** register stores the value of the internal temperature of the gripper measured by 2 thermometers. The values are stored in degrees Celcius.

- Address: 03ECh
- Read Only
- Values in °C

15 - 8	7 - 0
Temperature A	Temperature B

To read from this register, the users may send:

SEND: **56 04 03 EC 00 01 FD C9**

then the value returned by SusGrip RTU server is:

RESPONSE: **56 04 02 29 2A 52 B3**

where <29 2A> is temperature read by 2 thermometers,
here tempA = 29h = 41 °C, tempB = 2Ah = 42 °C,

<52 B3> is 2 bytes of CRC calculated by the server.

4.3.9 - GPIO Preset registers (0x07D4, 0x07D5)

The **Set GPIO Close Position** register sets the position to which the gripper moves when commanded to close in GPIO operation mode. Similarly, the **Set GPIO Open Position** register sets the position to which the gripper moves when commanded to open in GPIO operation mode. Like the Set Position register, the value must be doubled before sending.

- Close Address: 07D4h
- Open Address: 07D5h
- Write Only
- Value / 2 = Real physical position

15 - 0
Value / 2 = Preset opening/closing distance in mm

The users may setup these registers only once, then the values are written directly to the gripper's non-volatile memory, and will be used for the latter sessions of operation.

The example procedure to setup for GPIO mode operation is as follow:

1 - Power ON gripper;

2 - Connect to gripper's RTU server using RS485;

3 - Switch to Modbus RTU mode:

SEND: **56 06 07 D0 00 03 C4 A1**

RESPONSE: **56 06 07 D0 00 03 C4 A1**

(See section 4.3.2)

4 - Setup closing position for GPIO mode, example 10mm:

SEND: **56 06 07 D4 00 14 C5 6E**

RESPONSE: **56 06 07 D4 00 14 C5 6E**

where <00 14> = 20 = 2x 10mm;

5 - Setup opening position for GPIO mode, example 125mm:

SEND: **56 06 07 D5 00 FA 14 E2**

RESPONSE: **56 06 07 D5 00 FA 14 E2**

where <00 FA> = 250 = 2x 125mm;

6 - Switch back to GPIO mode:

SEND: **56 06 07 D0 00 07 C5 62**

RESPONSE: **56 06 07 D0 00 07 C5 62**

(See section 4.3.2)

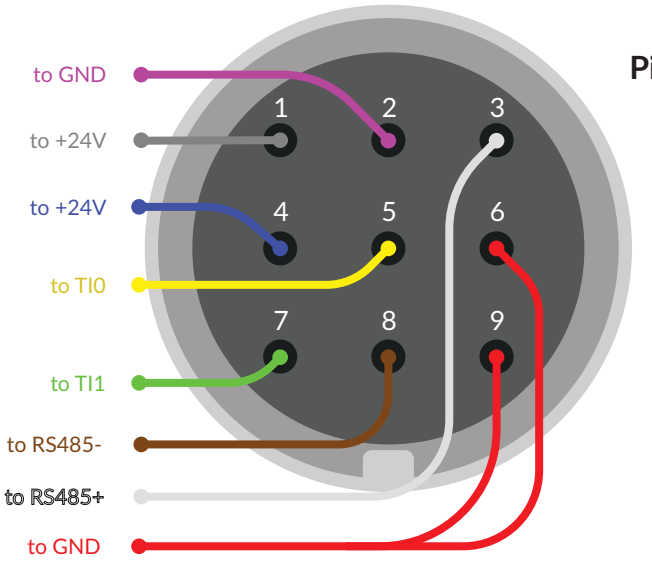
After this, the gripper can be operated in GPIO mode with the preset parameters.

5 - URCAP GUIDE

SusGrip is compatible with most Universal Robots' cobots and is a part of the Polyscope and PolyscopeX ecosystem. SusGrip enables maximum ease of integration into already existing systems that utilize Universal Robots' arms by providing intuitive URCap versions and simple cable connectivity.

