APICOO ROBOTICS

SusGrip Smart Gripper

USER MANUAL

August 2024
apicoorobotics.com

CONTENT

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 effecient yet robust operation for a wide range of applications.

SusGrip gripper is equipped with a high precision absolute encoder for accurate position monitoring of each finger, 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 Auto Self Locking. When the Self-Locking break is engaged, the gripper fingers are fixed in place and cannot be moved manually. For manual gripper adjustment during power-down, users may refers to the didcated section which will be eleborated later in this manual.

The modular design of the SusGrip allows users to easily swap and change 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).

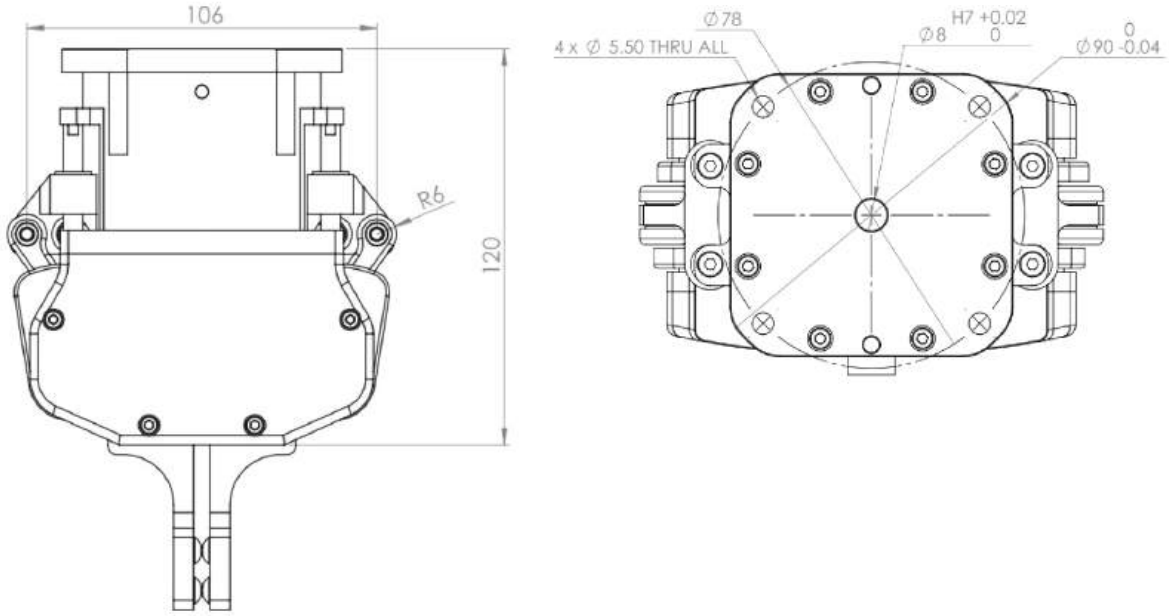


Figure 2.1 - General dimensions of the gripper:
Side view (left) and Top view (right)

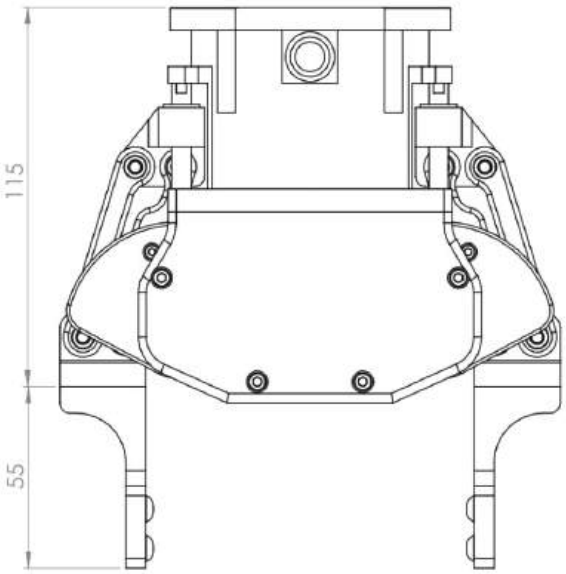
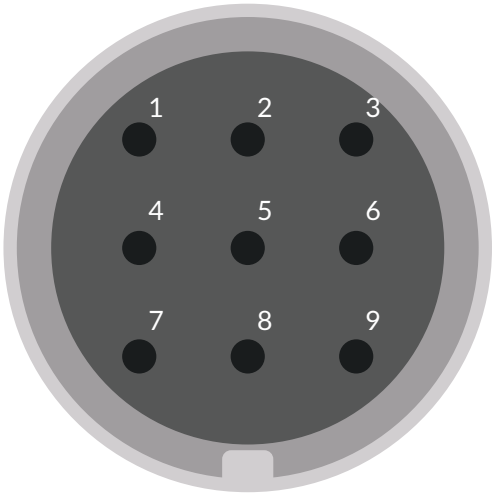


