qb robotics



Universal Robot Manual

PLEASE READ CAREFULLY THESE INSTRUCTIONS BEFORE USE. DO NOT DISCARD: KEEP FOR FUTURE REFERENCE.

Dear customer,

Thank you for purchasing our product.



The present document provides information at best of our knowledge at the time of publication. This document could present differences from the product and it is subject to changes without notice: the latest version is available on our webpage www.qbrobotics.com.

qbrobotics s.r.l. does not assume any responsibility for errors or omissions.

In no case qbrobotics s.r.l. will be responsible for any loss, problems or damages to persons or property arising from the use of this document.

The qb® logo and qbrobotics® are registered trademarks of qbrobotics s.r.l. In the following, the indications of (R) are omitted.

Table of Contents

1 Preface 2

1.1 Using this document 2

1.2 Symbols and conventions 2

2 Safety 2

2.1 Safety 2

3 Description 5

3.1 Device identification 5

4 Technical Data 6

4.1 Center Of Mass 7

4.2 Inertia Tensor 7

4.3 Environmental Specifications 7

4.4 Electrical Specifications 7

4.5 Fingers workspace 8

4.6 Adjustable wrist 9

4.7 Glove substitution 10

5 Mounting and Wiring 12

5.1 Tool mounting 12

5.2 Cable connections 15

6 Device Control 23

6.1 Digital I/O 23

6.2 C++ API 25

6.3 ROS 28

6.4 URCap 31

7 Maintenance and warranty 37

7.1 General 37

8 Certifications 38

9 FAQ and troubleshooting 40

9.1 PC drivers and requirements 40

9.2 The hand never connects to the robot system 43

9.3 Error C740A0 on e-Series robots 44

9.4 Grasping with a custom finger position 45

9.5 Simplified CAD models of the qb SoftHand 46

9.6 The demo is not as good as expected 46

9.7 The glove is deteriorated/broken 48

9.8 The hand does not reopen correctly 49
9.9 The hand is connected but does not move 50
9.10 The hand often disconnects while the robot is moving 50
9.11 The USB flash drive is not working 51

1 Preface

1.1 Using this document

The documentation must always be complete and in a perfectly readable state.
Keep the document accessible to the operating and, if necessary, maintenance personnel at all times.
Pass the document to any subsequent owner or user of the product.

1.2 Symbols and conventions

STYLE	DESCRIPTION
CRITICAL	Identifies information about practice or circumstances that can lead to critical damages on the device and to personal injury. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.
WARNING	Identifies information about practice or circumstances that can lead to damages on the device and to personal injury. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.
NOTE	Identifies remarkable information and additional notes.
TIPS	Identifies tips and highlights.
`Monospaced text`	Identifies file paths, file names and software functions.

2 Safety

2.1 Safety

2.1.1 Intended use

The product design is intended for grasping objects in the weight range from 1 to 2000 g.
Fragile, sharp or sharp-edged objects shall not be grasped.
Objects having the ratio between the biggest and the smallest dimensions ≥8 and the main dimension greater than 300 mm shall not be grasped.
The product is intended for installation on robotic arms: the safety features are stablished only for use as described in this document.
The safety of the product cannot be guaranteed in case of inappropriate use. One, single, inappropriate use can result in a permanent damage to the safety of the product. The responsibility for any damage caused by an incorrect choice of the manipulated object is the responsibility of the user.

2.1.2 Safety instructions

WARNING:

- Check that all the content is intact after removing it from the packaging.
- The device can be used only by specially trained staff.
- Disconnect the power supply before installation, cleaning or maintenance operations.
- Make sure that no residual energy remains in the system.
- Always operate the product within the specifications defined.
- Keep away from children and pets. Always set off or unplug when not in use.
- Never use aerosol products, petroleum-based lubricants or other flammable products on or near the end-effector.
- Do not use any damaged power cable, plug, or loose outlet. It may cause damages to the product or injury to people.
- Do not touch electrical components to avoid damages due to electrostatic charges.
- Make sure the end-effector is properly and securely bolted in place and cabled.
- Do not use if damaged or defective. Do not disassemble.
- Do not insert any objects between moving parts of the fingers.
- Keep head and face outside the reach of the end-effector.
- Do not wear loose clothing or jewelry when working with the end-effector.
- Disrespect of these precautions can affect safety of the device.

2.1.3 EC Directives on product safety

- The following EC directives on product safety must be observed.
- If the product is being used outside the UE, international, national and regional directives must be also observed.

Machinery Directive (2006/42/EC)

The Machinery Directive is applyed to our products. The products described here are "incomplete machines", so installation instructions are normally issued by qbrobotics.

Low Voltage Directive (2014/35/EU)

The Low Voltage Directive applies for all electrical equipment with a nominal voltage of 75 to 1500 V DC and 50 to 1000 V AC. The products described in this device manual do not fall within the scope of this directive, since they are intended for lower voltages.

2.1.4 Environmental conditions

Wrong environmental and operating conditions can lead to injuries, product damages and/or significant reduction to the product's life.

WARNING:
Any use or application deviating from intended use is deemed to be impermissible misuse. This includes, but is not limited to:

- Use before performing a risk assessment;
- Use outside the permissible operational conditions and specifications;
- Use in not low-dust environment;
- Use in places with high temperature or humidity;
- Use in wet places;
- Use in potentially explosive atmospheres;
- Use in medical and life critical applications;
- Use close to a human's head, face and eye area;
- Use as a climbing aid;
- Use in outdoor applications.

2.1.5 Environmental safety

The qb SoftHand Industry must be disposed of in accordance with the applicable national laws, regulations and standards.
All the components of this product have been chosen in accordance with the EU RoHS directive 2011/65/

EU: they are produced with restricted use of hazardous substances to protect the environment.
Observe national registration requirements for importers according to EU WEEE Directive 2012/19/EU.





2.1.6 Personnel qualification

If the personnel working with the product is not sufficiently qualified, may occur serious injuries and damage.

- Observe the national safety regulations and rules and general safety instructions.
- Before working with the product, the personnel must have read and understood the complete assembly and operating manual.
- All work may only be performed by qualified personnel.

2.1.7 Glove

The device is supplied with a glove (protective) basic accessory. qbrobotics declines all responsibility for the use of any other unauthorized glove, which may result in faulty operation of the device and compromise both the reliability and lifespan of the device.

2.1.8 Power supply

The equipment must be powered from a PELV source in order to avoid electrical hazards.
Typical input power supply for external driver:

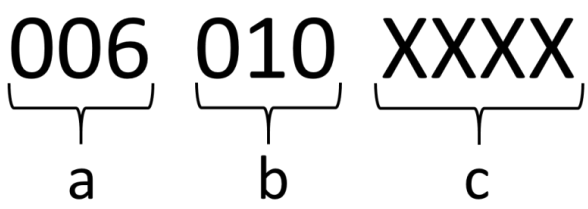
- Power: 30W;
- Voltage: from 23V to 26V.

3 Description

qb SoftHand Industry is an anthropomorphic robotic hand based on soft-robotics technology, flexible, adaptable and able to interact with the surrounding environment, objects and humans while limiting the risk of hurting the operators, spoiling the products to be handled, and damaging the robot itself.
The qb SoftHand is adaptable and can grasp different objects without any change in the control action, showing an unparalleled level of simplicity and flexibility.
Thanks to its soft nature the hand by qbrobotics exploits the principles of synergies in a simple and intrinsically intelligent design that is not only safe in unexpected human-robot interaction, but also adaptable to grasp different objects without any change in the control action.
The combination of these innovations results in a flexible prehensile device that can grasp a wide variety of objects. The single-motor actuation makes the hand plug-and-play and simple-to-control (one single motor requires one single control signal to close and open the whole hand) and affordable.

3.1 Device identification

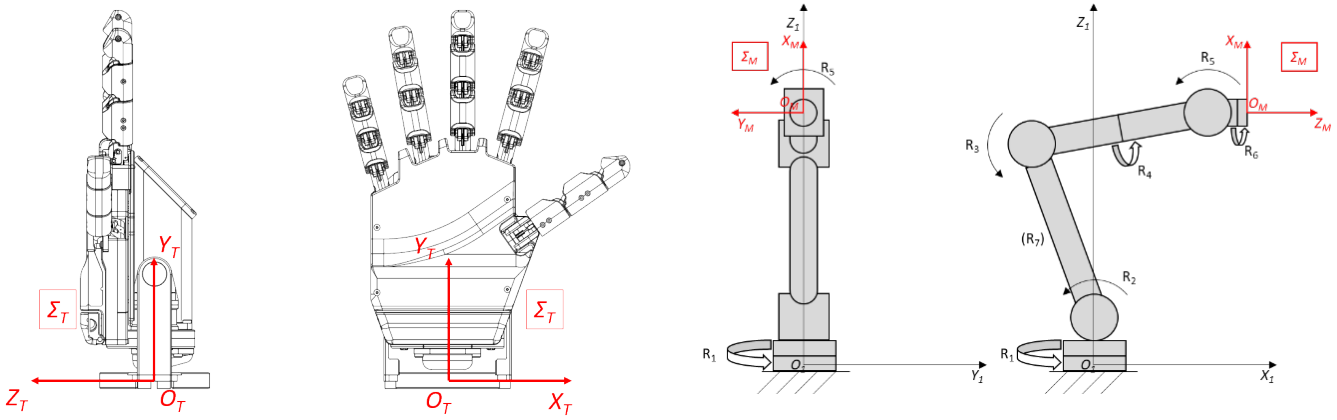
On the rear part of the device there is its serial number, that you can read following the structure below.



- (a) family
- (b) model
- (c) progressive number

4 Technical Data

MECHANICAL	UNIT	NOMINAL	MIN	MAX
weight	[kg]	0.89	-	-
wrist mounting range	[deg]	-	0	90
wrist mounting resolution	[deg]	7.5	-	-
power grasp payload	[kg]	-	-	2.0
pinch grasp payload	[kg]	-	-	0.6
hanging payload	[kg]	-	-	5.0
full closing time	[s]	-	-	1.2



The mechanical interface coordinate system $\mathbf{\Sigma_M}$ (O_M; X_M, Y_M, Z_M) of an articulated robot, as defined by ISO 9787:2013. In particular, the center of the coordinate system, O_M, is on the interface surface of the robot tool flange and the Z_M axis is coincident with the tool flange axis of symmetry. R_i is the i-th revolute joint of the robot. (O_1; X_1, Y_1, Z_1) is the base coordinate system of the robot.