5.1 - Connections to URCap

For UR cobots whose end-tool connection is supported by SusGrip, simply connect the gripper to the robot's end-tool port using the provided URCap cable. Then skip to the next section to make SusGrip controllable with Polyscope. Please refer back to chapter 2 section 2.4 for further information. The figure below illustrates the pin mapping of the URCap cable:



Pin No.	Cable color	Description
1	GREY	+24V input
2	PINK	Jointed with 9
3	WHITE	RS485+
4	BLUE	Jointed with 1
5	YELLOW	Not used
6	RED	GND
7	GREEN	Not used
8	BROWN	RS485-
9	RED	GND

Figure 5.1 - SusGrip gripper connections to robot

For UR cobots whose end-tool connection is **NOT** supported, SusGrip **CANNOT** be used with Polyscope as a URCap program. In this case, users must connect the gripper to robot's control box following the general pin mapping of the provided cable. Control of the gripper is hence similar to general applications as described in the previous chapter.

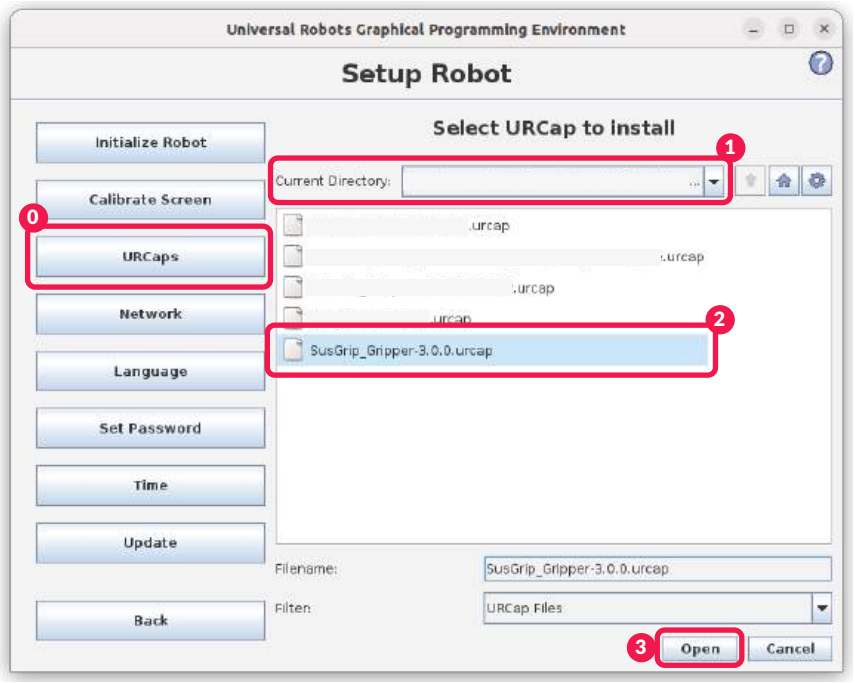
5.2 - Installing URCap plug-in for Polyscope

To install SusGrip URCap plug-in, users should prepare an USB removable drive with the .urcap file. This is either provided by Apicoo when the users purchase the URCap package, along with the required URCap cable, or downloaded from the official website of Apicoo Robotics.