Figure 2.2 - Gripper finger dimensions

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

Figure 2.4 - Communication port on the SusGrip gripper

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

2.3 - Product ratings

ENTRY	VALUE	UNIT
Grasp force	100	N
Stroke	0-128	mm
Form-fit payload	05	kg
Voltage	24	V
Max current	2	A
Repetition accuracy	0.1	mm
Position resolution	0.5	mm
Speed	120 (All fingers)	mm/s
Weight	1250	gr

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 connectivity to multiple models of industrial robots. The figure below demonstrates the setup for the product:

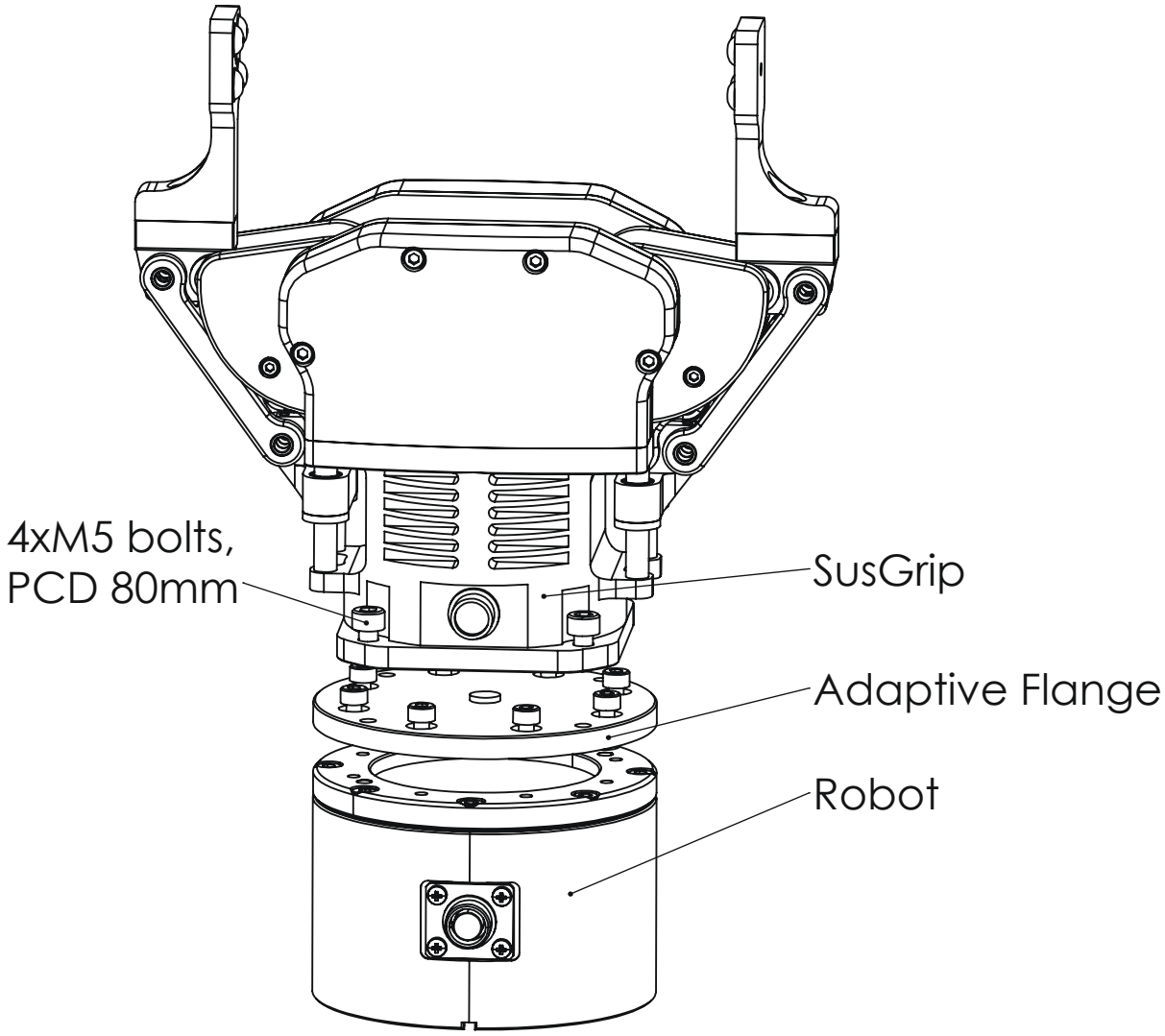


Figure 3.1 - Installing SusGrip gripper onto robotic arm

3.2 - Electrical setup

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

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

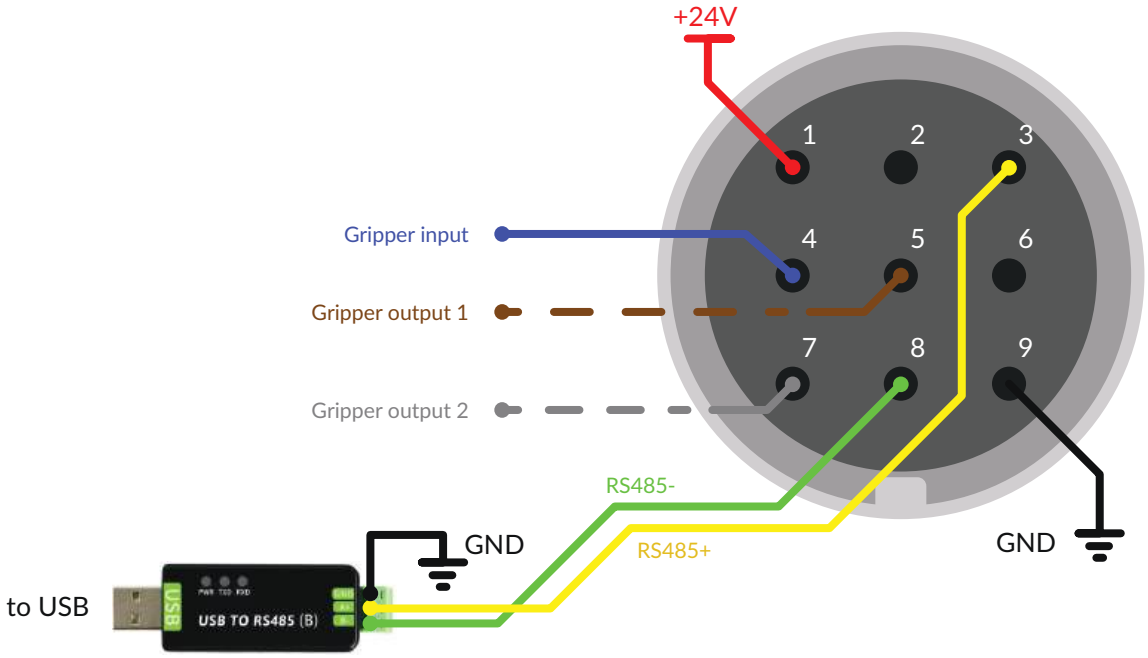


Figure 3.2 - SusGrip gripper connections to robot control box

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, DO make sure to configure the pinout appropriately for correct operation.