The system $\boldsymbol{\Sigma_T}$ (O_T; X_T, Y_T, Z_T) is the Tool Coordinate System of the device.
The center, O_T, coincides with the center O_M. The Z_T axis is normal to the palm, outgoing positive, and the Y_T axis is normal to the flange plane.
This definition implies that Σ_T is the same for left and right one. So, the positive orientation of X_T axis will be on the side of the thumb in the right hand and on the side opposite to the thumb in the left hand.

4.1 Center Of Mass

coordinates system Σ_T

qb SoftHand Industry State	UNIT	X	Y	Z
fully open	[mm]	2.4	80.5	14.3
fully closed	[mm]	1.1	75.3	17.5

4.2 Inertia Tensor

coordinates system Σ_T

qb SoftHand Industry State	UNIT	XX	XY	XZ	YY	YZ	ZZ
fully open	[kgmm^2]	-9535	-90	335	-9888	-1449	1343
fully closed	[kgmm^2]	-8090	-72	155	-7963	-1752	1371

4.3 Environmental Specifications

Specification	UNIT	NOMINAL	MIN	MAX
operating temperature	[°C]	-	-5	50
storage temperature	[°C]	-	-20	50
noise level	[dB]	52	40	61

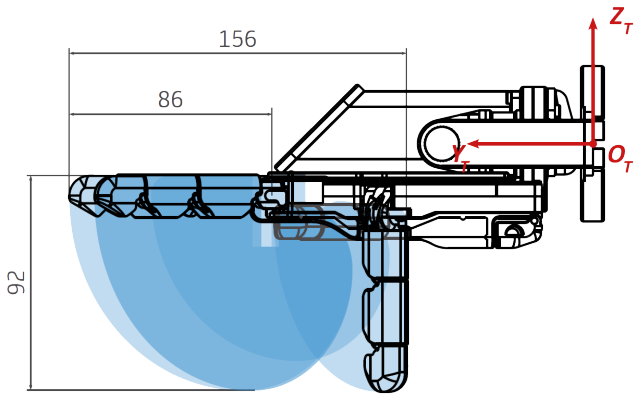
4.4 Electrical Specifications

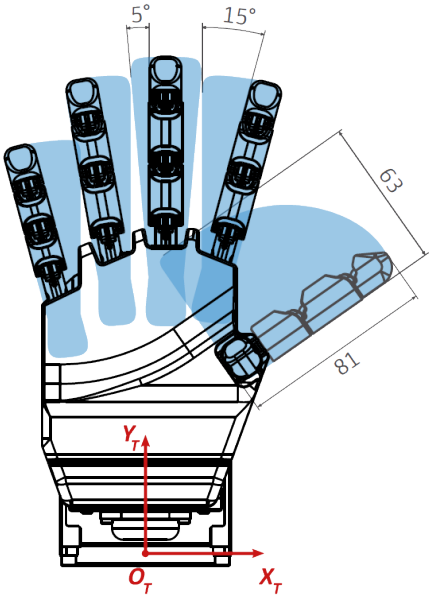
Specification	UNIT	NOMINAL	MIN	MAX
operating voltage	[V]	24	23	26
power consumption	[W]	13	3	15
Control Specification				

communication protocols: EtherCAT, UDP
FEATURE
plug-and-play[1]
soft, human like fingers
adjustable wrist mounting position
splash, water, and dust resistance: IP65
interchangeable glove for special applications

4.5 Fingers workspace

Figures below show the hand workspace, in particular the fingers' range. Dimensions are in millimeters.





4.6 Adjustable wrist

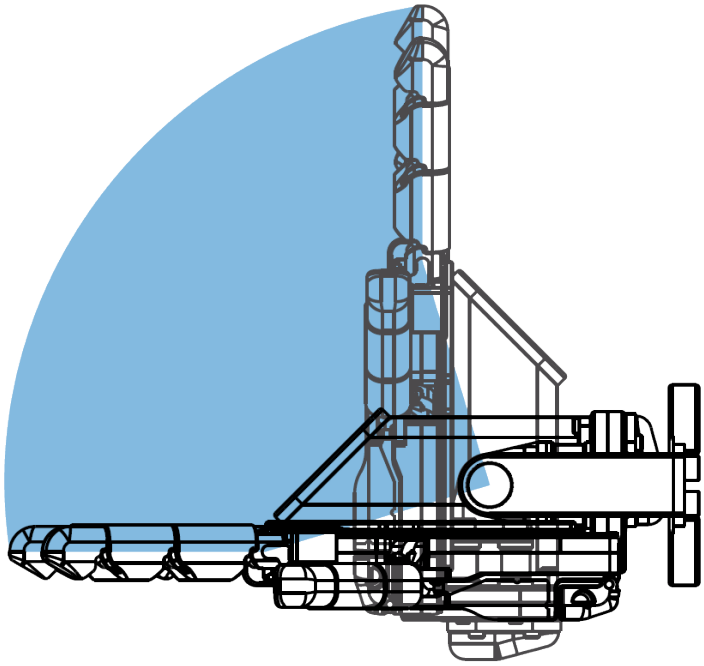
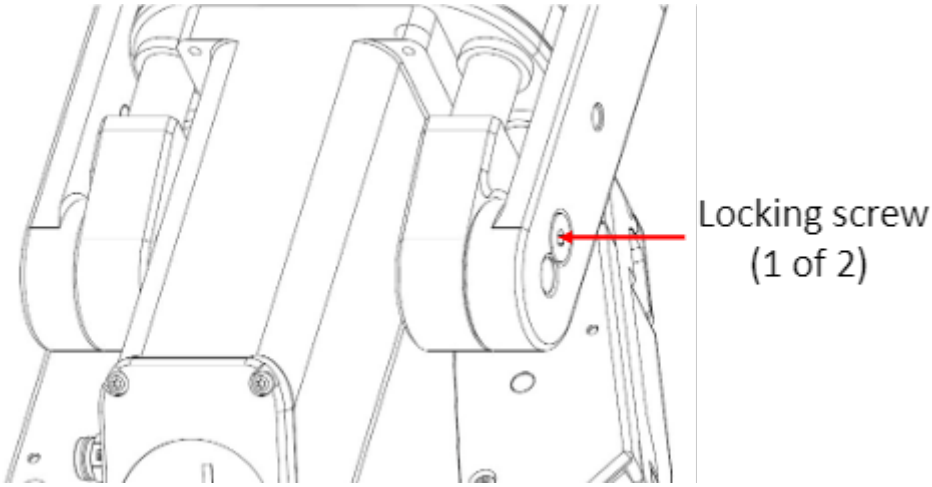


Figure 1. — Extreme positions of the wrist pre-set

You can adjust the pitch pre-set of the wrist in three positions: 0°, 45° and 90° by 2,5mm allen wrench.



Please, follow the sequence below to change the wrist pre-set.

1. remove the 2 screws by 2,5mm allen wrench;
2. rotate the hand at the desired angle;
3. lock the joints by the two screws and a maximum torque of 2Nm.

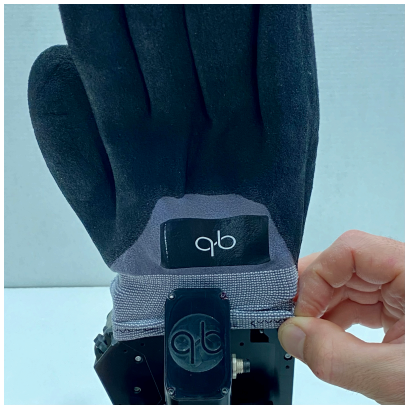
This feature will help you to assemble the hand to the robot flange.

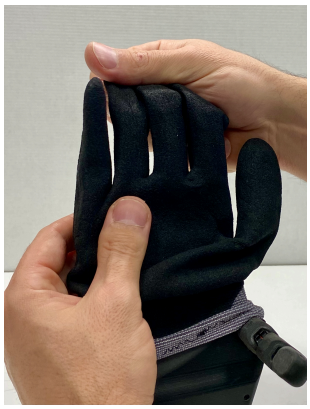
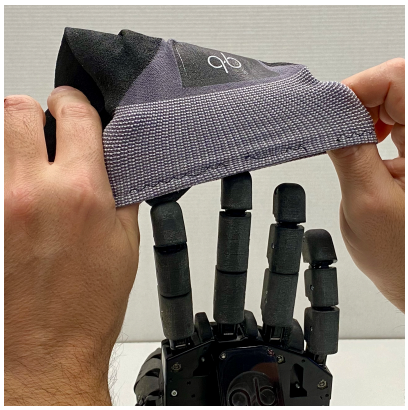
WARNING:

Do not apply a torque more than 2Nm to lock the screws.

4.7 Glove substitution

The device is supplied with a glove (protective) basic accessory.
The glove is interchangeable and can also be replaced by the user. It is recommended to always use the robotic hand with accessory gloves provided by qbrobotics. Alternatively, the user can decide to use commercial gloves of the same type (reference size 11/XXL), taking charge of evaluating the change in performance of the robotic hand.

#	INSTRUCTIONS	PICTURES	
1	Open the hinge near the wrist and remove the blocked part of the glove.		
2	Pull up the glove from the wrist up to the thumb.		

#	INSTRUCTIONS	PICTURES
3	Pull up all the fingertips to let the glove to stay straight. Carefully remove the thumb from the glove.	 
4	Slowly remove the glove finger by finger and than widely open it from the wrist.	

You can follow these instructions from 4 to 1, to put a new glove.

5 Mounting and Wiring

The qb SoftHand Industry Kit allows you to connect the device to your robot arm. The kit consists of:

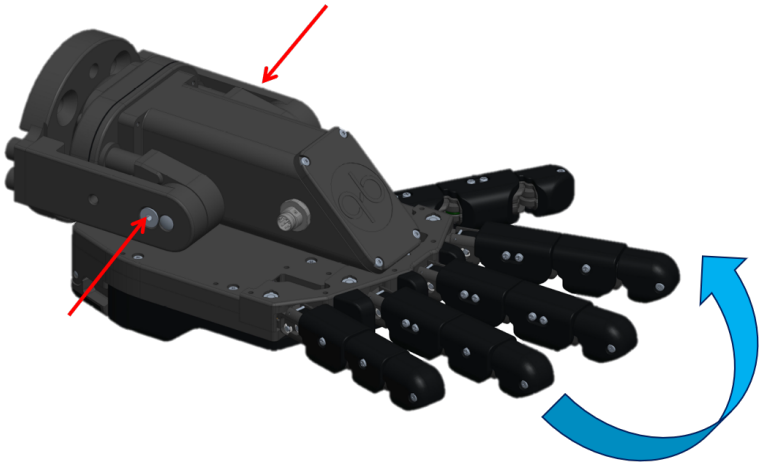
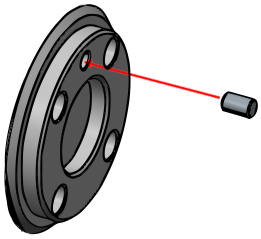
- N.1 qb Softhand Industry 24V;
- N.1 external driver **G-DCWHI2.5/100EES**;
- N.1 ethernet cable **EC_CBL_UTP_6_RJ45_BLK_050**;
- N.1 Mini-B USB cable **EC_CBL_USB2_A_MINI_100**;
- N.1 power cable **EC_CBL_22_02_L50_RR_200**;
- N.1 power cable **EC_CBL_22_01_L5_RR_100**;
- N.1 STO cable **EC_CBL_26_CRMP3_WIRE_050**;
- N.1 power line communication bridge with main cable of 3m **EC_CBL_26_M8FA8_CRIMP_100**;
- N.1 USB pen drive:
  - Manual;
  - Datasheet;
- N.1 cylindrical pin EN ISO 8734 A d6x10 h6;
- N.4 metrical screws EN ISO 10642 M6x10;
- N.2 Allen hex keys.