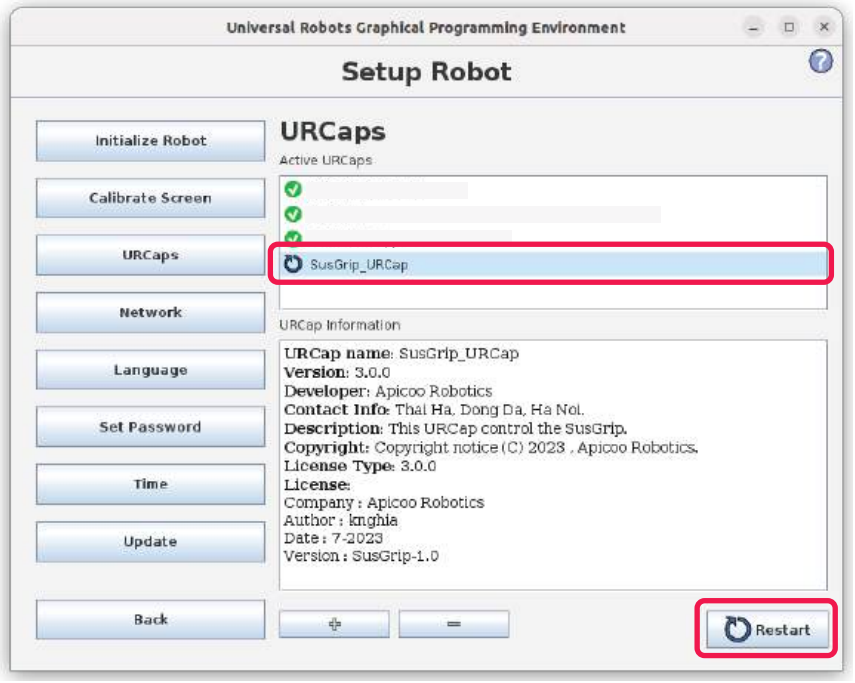
ATTENTION:
Apicoo suggests that users obtain software for URCap only from the official website.
To prevent security risks, please only download from apicoorobotics.com.

5.2.1 - For Polyscope version 3 (UR3, UR5, UR10, and non "e" URx robots):
Insert the USB removeable device containing the .urcap file downloaded from Apicoo Robotics' official website into the USB port located at the upper right-hand side corner of the Teach Pendant.

From the welcome screen on the Teach Pendant, choose "Setup Robot", select "URCaps", then hit the plus-sign icon to add new URCap. Navigate to the directory containing the .urcap file and select "Open" to install SusGrip URCap:



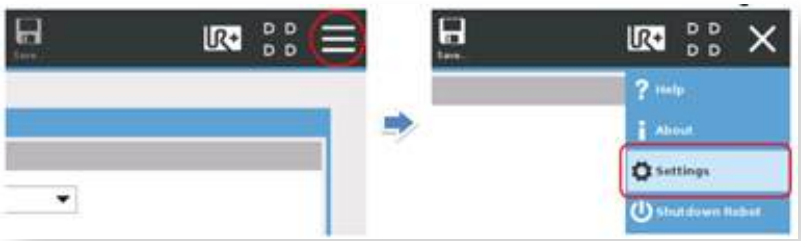
Then, users will be asked to restart the Teach Pendant for changes to take effect. Simply select the "Restart" button:



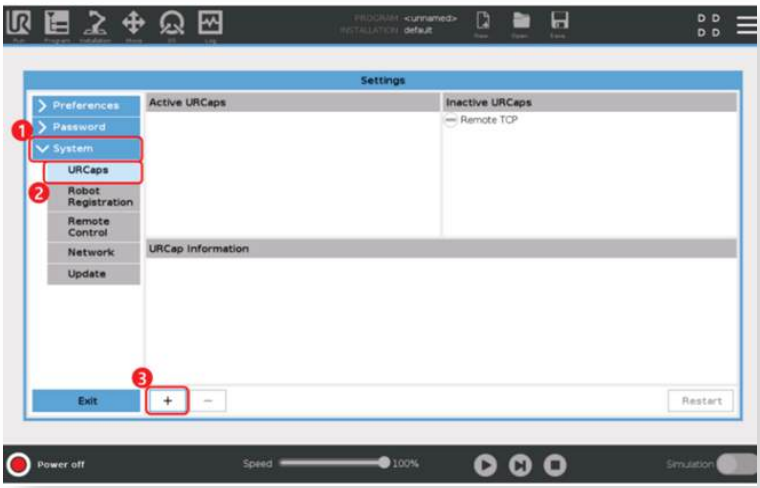
5.2.2 - For Polyscope version 5 (URx_e robots):

Insert the USB removeable device containing the .urcap file downloaded from Apicoo Robotics' official website into the USB port located at the upper right-hand side corner of the Teach Pendant.