3.3 - Performing tests on the SusGrip gripper

To perform tests on the gripper, the users can either do wires shorting directly or send commands through the SusGrip GUI application.

3.3.1 - Testing the gripper by hardware

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

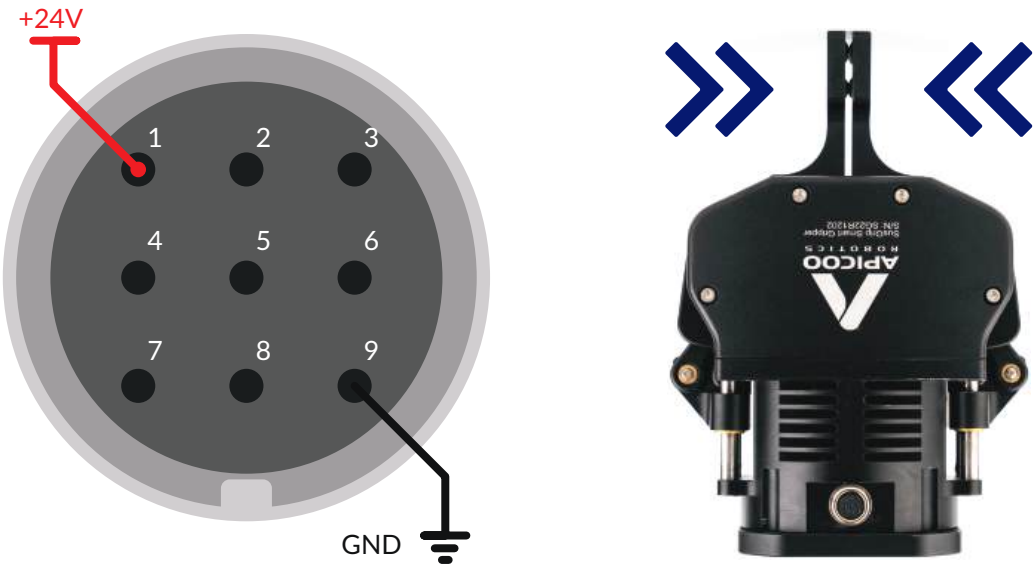


Figure 3.3 - Gripper closes its fingers when Pin 4 left open

Next, pull the input pin of the gripper to GND by shorting Pin 4, or the green wire, to either Pin 9 (the black wire), or Pin 6 (the white wire). If by doing so, the gripper opens its fingers, then the functionality of the gripper can be verified as OK:

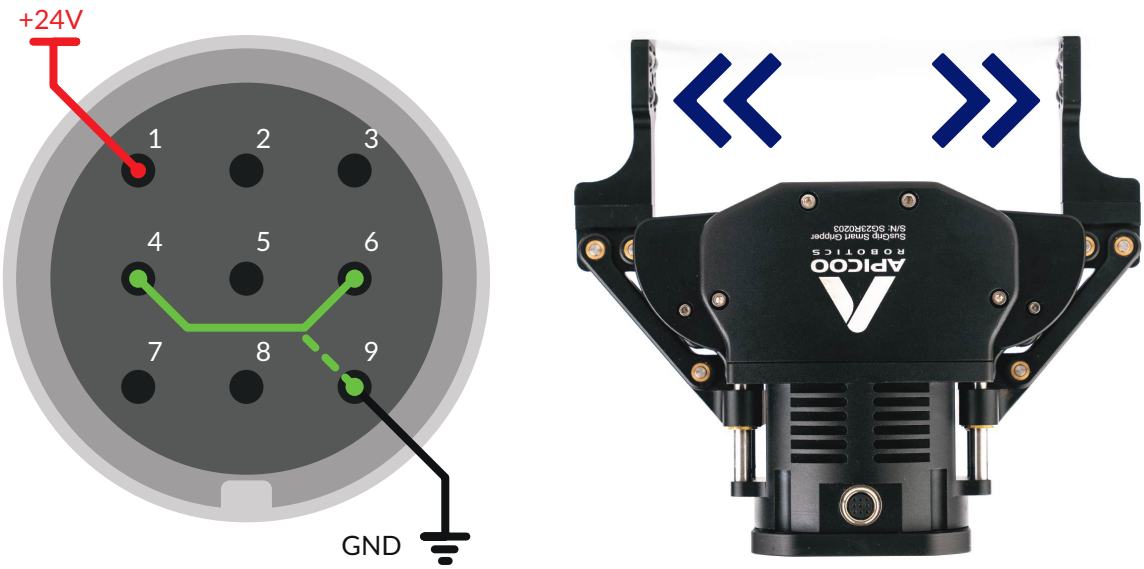


Figure 3.4 - Gripper opens its fingers when Pin 4 connected to GND

3.3.2 - SusGrip GUI software installation and setup

To ensure easy setup precedure for the 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 through direct contact with Apicoo Robotics support agents.

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

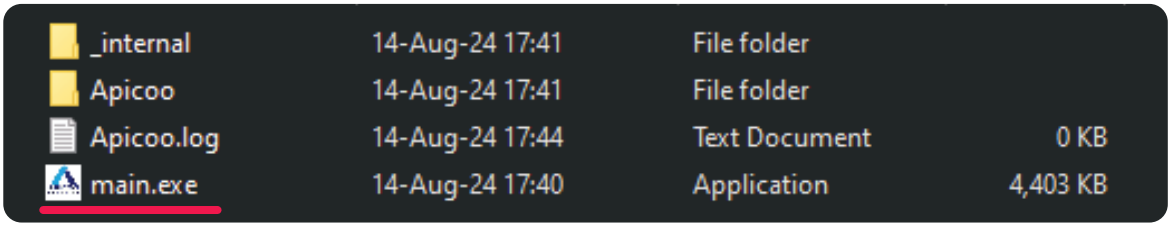


Figure 3.5 - SusGrip GUI application files

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

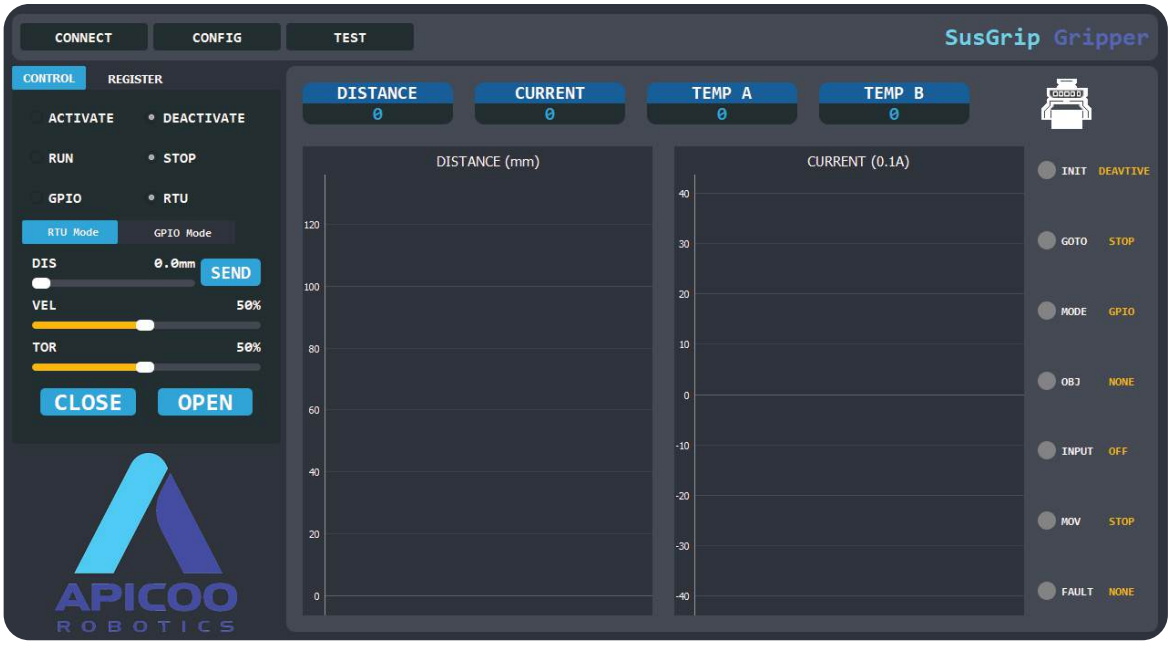


Figure 3.6 - SusGrip GUI application

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 with 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.