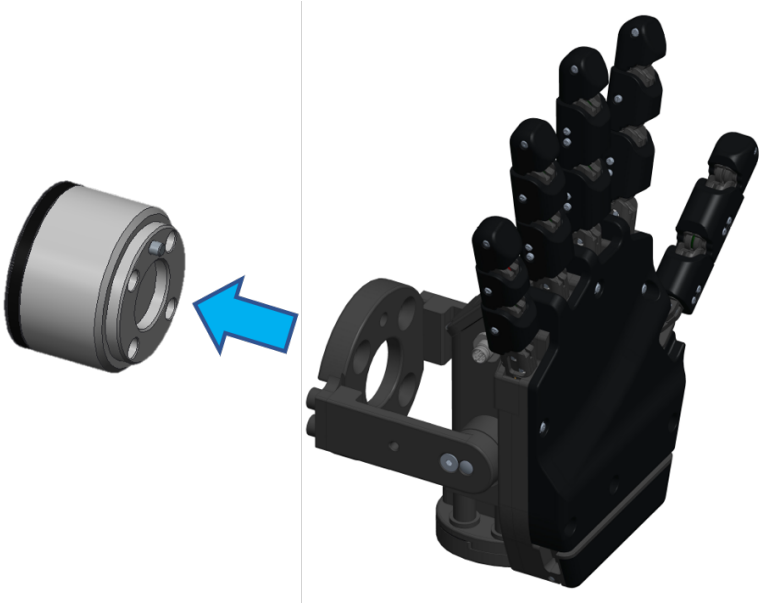
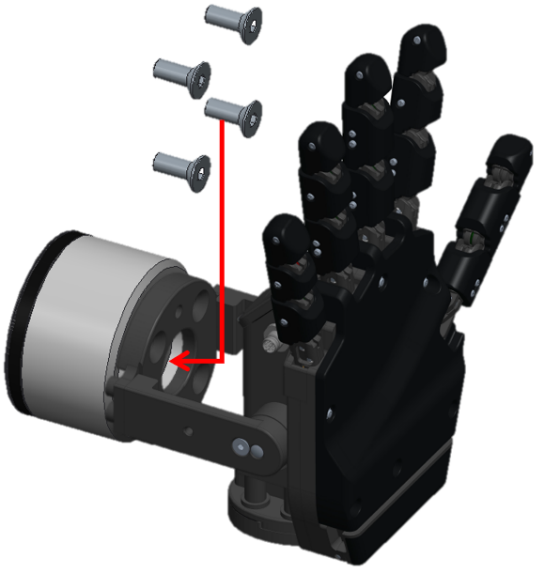
5.1 Tool mounting

5.1.1 SoftHand Industry mounting

Use only the screws provided within the package. Longer screws could damage the robot or the hand.

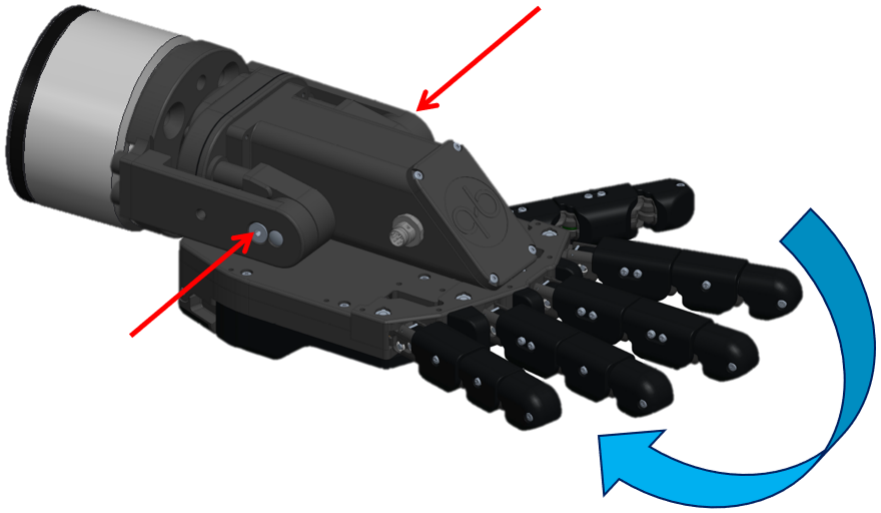
To assemble the hand on the robot arm, please follow the instructions of following table:

#	INSTRUCTIONS
1	Unlock the wrist removing the 2 screws by 2.5mm allen hex key and rotate the flange. 
2	Keep the device as shown in figure, then replace the 2 locking screws. 
3	Insert into the 6 mm hole on the wrist of the robot. 

#	INSTRUCTIONS
4	Center the qb SoftHand Industry on the robot’s flange. 
5	Fasten the device to the robot's wrist by tightening the four screws. 

You need the 4mm Allen wrench.

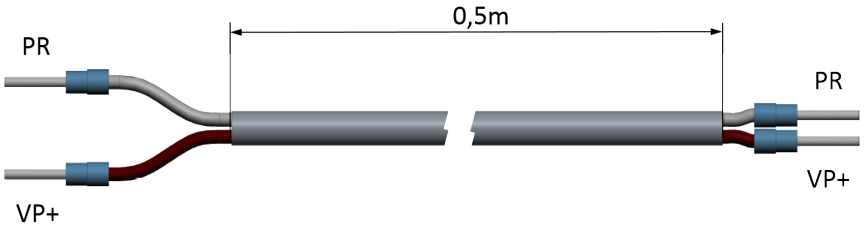
#	INSTRUCTIONS
6	Unlock the wrist (see point 1) and rotate the device until the desired position (0°, 45° or 90°). Fix the device by tightening the 2 locking screws.



5.2 Cable connections

5.2.1 Set of cables

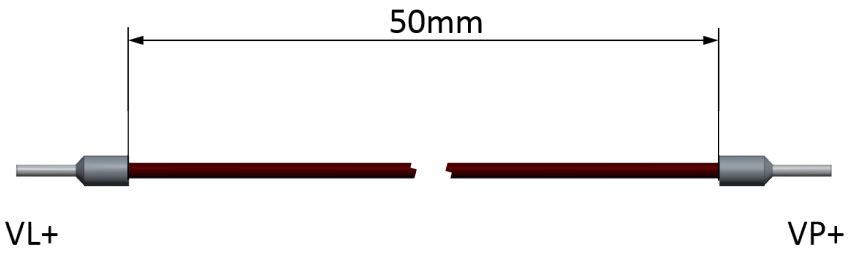
ID	Cable Name
C1	Power cable **EC_CBL_22_02_L50_RR_200**



Power cable to supply the external driver.

C2 Power cable

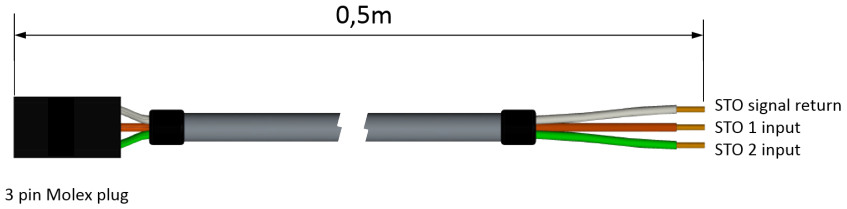
EC_CBL_22_01_L5_RR_100



Power cable to supply auxiliary port.

C3 STO cable

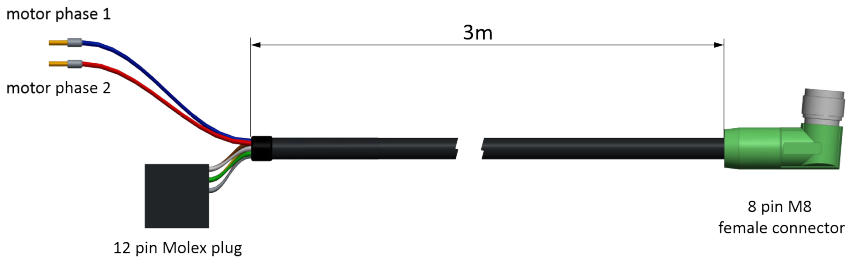
EC_CBL_26_CRMP3_WIRE_050



This cable has a free end to allow the user to connect to a generic terminal block.

C4 Main cable

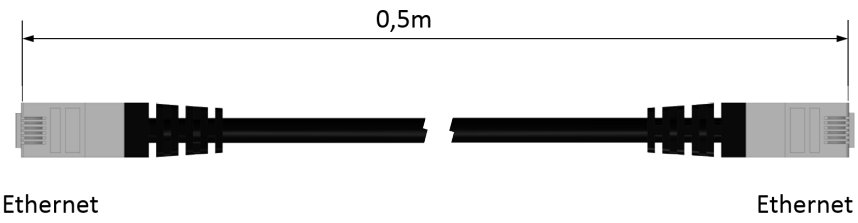
EC_CBL_26_M8FA8_CRIMP_100



Main cable to connect the Device to the external driver.