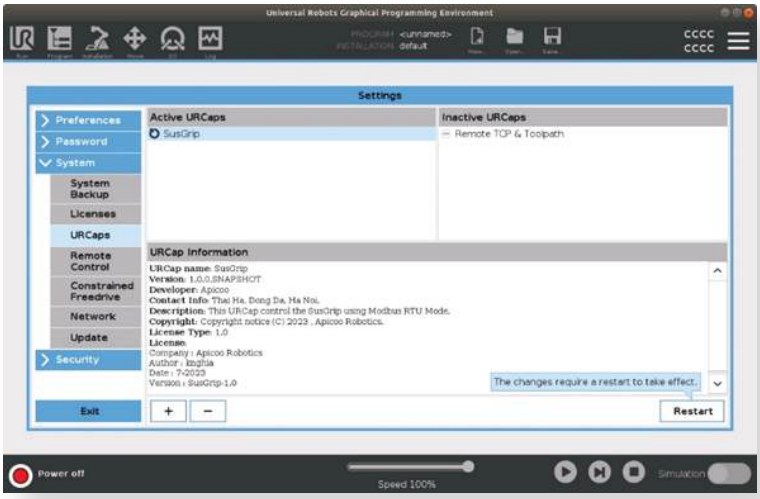
From the home screen on the Teach Pendant, go to Settings by expanding the "three-stripe button" on the very top left corner of the screen, then select "Settings":



In the Settings menu, expand the "System" category on the left-hand side panel and select "URCaps":



Navigate to the directory containing the .urcap file and select "Open" to install Sus-Grip URCap. After this, users will be asked to restart the Teach Pendant for changes to take effect. Simply select "Restart":

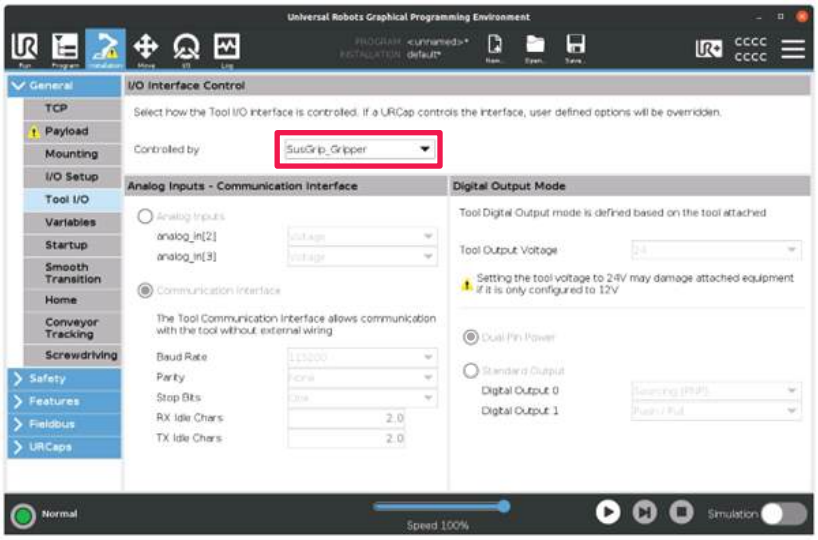


5.3 - Pre-programing Setup

This section mainly addresses tool I/O configuration in case the users are struck with errors or warnings. If no errors or warnings are present, users may skip this section and proceed to main programming.