Afterward, connect the USB TO RS485 device to any USB port available on a computer. Upon successful connection, a blue LED on the gripper body turn on then remain 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:

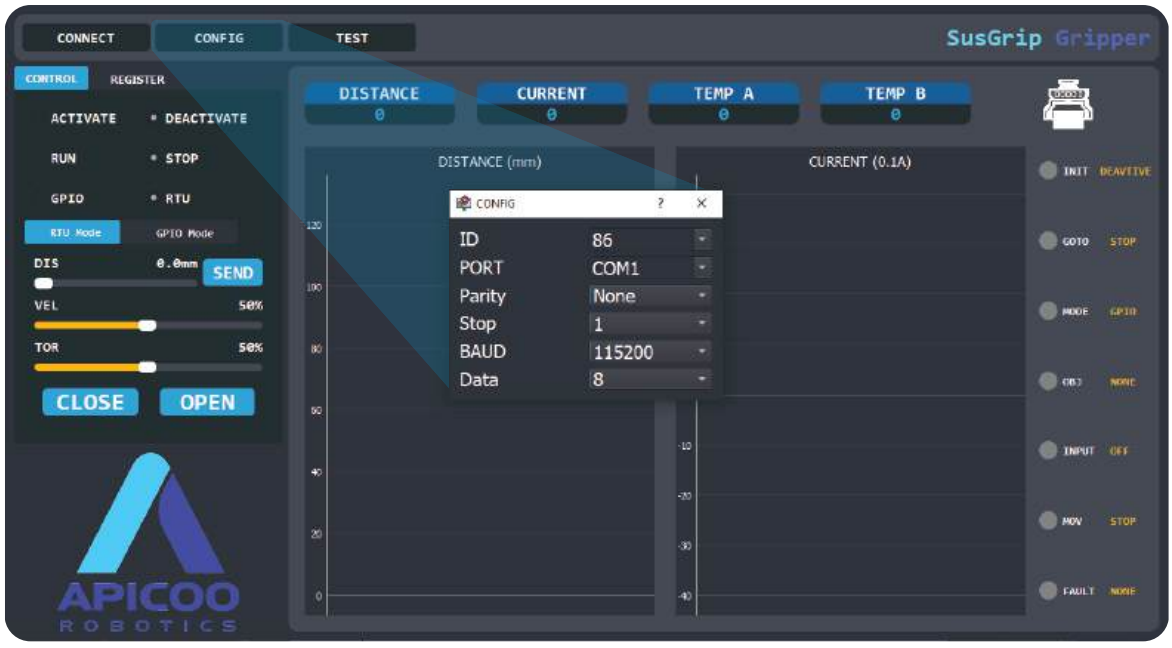


Figure 3.8 - On the SusGrip GUI application, click on “CONFIG”

Then, click on “CONNECT”:

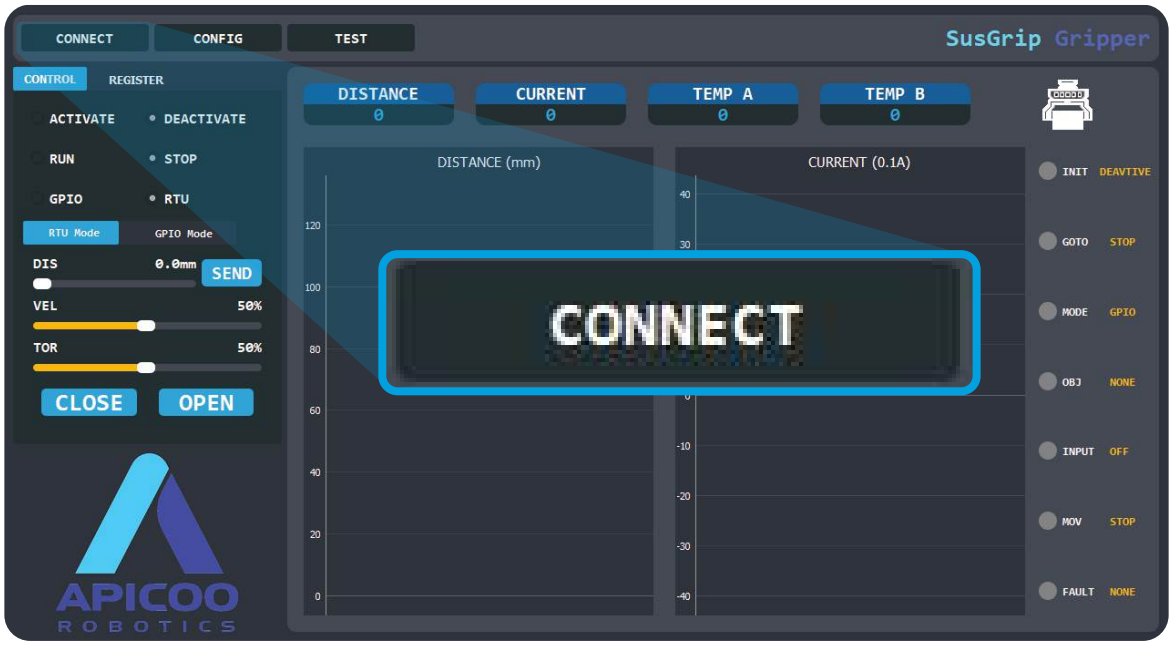


Figure 3.9 - Click on “CONNECT” to start controlling the gripper

Finally, with the sliders on the left-hand side tab, the 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.