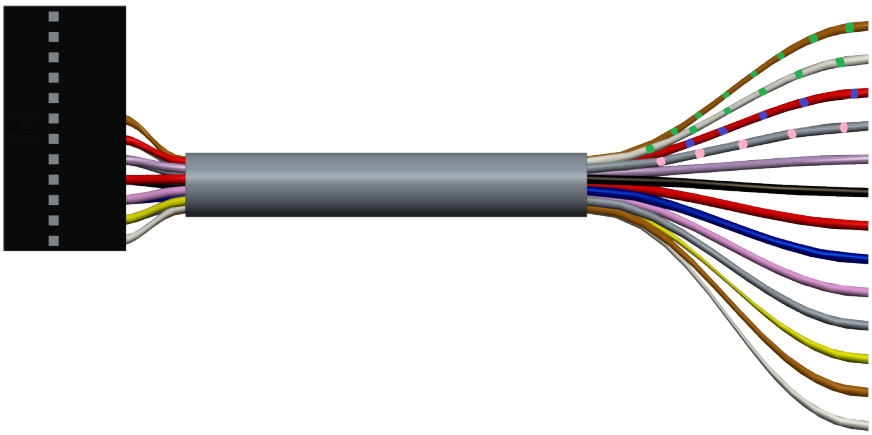
C5 Ethernet cable

EC_CBL_UTP_6_RJ45_BLK_050



C6 Digital I/O cable

EC_CBL_26_CRMP24_WIRE_050

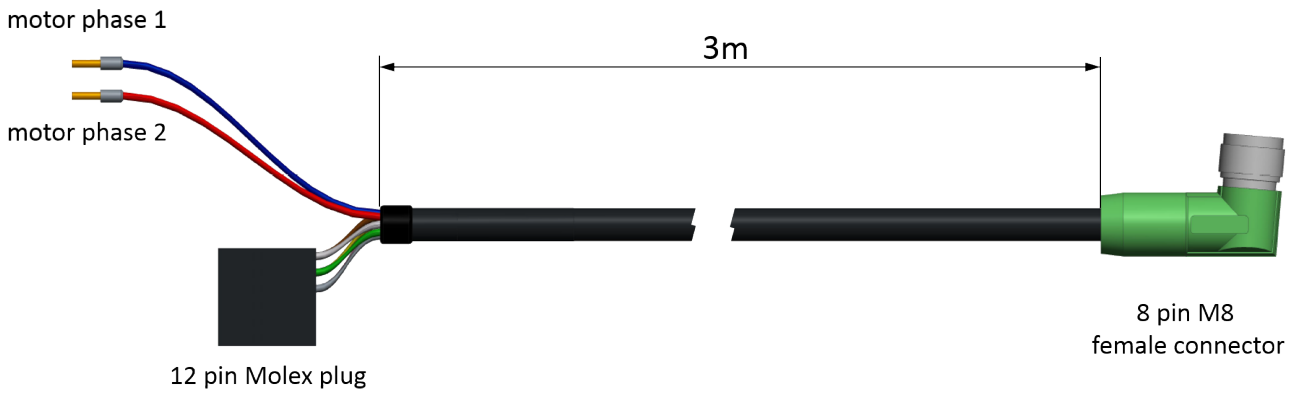


C7 Mini-B USB cable

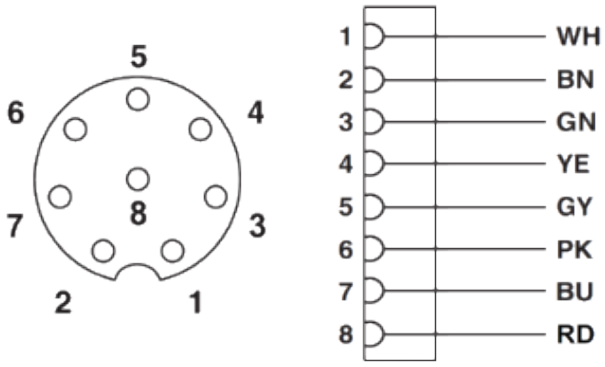
EC_CBL_USB2_A_MINI_100



M8 connector pinning

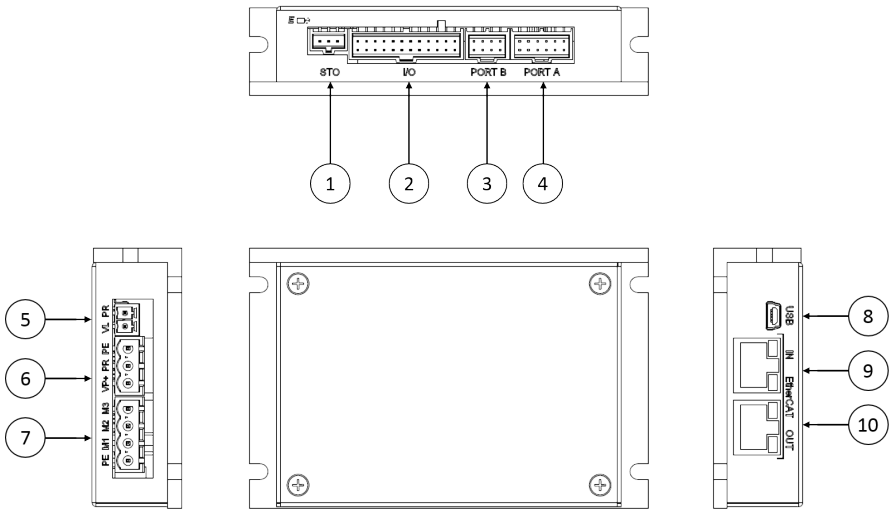


The pinning of the female M8 connector, 8 poles, is given below:



Pin n.	Wire color	Purpose
1	White	SSI 5 VDC
2	Brown	SSI GND
3	Green	SSI clock +
4	Yellow	SSI clock -
5	Gray	SSI data +
6	Pink	SSI data -
7	Blue	motor phase 1
8	Red	motor phase 2

Driver ports



Port n.	Type	Function	Pins

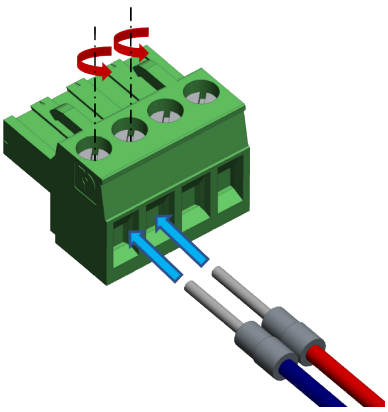
1	2.54 mm Pitch Molex	STO	3
2	2.54 mm Pitch Molex	Feedback port C and I/O	24
3	2.54 mm Pitch Molex	Feedback port B	8
4	2.54 mm Pitch Molex	Feedback port A	12
5	Phoenix 3.81 mm Pitch 'HC'	Auxiliary supply input	2
6	Phoenix 5 mm Pitch 'HC'	Main Power	3
7	Phoenix 5 mm Pitch 'HC'	Motor phases	4
8	USB Device Mini-B	USB	5
9	EtherCAT in	RJ-45	8
10	EtherCAT out	RJ-45	8

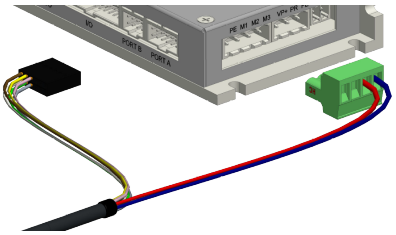
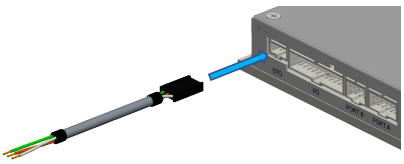
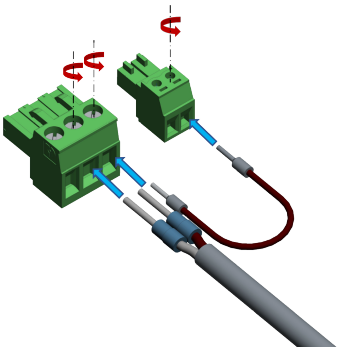
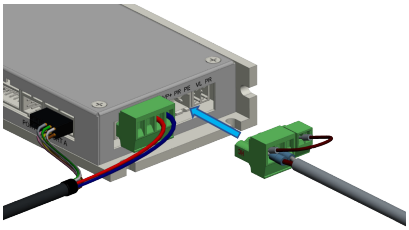
5.2.2 Cabling sequence

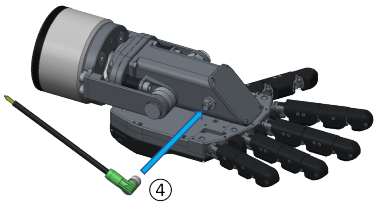
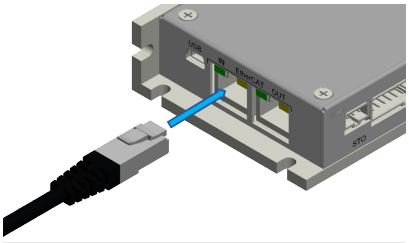
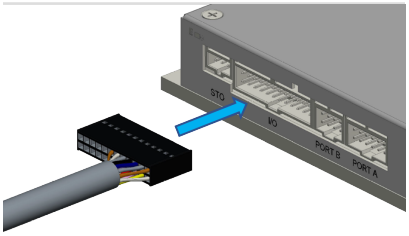
WARNING
Do not use force on the cables to remove the wiring. Act only on the connectors. Use only the provided set of cables.

WARNING
The entire wiring process should be performed without powering the external driver. Be sure to complete all steps before powering the system!

#	INSTRUCTIONS
1	Assembly the motor phases of the main cable (**C4**) to the 4-pin connector for the **port** 7 of the driver; block the cables turning clockwise the two screws. Make sure to connect the red cable to the **M2** pin and the blue cable to the **M3** pin. Pay attention to connect cables in the correct way. You need the flathead screwdriver.



#	INSTRUCTIONS	
2	Assembly the main cable (**C4**) to the external driver. Connect the 12-pins Molex plug to the **port 4** and the 4-pins connector to the **port** 7.	
3	Connect the STO plug (**C3**) to the external driver **port 1.** Read Safe Torque Off function (see Page 21) section for more information.	
4	Assembly the power cables (**C1**) and (**C2**) to the 3-pins connector for the **port 6** of the external driver and the 2-pin connector for the **port 5** of the external driver; block the cables turning clockwise the three screws. Make sure to connect the two brown cables to the **VP+** pin, the white cable to the **PR** pin and the other end of brown cable to the **VL+** pin. ⚠ Pay attention to connect cables in the correct way. ℹ You need the flathead screwdriver.	
5	Connect the power cable to the **port 5** of the external driver, by the 2-pin connector and the power cable to the **port 6** of the external driver, by the 3-pin connector. ✖ Do not power the device!	