For Polyscope version 5, if prompted with the following warning: **Please assign control to this URCap on the tool I/O installation tab.**, proceed with the steps below:

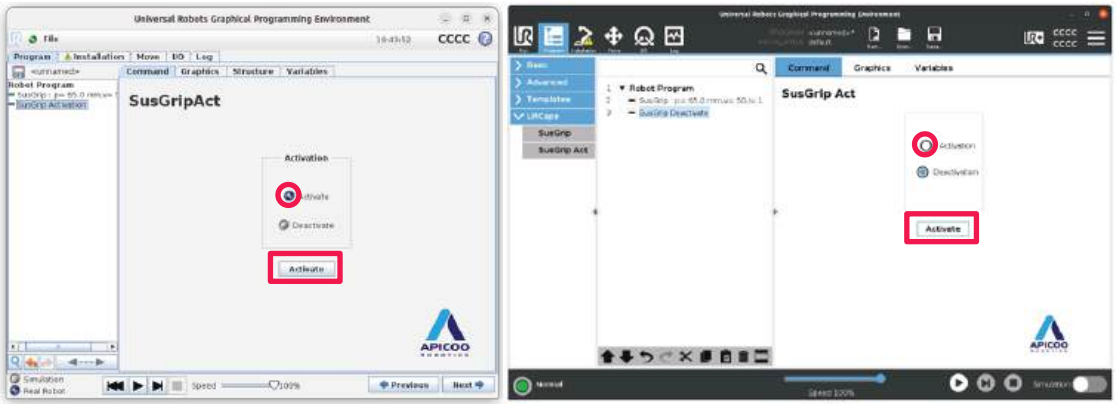
Open the “Installation” tab, which is located on the ribbon on the upper left corner of the Teach Pendant screen. Then expand the “General” section on the left-hand side panel and select “Tool I/O”:



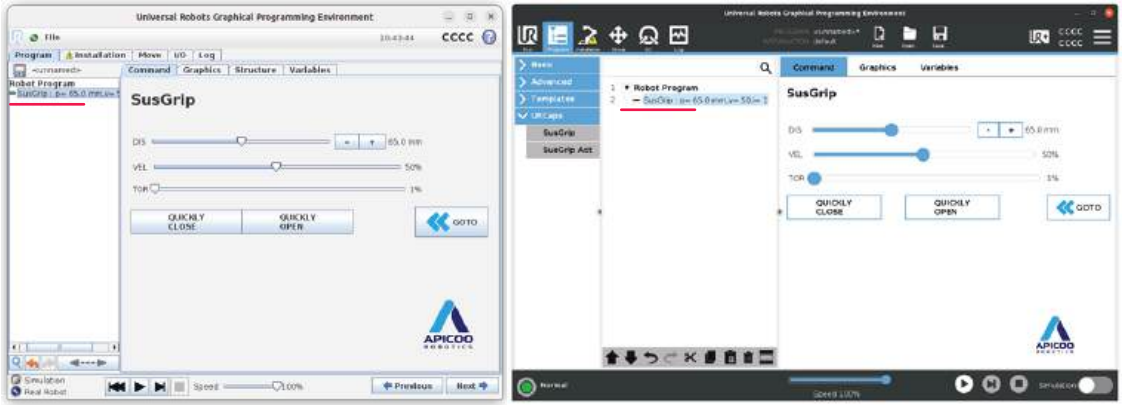
In the “Controlled by” drop down list, choose “SusGrip_Gripper” and let the software configure automatically.

5.4 - Programming SusGrip gripper with Polyscope

For both Polyscope version 3 and version 5, it is recommended to add a “Before Start” block, within which users should add the “SusGrip Activation” command:

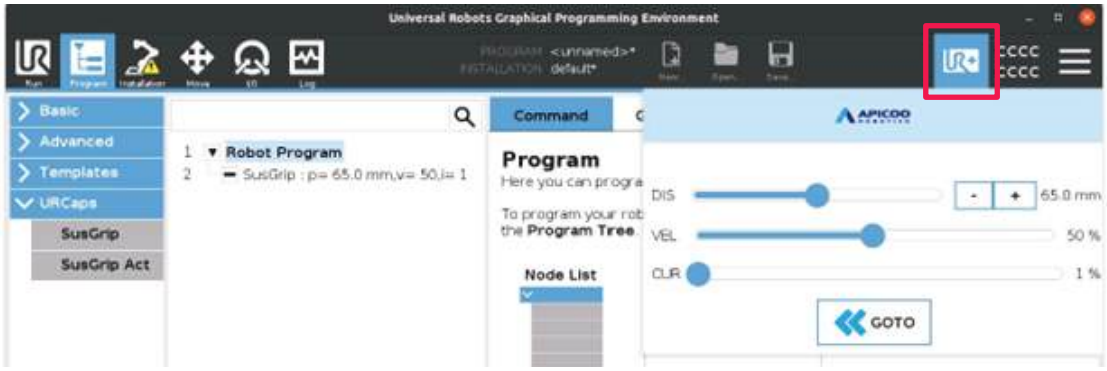


While programming in Polyscope, SusGrip gripper should be considered as a complete robot entity. There fore it is highly recommended that users add a new “Robot Program” block:



The command interface should be self-explanatory and intuitive. Notice the “GOTO” button on the far right-hand side on the “Command” panel with the double arrow icon: This sends the current command parameters on the screen (for example DIS 10mm VEL 50% TOR 20%) instantly and directly to the gripper without users having to run the entire program tree. This is particularly useful during the programming process when users want to try out certain setpoints without having to the main program from top to bottom.

Additionally, the same effect can be achieved in Polyscope version 5, regardless of whether users are in the “Program” tab or not, through the exclusive URCap Toolbar:



5.5 - Script functions

For advanced control of SusGrip gripper, Apicoo provides numerous script functions that can be found in the script menu in Polyscope. Below is the list of available functions:

susgrip_activate()

func Activate the gripper.

susgrip_deactivate()

func Deactivate the gripper.

susgrip_set_pos(pos)

func Command the gripper to move to the desired position.

param ***pos*** Target position, in range (0 - 130)mm.

susgrip_set_vel(vel)

func Set the velocity at which the gripper travels when commanded to move.

param ***vel*** Travel velocity, given as %.

susgrip_set_tor(tor)

func Set the torque strength with which the gripper reaches the commanded target position.

param ***tor*** Torque strength, given as %.

susgrip_set_pvt(pos, vel, tor)

func Command the gripper to move to the desired position with specified velocity and torque.

param ***pos*** Target position, in range (0 - 130)mm.

vel Specified velocity, given as %.

tor Specified torque, given as %.

susgrip_move_is_finish()

func Get the moving status flag of the gripper.

return ***0*** Move not finished, is moving.

1 Move finished, stopped moving.

susgrip_get_pos()

func Get the instantaneous position of the gripper's fingers.

return Gripper's fingers instantaneous position, in range (0-130)mm.

susgrip_get_obj()

func Get the object detection status flag of the gripper.

return ***0*** No object detected.

1 Object detected while closing.

2 Object detected while opening.

3 Object dropped detected.

susgrip_get_fault()

func Get the errors flag of the gripper.

return ***0*** FLT_NONE, no error.