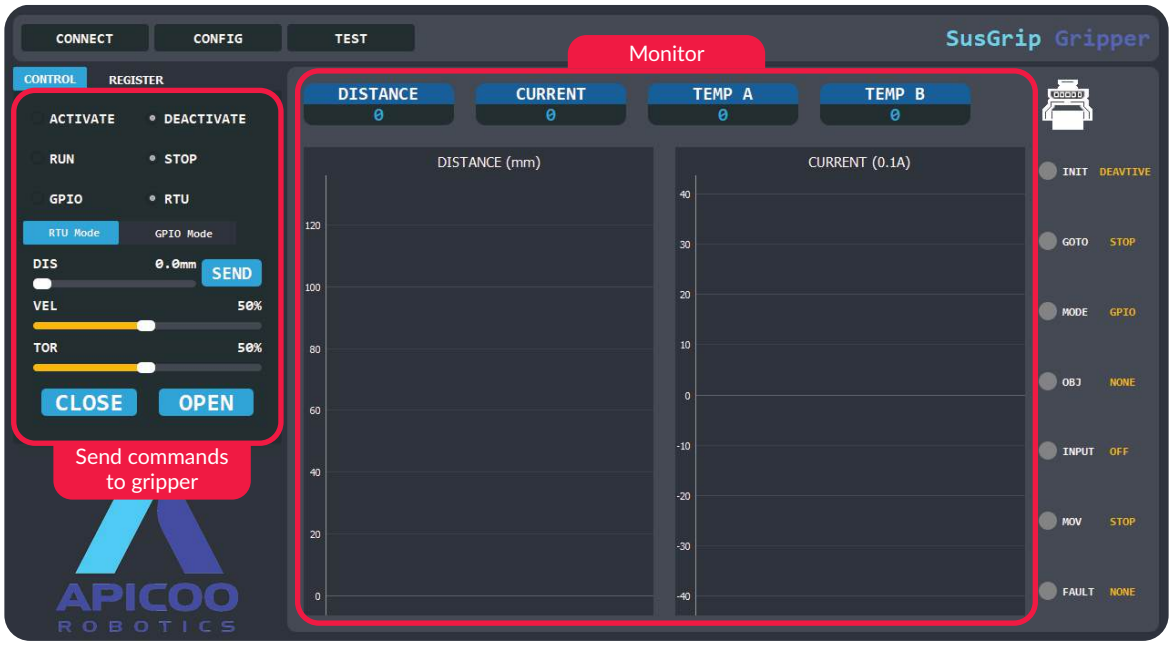


Figure 3.10 - Controlling and monitoring panels

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

4 - OPERATION GUIDE

Apicoo offers in total 2 modes of control to the SusGrip gripper: GPIO mode and RTU mode. Upon powering up, the gripper will default 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 up to HIGH logic level by default. The gripper closes its fingers when Pin 4 is written HIGH, and opens its fingers when Pin 4 is written LOW.

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, the 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 it 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 1-5cm/s)
[2003]: 1-255 (% max torque in current 1.2-4.0A)
[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% (2.5cm/s), and torque to 50%:

[2000]: 7 (Mode of operation - **STRICTLY**)
[2001]: _
[2002]: 127 (50% max velocity: 2.5cm/s)
[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:.

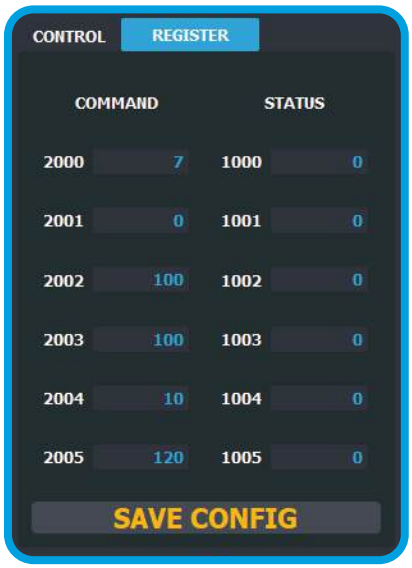


Figure 3.12 - Click on SAVE CONFIG to apply changes to preset parameters

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

4.2 - Operating SusGrip gripper in ModbusRTU mode

The ModbusRTU protocol of the SusGrip gripper is configured as follow:

Baudrate:	**115200**
Data bit/Parity/Stop bit:	**8/N/1**
Modbus functions:	**FC04, FC16**
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 of these registers is elaborated.

Register 2000 controls the gripper status and mode of operation:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	x	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: ModbusRTU mode with RS485
- 1: GPIO mode

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

Register 2001 controls the instantaneous position (distance between fingers) of the gripper:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x								

Bit 0-7 take the values of distance in milimeters from 0 to 255, divided by 2, the actual setable value is constrained from 0 to 132 mm

Bit 8-15 are reserved, DON'T CARE.

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

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 control the preset values for closing distance and opening distance in GPIO mode, respectively:

Bit 0-7 take the values of distance in milimeters from 0 to 255,
the actual setable value is constrained from 0 to 132 mm
Bit 8-15 are reserved, DON'T CARE.

Register 1000 shows the gripper status:

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: ModbusRTU 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 released

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 holds the information on error codes during operation process of the gripper:
- 0000 0000: No error
- 0000 0001: Driver current limit exceeded (DRV OCTW)
- 0000 0010: Driver error (DRV FAULT)
- 0000 0011: Over-current
- 0000 0100: Gripper overheated
- 0000 0101: Target object out of gripper's range

Registers 1001, 1002, 1003, and 1004 show the position (distance between fingers) set by the users to register 2001, the actual position (distance between fingers) read by encoder, the torque (current) being applied, and the temperature of the gripper, respectively:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x								

Bit 0-7 hold the value of the respective data on set position (distance), actual position (distance), current applied, and temperature, from 0-255. Position data is in milimeters, torque (current) data is in % max value, and temperature data is in degree Celcius.
Bit 8-15 are reserved.

Register 1005 is reserved.

Example: Manipulating registers in ModbusRTU 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 ModbusRTU 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