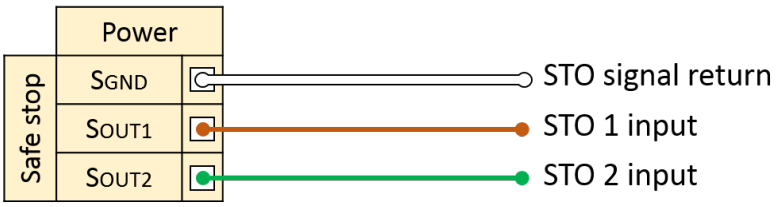
#	INSTRUCTIONS	
6	Insert the M8 connector of the main cable (**C4**)into the receptacle on the hand. Lock the M8 connector by manually tightening its threaded ring on the body of the receptacle (max tightening torque= 0.2 Nm). ⚠ Ensure that the device is not powered until this step is completed! ⚠ Be careful to insert the connector in the right direction; Do not rotate the M8 connector when inserted into the receptacle; Minimum curve radius of the cable (4) = 51mm.	
7	Depending on which control mode is desired for the qb SoftHand Industry, connect: • or the Ethernet **EC_CBL_UTP_6_RJ45_BLK_050** (**C5**) cable to the **port 9** of the external driver; • or the Digital I/Os cable **EC_CBL_26_CRMP24_WIRE_050** (**C6**) to the **port 2** of the external driver. • or the Mini-B USB cable **EC_CBL_USB2_A_MINI_100** (**C7**) to the **port 8** of the external driver. ℹ For more information, read carefully the dedicated sections of the Device Control chapter. (see Page 23)	 

Safe Torque Off (STO) function

Activation of the Safe Torque Off (STO) function causes the drive to stop providing power, which controls the motor movement and the motor can only be electrically activated, when both STO inputs are active.
The STO may be used to prevent unexpected motor rotation while the drive remains connected to a power supply.
Whenever one of the inputs is no longer active, power that can generate motion is no longer provided to the motor by the drive.

The STO cable must be connected to a safety device and you can connect it in two ways:

Connection to Robot control box. In this way the STO is managed by the control box and its safety protocols.



For connection to the control box, please refer to the manual of the robot you are using.

External power connection. In this way you can keep the motor enabled and you have to turn off the external power source to disable it.



6 Device Control

6.1 Digital I/O

It is possible to control the qb SoftHand Industry via 24VDC digital inputs and outputs.

Control of the device via digital I/O requires the connection between the Robot Controller/PLC and the external driver via the supplied STO **EC_CBL_26_CRMP24_WIRE_050 (C6)** cable.

The **EC_CBL_26_CRMP24_WIRE_050** cable has a free end to allow the user to connect to a generic terminal block. In the other end there is a Molex 24 pins connector, for the external driver's **port 2**.

Below is the colour legend and pin numbering on port 2 of the external driver.

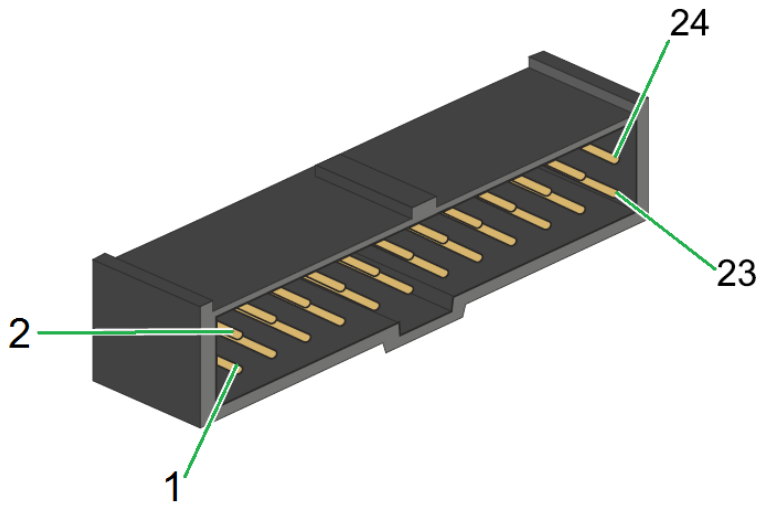


Figure 2. — Digital I/Os port of provided driver.

WARNING:
Pins with an odd number in the bottom row and an even number in the top row.

12	IN_GND
13	IN 1
14	IN 2
15	IN 3
16	IN 4
17	IN 5
18	IN 6
19	OUT 4
20	OUT 3
21	OUT 2
22	OUT 1
23	OUT_VDD
24	OUT_GND

The following table shows the actions related to the trigger of the DIGITAL INPUTS.

Digital Input Port	Triggered Action

IN1	fully opening at maximum speed
IN2	fully closing at maximum speed and force
IN3	fully closing at 25% of speed and maximum force
IN4	fully closing at 25% of speed and maximum force
IN5	fully closing at 25% of speed and 75% of force
IN6	correct termination procedure to use before switching off
IN_GND	digital inputs ground

Table 1. — Digital Inputs

WARNING:
the termination procedure (**IN6**) is called for every switching off and it's needed for the correct behavior of device and to limiting its hysteresis.

With regard to the DIGITAL OUTPUTS of the external driver, please refer to the table below.

Digital Output Port	Feedback
OUT1	**0** (0VDC) when the motor is stationary
OUT2	**0** (0VDC) when the number of cycles exceeds the maintenance threshold
OUT3	**0** (0VDC) when there is an error in the program or in the driver parameters
OUT4	**0** (0VDC) when the driver is in protection for some electrical or mechanical problem
OUT_VDD and OUT_GND	references needed to define the 24VDC. Connect OUT_VDD to 24V and OUT_GND to its ground reference (0V)

Table 2. — Digital Outputs

6.1.1 Example of control with Digital I/O

Using just two digital outputs DO1 and DO2 (at 24VDC) on the Robot Controller/PLC, it is possible to use the qb SoftHand Industry by activating the logic states of the two outputs.

For example, switching DO1 to high value, the device will open completely, while setting DO2 it will have a complete closure (with maximum speed and force). This is because the called functions are respectively the first and second digital inputs of the external driver of the qb SoftHand Industry.

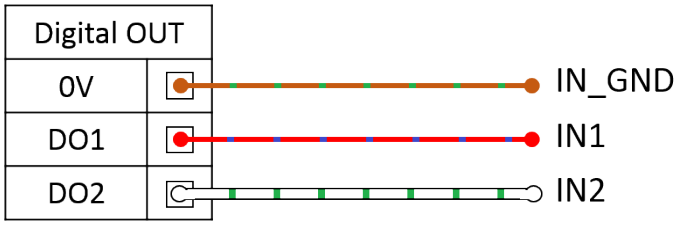


Figure 3. — Example of control via a generic PLC control or Robot Controller

For all other available configurations, please refer to Digital Inputs table (see Table 1) .

In a similar way it is possible to retrieve the status of the device and some auxiliary information through its digital outputs. For example, DI1 will be low whenever the qb SoftHand Industry is in a steady state, whether it is related to a socket or a rest condition (it will be at 1 when the motor will be moving).

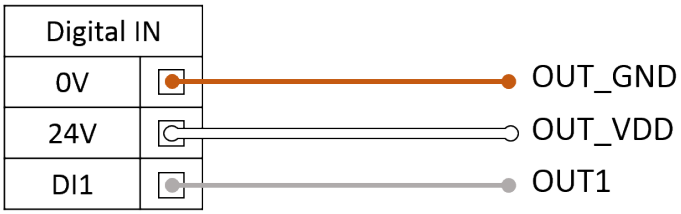


Figure 4. — Example of status reading via a generic PLC control or Robot Controller

For all other states available, please refer to Digital Outputs table (see Table 2) .

> All these states are activated by a pulse of a few tenths of a second on the respective 24VDC digital input (with respect to the IN_GND which acts as a ground for digital inputs). If the input remains high, the others can still be activated, but the high one will not activate until it returns to "0" (0 VDC) and then again to "1" (24 VDC). It is not possible to activate multiple commands simultaneously. To give the next input, the previous one must already have been completed.

6.2 C++ API

C++ libraries are provided by qbrobotics in the form of precompiled packages(.so, .dll, ...).

> Control of the device via C++ API exploits Ethernet UDP connection and requires the connection between the PC and the external driver via the supplied Ethernet **EC_CBL_UTP_6_RJ45_BLK_050 (C5)** cable.

It is possible to download these libreries from qbrobotics website[1]. They are provided within a CMake project so that,following the very basic installation steps, it is possible to build the example code from scratch and test the qb SoftHand Industry functionalities. For example, on a Linux system the commands to be executed on the terminal are:

```
mkdir build
cd build
```

1. https://qbrobotics.com/wp-content/uploads/2021/07/qbsofthand_industry_api_1.0.3.zip

```
cmake ..
make
```

The example code is stored under the src directory (*main.cpp*). The building process generates an executable called *qbsofthand_industry_api_example* in the build directory.

WARNING:
Before executing the example code, please be sure to have properly connected and powered the device by following the instructions described in the previous chapters of this documentation.

The example prompts on screen the qb SoftHand Industry information and close (and reopen) it just one time.

WARNING:
Note that depending on the operating system you are using, specific libraries must be linked. If you need special architecture binaries, please contact our Support Team[2].

It is required to be connected to the device through a proper Ethernet network. The device default parameters are the followings so please set up your network adapter settings to fit this configuration.

- Network IPv4: 192.168.1.110
- Network Mask: 255.255.255.0
- Network Gateway: 192.168.1.1
- Network DHCP: disabled

For example, you may give the *192.168.1.100* static IPv4 to your network adapter.

It is possible to change the device network parameters later, using a dedicated API method.

6.2.1 Class methods description

qbSoftHandIndustryAPI()

Initialize the qb SoftHand Industry API handler by opening the proper UDP socket and by testing its communication. The constructor tries to connect to the qb SoftHand Industry for at most 30 seconds and then returns. Through the method 'isInitialized()' is possible to test whether the initialization succeeded or not. The default IPv4 address used for the UDP socket connection is 192.168.1.110.

2. mailto:support@qbrobotics.com

qbSoftHandIndustryAPI(const std::string &device_ip)

Initialize the qb SoftHand Industry API handler by opening the proper UDP socket and by testing its communication. The constructor tries to connect to the qb SoftHand Industry for at most 30 seconds and then returns. Through the method 'isInitialized()' is possible to test whether the initialization succeeded or not.

Arguments:

- *device_ip*: The IPv4 address used for the UDP socket connection. It must be a valid IPv4 address.

~qbSoftHandIndustryAPI()

Close the UDP socket and perform the proper shutdown procedures.

float getCurrent()

Get the actual qb SoftHand Industry motor torque in percent value w.r.t. the maximum value.

float getPosition()

Get the actual qb SoftHand Industry motor position in percent value w.r.t. the maximum value.

float getVelocity()

Get the actual qb SoftHand Industry motor velocity in percent value w.r.t. the maximum value.

std::string getStatistics()

Get the qb SoftHand Industry device information.

bool isInitialized()

Return true if the initialization procedure has succeeded.

int setClosure(const float &position)

Send the given percent-position closure command reference to the qb SoftHand Industry.