1 FLT_nOCTW, driver over-temperature warning.

2 FLT_nFAULT, driver over-current error.

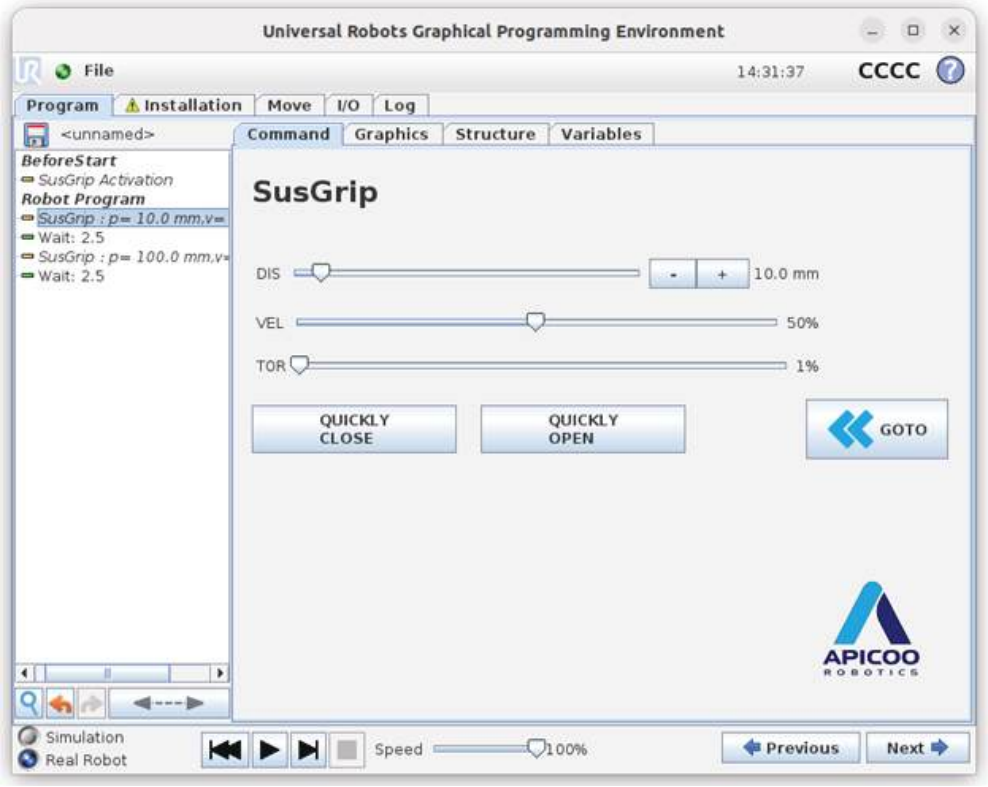
3 FLT_OVER_TEMP, gripper over-temperature.

4 FLT_OVER_CUR, gripper over-current.

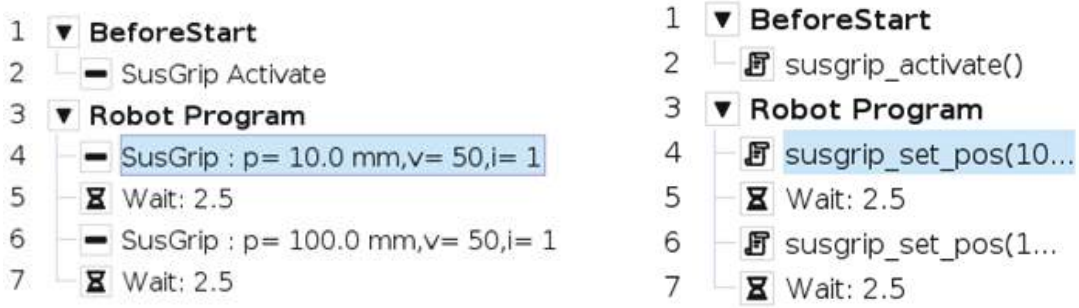
5 FLT_OVER_DIS, gripper position out of range.

5.6 - Example URCap programming with Polyscope

Below is a simple program tree for SusGrip gripper in Polyscope version 3. The “Before Start” block activates the gripper. The main “Robot Program” block first closes the gripper to 10.0mm, waits for 2.5s, opens the gripper to 100.0mm, then waits for 2.5s:



Below is a simple program tree for SusGrip gripper in Polyscope version 5. The “Before Start” block activates the gripper. The main “Robot Program” block first closes the gripper to 10.0mm, waits for 2.5s, opens the gripper to 100.0mm, then waits for 2.5s:



Notice that the program tree on the left uses commands from GUI, while the one on the right uses script functions to achieve the same behavior.