Arguments:

- *Position*: the percent-position command reference, in range [0, 100] % where `0` is the fully open configuration, and `100` is the fully closed. return `0` on success; `-1` if the position is out of range.

int setClosure(const float &position, const float &velocity, const float &current)

Send the given percent-position closure command reference to the qb SoftHand Industry, together with the speed reference command and the maximum force that should be applied during the grasp. This is the most complete command to send a reference to the qb SoftHand and should be used for special cases. In normal usage, the simpler version above should be preferred. All the parameters are expressed in percent w.r.t. the maximum possible value.

Arguments:

- *Position*: the percent-position command reference, in range [0, 100] % where `0` is the fully open configuration, and `100` is the fully closed.
- *Velocity*: the speed command reference, in range [12.5, 100] % where `12.5` is the minimum velocity of the closure, and `100` is full speed.
- *Current*: the current threshold for the motor, in range [62.5, 100] % where `62.5` is the minimum force that the hand can apply, and `100` is the maximum return `0` on success; `-1` if at least one of the percent values is out of range.

int setIP(const std::string &net_ip, const std::string &net_mask, const std::string &net_gateway)

Change the qb SoftHand Industry IPv4 address, network mask and gateway of the device. It is worth noticing that the device should be connected first to use this command.

Arguments:

- *net_ip*: the device network IP address, e.g. 192.168.1.110.
- *net_mask*: the device network mask, e.g. 255.255.255.0.
- *net_gateway*: the device network gateway, e.g. 192.168.1.1.

return* **0** on success; **-1** if at least one of the given IPv4-format addresses is wrong.

void waitForTargetReached()

Wait until the qb SoftHand Industry has reached the commanded position. This method should be called after a *setClosure* one, to wait for the action to be completed.

6.3 ROS

The *qb SoftHand Industry* can be fully controlled trough Ethernet with ROS Noetic.

6.3.1 Installation

Requirements

If you have never set it up, you probably need to add your Linux user to the `dialout` group to grant right access to the serial port resources. To do so, just open a terminal and execute the following command:

```
sudo gpasswd -a <user_name> dialout
```

where you need to replace the `<user_name>` with your current Linux username.

Note: don't forget to logout or reboot.

Ubuntu Packages

If you prefer to leave your catkin workspace as it is, you can simply install all the ROS packages from the Ubuntu official repositories:

```
sudo apt update
sudo apt install ros-noetic-qb-softhand-industry
```

Sources

Since you are interested in the ROS interfaces for our devices, it is assumed that you are familiar at least with the very basics of the ROS environment. If not, it might be useful to spend some of your time with ROS[3] and catkin[4] tutorials.

Install the *qb SoftHand Industry* packages for a ROS user is straightforward. Nonetheless the following are the detailed steps which should be easy to understand even for ROS beginners:

1. Clone the `qb_softhand_industry` packages to your Catkin Workspace, e.g. `~/catkin_ws`:

   ```
   cd `~/catkin_ws/src`
   git clone <https://bitbucket.org/qbrobotics/qbshin-ros.git>
   ```

2. Install ROS dependencies using `rosdep`:

   ```
   cd `~/catkin_ws`
   rosdep install --from-paths src --ignore-src -r -y
   ```

3. Compile the packages using `catkin`:

3. http://wiki.ros.org/ROS/Tutorials
4. http://wiki.ros.org/catkin/Tutorials

```
catkin_make
```

Note: depending on your ROS installation, you may need some extra packages to properly compile the code. Please, be sure that you have already installed at least `ros-noetic-ros-controllers`, `ros-noetic-transmission-interface`, `ros-noetic-joint-limits-interface`, `ros-noetic-combined-robot-hw`, and their dependencies (*e.g. use* `sudo apt install <ros-pkg>`).

If you were not familiar with ROS you should be happy now: everything is done! Nonetheless, if you encounter some troubles during the compilation, feel free to ask for support on our Bitbucket[5].

Device Setup

To connect a *qb SoftHand Industry* to your system is **read carefully** the Mounting and Wiring Chapter (see Page 12) to understand all the requirements and advices. However the communication happens via Ethernet, (default address 192.168.1.110) so you have to configure your wired connection with an IP of the same family.

6.3.2 Usage

Launch file

Once the device is physically connected to your system, you can launch the device node with the following command line:

```
roslaunch qb_softhand_industry_control shin_control.launch use_rviz:=true
 use_controller_gui:=true
```

The arguments explained

- `use_controller_gui [false]` : Choose whether or not to use the controller GUI.

Additional arguments

- `control_duration [0.001]` : The duration of the control loop expressed in seconds.
- `ip_address [192.168.1.110]` : The IP address of the ELMO driver.
- `flange_angle [0]` : The angle of the configurable flange (in radians).
- `max_repeats [3]` : The maximum number of consecutive repetitions to mark retrieved data as corrupted.

5. https://bitbucket.org/account/user/qbrobotics/projects/ROS

The followings are particular control modes which are enabled with few parameters, but the concepts of this paragraph hold for all of them.

GUI Control

This control mode is the one suggested to test that everything is working as expected. You are able to open and close the *qb SoftHand Industry* interactively, but nothing more than this.

> You will probably need this only the very first times and for debugging.

To start this mode just add `use_controller_gui:=true` to the general `roslaunch` command.

After a while a GUI should appear to screen with two empty drop down menus, a red enable button below them, and a *speed scaling* slider at the bottom.

1. Select the *Controller Manager* namespace from the left menu, e.g. `/<robot_namespace>/control/controller_manager` (where `<robot_namespace>` is an additional argument of the launch file needed with several devices). This enables the right menu which provides all the controllers available for the connected device.
2. Select the *shin synergy trajectory controller* from the second drop down menu and enable it through the circular button.
3. A single slider will appear in the GUI to control the closure of the hand, which ranges from `0` (hand completely open) to `1` (hand completely closed). Move the slider to perform a complete grasp or to partially open/close the *qb SoftHand Industry*. You can also vary the speed through the bottom *speed scaling* slider if you like a faster/slower motion. No other timing constraints can be set in this mode.

6.4 URCap

> Control of the device via C++ API exploits Ethernet UDP connection and requires the connection between the Universal Robots Control Box and the external driver via the supplied Ethernet **EC_CBL_UTP_6_RJ45_BLK_050 (C5)** cable.

6.4.1 Installation

It is possible to find the most up-to-date installation process in the official robot manual provided by Universal Robots.

6.4.2 URCap Overview

The easiest solution to control the qb SoftHand from the Universal Robots ecosystem is to exploit the URCap capabilities provided described below. The images used in this chapter refer to the e-Series Polyscope 5+, but a very similar interface is available in Polyscope 3+ for CB-Series robots.

Installation Node

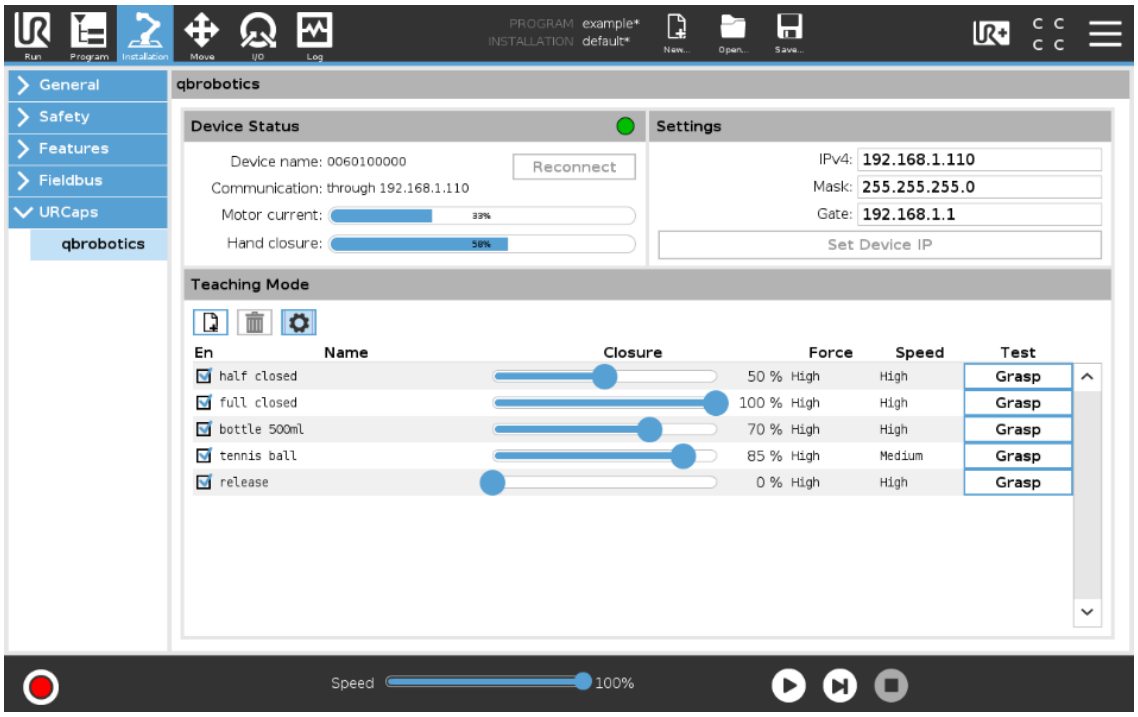


Figure 5. — URCap installation interface

The qbrobotics Installation tab is the place where all the information about the qb SoftHand state and its stored settings are displayed all together.

Device Status Panel

The Device Status panel shows a led icon on its top right corner to quickly identify the connection state:

- Red is for communication fault or non-connected device;
- Green, when everything looks good.

If there is a connection problem, the "Reconnect" button is enabled to manually reset the communication to the robot controller.

In the same panel are shown the identification name of the device and the IP address on which it is connected, e.g. “192.168.1.110”. Below, the actual motor current in percent value w.r.t. to maximum one and the actual hand closure state in percent value. If a program is running while visiting the Installation Node, you can see these status bar varying accordingly to the hand motion.

Teaching Mode Panel

To simplify the usability and re-usability of qb SoftHand Industry, it is possible to teach and store a given grasp value to recall it whenever you need it in a given Program Node.

For each, you can set the grasp command value through a percent position slider, and you can test it buy pressing the "Grasp" button on the right of the chosen line. If you are not happy with the given closure, just iterate this procedure until you are fine with the teached grasp. All of this named grasp value can be recalled by every Program Node at any time.

If a stored grasp previously set is modified at a given time, all its occurrences used in any program are modified accordingly to the new value set. This is possible thanks to a grasp database saved in the current Installation loaded.

> **WARNING**:
> If you change or remove an Installation file on your robot you may not be able to exploit this linking capability any longer because the stored grasp database no longer exists. On the other hand, each saved program still behaves the same as before because each qb SoftHand Program Node stores the grasp value also inside the program data.

In addition it is possible to add or remove a grasp value, which is visualized in a tabled fashion.

Program Node

This Node aimed to send a command to the qb SoftHand at a given time in a robot program.

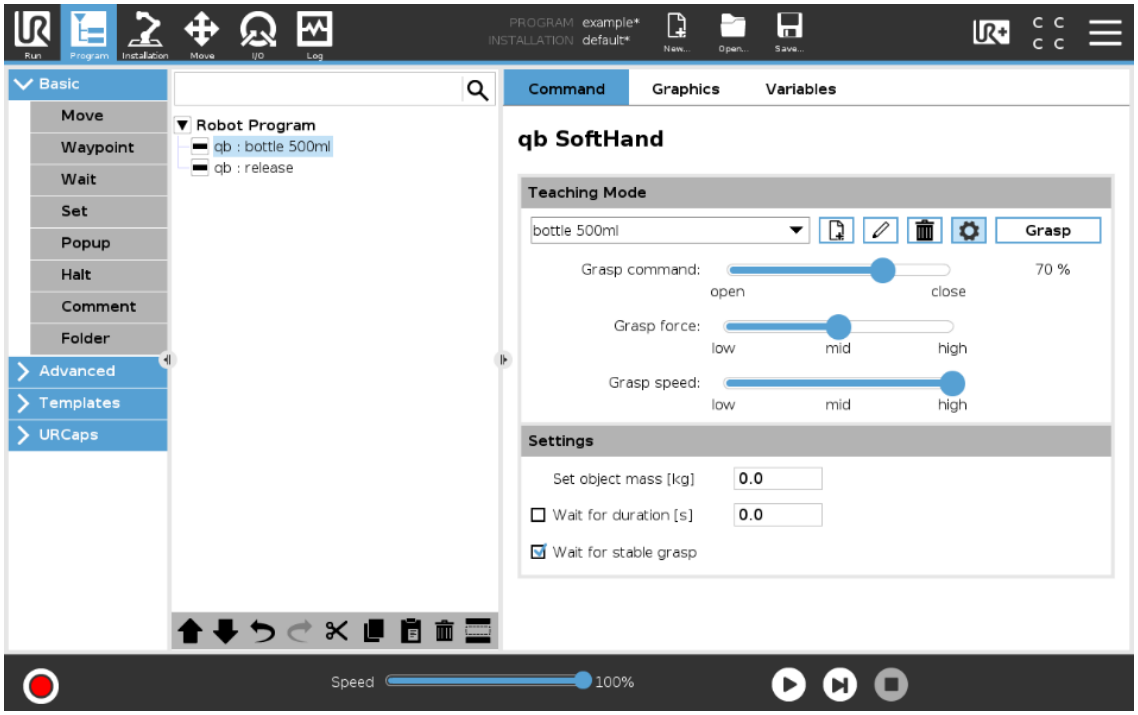


Figure 6. — URCap programming interface

Teaching Mode Panel

This panel is strictly linked to the one in the Installation Node since you can recall, add, modify or remove a stored grasp value from the internal database. The values shown in the slider are the ones sent to the qb SoftHand Industry while the program is running.

If you have selected a stored grasp and change its command value through the slider, you change not only the current Program Node value, but every linked grasp with the same name.

While creating a robot program, you can check if the grasp command is suitable by testing it through the *Grasp* button on the top right of the panel.

Settings Panel

If you are grasping an object with a known mass, you can add it in the proper field to automatically sum it up with the qb SoftHand one. Remember to set it to 0 when you release the object. Additionally, you can specify if this Node has to add a wait during the grasp or not. You can choose one of the two predefined options:

- Wait for a given amount of time expressed in seconds;
- Wait until the grasp is stable.

If none are satisfying for the given application, feel free to add a Wait Node after the qb SoftHand one. If both the options are selected, they are evaluated in sequence: first it waits for the specified custom time and then checks whether the grasp is already stable or not.

e-series Toolbar

This toolbar lets you quickly check the qb SoftHand state and tech new grasp values. The two panels displayed are almost identical to the ones described in the Installation and Program Nodes.

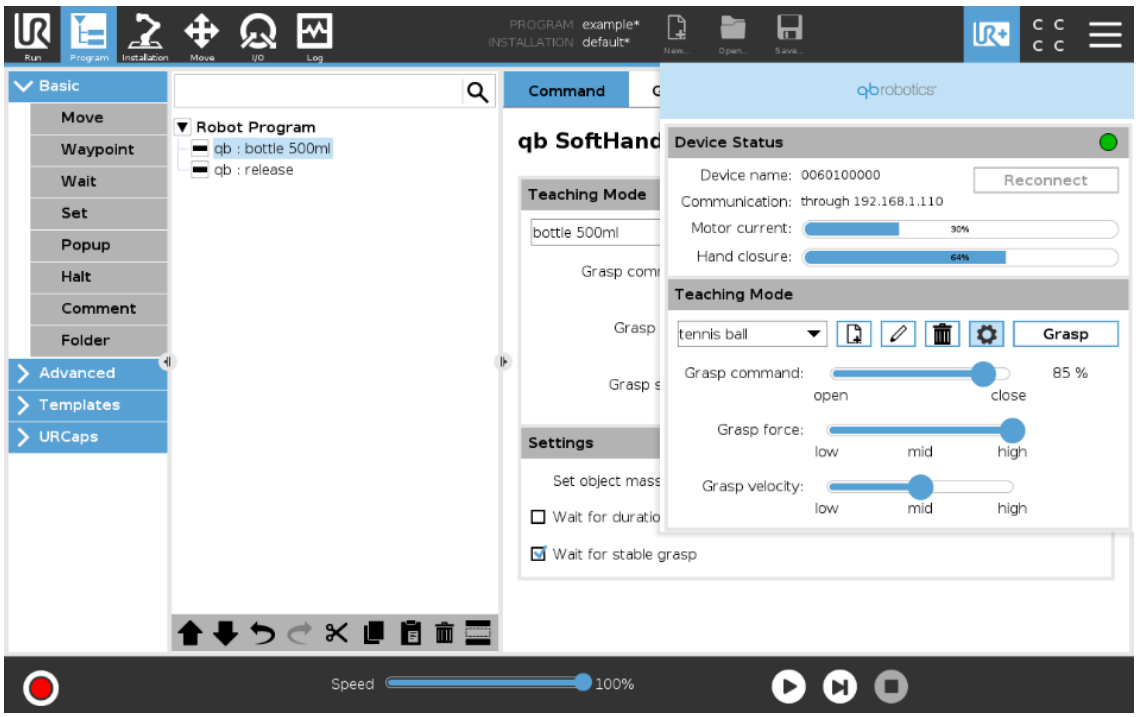


Figure 7. — URCap toolbar

Program Example

The following example is a simple pick-and-place robot program where the qb SoftHand is used as grasping tool.

The example should be self-explainatory, but it is worth noticing that in many cases a "pre-grasp" can help to accomplish a task in a better fashion. This is strictly related to the specific task, but you can add as many qb SoftHand Program Nodes as you need to let the hand pass in an even narrow spot. For example, you may need a pre-grasp to be able to open a door through its handle.

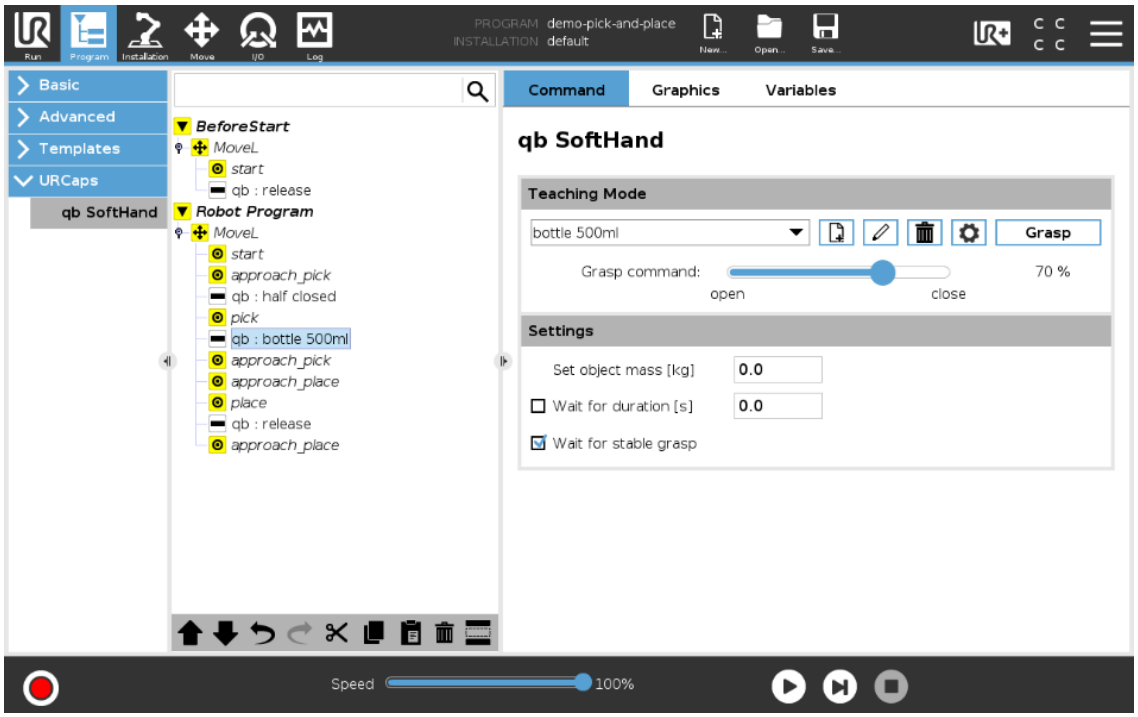


Figure 8. — URCap programming example

URCap enabled script functions

> **WARNING:**
> These commands are meant to be used by expert users only.

Apart from the UI features described above, you can add some custom function in your robot program by using our script methods:

- `bool qbsofthand.industry_connect(string ip_address)` : opens the UDP socket on the given IPv4 address and tests its communication.
- `void qbsofthand.industry_disconnect()` : closes the UDP socket and performs the proper shutdown procedures.
- `int qbsofthand.industry_getCurrent()` : returns actual motor torque in percent value w.r.t. the maximum value.
- `int qbsofthand.industry_getPosition()` : returns actual motor position in percent value w.r.t. the maximum value.
- `string qbsofthand.industry_getStatistics()` : returns details about device internal parameters and it is mainly aimed to our support team during incident investigation.
- `bool qbsofthand.industry_isReachable()` : returns "true" if the socket communication is active and responsive.
- `void qbsofthand.industry_setCommand(int position)` : sends a reference position command to the device; the integer parameter must be a percent value, i.e. in range 0 — 100.
- `void qbsofthand.industry_setCommandFull(int position, int speed, int force)` : sends a reference position command to the device, together with a speed reference and a force

threshold; all the integer parameters must be percent values, respectively in range 0 — 100, 12.5 — 100, 62.5 — 100.

- `void qbsofthand.industry_setDeviceIP(string net_ip, string net_mask, string net_gateway)` : changes the IPv4 address, network mask and gateway of the device; it is worth noticing that the device should be power off after this command.
- `void qbsofthand.industry_waitForTargetReached()` : waits until the hand has reached the commanded position; this method should be called after a 'setCommand' one, to wait for the action to be completed.

7 Maintenance and warranty

7.1 General

Products of the company qbrobotics s.r.l. are produced using the most modern production methods and are subject of strict quality inspections. All information regarding our Warranty Policy can be found at this[6] site.

6. https://qbrobotics.com/technical-support/

8 Certifications

Standards applied under development of the product is listed in this section. When an EU Directive number is noted in brackets it indicates that the standard is harmonized under that Directive.
ISO 12100:2010
EN ISO 12100:2010 (E) [2006/42/EC]
Safety of machinery – General principles for design – Risk assessment and risk reduction.
The product is evaluated according to the principles of these standards.
ISO 10218-2:2011
EN ISO 10218-2:2011(E) [2006/42/EC]
Robots and robotic devices – Safety requirements for industrial robots Part 2: Robot systems and integration
The product is prepared for compliance with robot system requirements defined in these standards.
ISO/TS 15066:2016
RIA TR R15.606
Robots and robotic devices – Safety requirements for industrial robots – Collaborative operation This is a Technical Specification (TS), **not** a standard. The product is prepared for easy integration in compliance with provisions in this Technical Specification, see more in the safety chapter.
EN ISO13849-1:2015
EN ISO13849-2:2012
Part 1: Safety of machinery. Safety-related parts of control systems. General principles for design.
Part 2: Safety of machinery - Safety-related parts of control systems - Part 2: Validation
The product is evaluated according to these standards.

ISO/TR 20218-1:2018
Robotics – Safety requirements for industry robots Part 1: Industrial robot system end of arm tooling (end-effector)
This is a Technical Report (TR), **not** a standard. The product is designed according to principles in this Technical Report.
ISO 9409-1:2004 [Type 50-4-M6]
Manipulating industrial robots – Mechanical interfaces
The EE flange conforms to type 50-4-M6 of this standard. Robots should also be constructed according to this standard to ensure proper fitting.
EN 60529/A2:2013
Degrees of protection provided by enclosures (IP Code)
This standard defines enclosure ratings regarding protection against dust and water. SoftHand Industry is designed and classified with an IP rating according to this standard.
EN 61000-6-4:2019 "Electromagnetic compatibility (EMC) – Part 6-4 Generic standards – Emissions for industrial environments"
EN 61000-6-2:2019 "Electromagnetic compatibility (EMC) – Part 6-2 Generic standards – Immunity for industrial environments".
EN 61000-4-2:2009 " Electromagnetic compatibility (EMC) – Part 4-2: Testing and measurements techniques – Electrostatic discharge immunity test".
EN 61000-4-3:2006/A1:2008/A2:2010 " Electromagnetic compatibility (EMC) – Part 4-3: Testing and measurements techniques – Radiated, radio-frequency, electromagnetic field immunity test".
EN 61000-4-4:2012 " EN 61000-4-8:2010 " Electromagnetic compatibility (EMC) – Part 4-4: Testing and measurements techniques – Electrical fast transient/burst immunity test".
EN 61000-4-5:2014/A1:2017 " EN 61000-4-8:2010 " Electromagnetic compatibility (EMC) – Part 4-5: Testing and measurements techniques – Surge immunity test".

EN 61000-4-6:2014 " EN 61000-4-5:2014/A1:2017 " EN 61000-4-8:2010 " Electromagnetic compatibility (EMC) – Part 4-6: Testing and measurements techniques – Immunity to conducted disturbances, induced by radiofrequency fields".
EN 61000-4-8:2010 " Electromagnetic compatibility (EMC) – Part 4-8: Testing and measurements techniques – Power frequency magnetic field immunity test".
EN 61000-4-11:2004/A1:2017 " Electromagnetic compatibility (EMC) – Part 4-11: Testing and measurements techniques – Voltage dips, short interruptions and voltage variations immunity test".

9 FAQ and troubleshooting

For any issue not listed below, please contact our support team[7].

9.1 PC drivers and requirements

9.1.1 OS requirements

The minimum hardware requirements ensure basic compatibility with the supported operating systems, while the recommended requirements provide a smoother and more stable experience, especially when running Windows 10.

A computer with a dual-core processor and at least 4 GB of RAM is sufficient for the necessary operations, but for better performance a quad-core processor with at least 8 GB of RAM is recommended. Using an SSD drive, although not mandatory, significantly improves operating system responsiveness and loading times.

In terms of connectivity, at least one USB port is required for FTDI drivers and a network connection (wired or wireless). A Full HD monitor is also recommended for improved readability of graphical interfaces.

All the Long-Term Support (LTS) releases of the following OS that have not reached the End-Of-Life (EOL) term are officially supported.

Component	Minimum	Recommended
CPU	Dual-core ≥ 2.0 GHz (Intel i3 4th gen or AMD eq.)	Quad-core ≥ 2.5 GHz (Intel i5/i7 6th gen or AMD eq.)
RAM	4 GB	8 GB or more
Storage	64 GB free (HDD)	128 GB+ SSD
Graphics	Integrated (Intel HD Graphics or eq.)	Recent integrated or entry-level GPU (e.g. NVIDIA)
Ports	1× USB 2.0/3.0	2+ USB (preferably 3.0)
Network	Integrated Ethernet or Wi-Fi	Ethernet + Wi-Fi
Monitor	1280×720	1920×1080 (Full HD)

7. mailto:support@qbrobotics.com

WARNING:

Using virtual machines (Windows or Linux) can be considered; however, this is not recommended. Its configuration could introduce issues such as increased latency and packet loss in network communications, potential incompatibilities with hardware drivers, and issues with system time synchronization..

Linux Ubuntu

Ubuntu 16.04, 18.04 and 20.04 are supported and recommended w.r.t. the other Linux distributions.

NOTE:

Note: different Linux distributions should also work fine, but they are not officially supported.

Microsoft Windows

As far as July 2020, Windows 10 version 1809+ and Windows 10 Enterprise 2015+ are supported and recommended.

9.1.2 Drivers installation

If you are using either Microsoft Windows or Apple macOS it is necessary to download and install the communication drivers from the FTDI VCP Drivers[8] page.

Linux Ubuntu

LINUX FRIENDLY:

You do not need additional drivers if you are running on any Linux distribution.

However, if you have never set it up, you probably need to add your linux user to the `dialout` group to grant right access to the serial port resources. To do so, just open a terminal and execute the following command:

```
sudo gpasswd -a <user_name> dialout
```

where you need to replace the `<user_name>` with your current linux username.

8. https://www.ftdichip.com/Drivers/VCP.htm

NOTE:

Note: don't forget to logout or reboot.

Microsoft Windows

Install the driver for your operating system version to allow the serial port access through something like `` `COM7` ``.

WARNING:

Be aware that we support only serial port names between `` `COM1` `` and `` `COM9` ``.
Each time you connect a new device to your PC (or even a device to a new PC), you should check whether it has been set in the proper port range or not.
To change the serial port number:

1. go under `"Control Panel > Hardware and Sound > Device Manager"`;
2. explore the `"Ports (COM & LPT)"` branch, right-click on the displayed `` `COM` `` port and select `"Proprieties"`;
3. from there, select the `"Port Settings"` tab and click on the `"Advanced"` button`;`
4. lastly, choose from the drop-down menu a `` `COM` `` number between `` `COM1` `` and `` `COM9` `` and click on `"OK"`.

9.1.3 API installation requirements

Before using the qbAPI on your system, if not already installed it is good use to install a c++ compiler. Below here are listed all the necessary instructions for each operative system.

Linux Ubuntu

LINUX FRIENDLY:

You should have both `"gcc/g++"` and `"make"` already installed if you are running on any Linux distribution.

Apple macOS

Download `"XCode"` from the App Store, this will install `"gcc/g++"` and `"make"` utilities.

Microsoft Windows

Download MinGW[9] and install it. Open MinGW Installation Manager, from the left panel select basic setup, then from the right panel select mingw32-base and mingw32-gcc-g++, then click on `Installation -> Apply Changes`. This will install the *gcc/g++* compiler. To use it from the command line you need to provide to windows the binary path to the executable. Go to System Properties and click on Environment Variables. In the System Variable windows, look for `path`, select it and click `edit`. Go to the end of the Variable Value field, add a `;` separator and add the path to the binary folder for gcc (usually it is in `C:\MinGW\bin`).

Download the *make* utility from here[10]. Follow the installation instruction. In the end you will need to add the binary path to the Environment Variables. To do that follow the previous steps. (Usually the binary folder for the make utility is in `C:\Program Files (x86)\GnuWin32\bin`).

NOTE:

Note: if you have the CMD already opened when performing the installation, you probably will need to reopen a new one to be able to use the utilities.

9.2 The hand never connects to the robot system

9.2.1 Problem

The user interface says that the device is not connected even if it should be.

9.2.2 Solution

The first steps to try to solve the issue are the followings:

- Check whether the wiring is correct, cf. Mounting and Wiring (see Page 12).

Please, also have a look at each connector and check if any is damaged.
Do not force a connector plug if it seems not to match properly its location. It may be already damaged and you could brake it completely!

- Make a power cycle:
  - If the hand is connected to the power source of the robot Control Box, make a power cycle of the robot.
  - If the hand is connected to an external power source, make a power cycle of the device.

Be aware that you need to leave the system without power for at least 30 seconds during the power cycle.

9. http://www.mingw.org
10. http://gnuwin32.sourceforge.net/packages/make.htm

Only in this way we can assure that the system has completely reset its electrical state.

If these steps do not solve the problem — i.e. the device still does not connect to the robot system — please raise a support request with at least the following details at our help desk portal:

- The device serial number;
- The robot system type and software version;
- The tools you are using to control the device;
- The power supply configuration;
- Any error reported in the robot logs tab;
- Any additional info regarding what you were doing before the first failure;



Be aware that the device must not violate our warranty policy, cf. maintenance and warranty (see Page 37) in order to ensure the best support service for our customer.
Please, do not forget to mention in your request if you have made forbidden hardware/software modification to the device before/after the issue.

9.3 Error C740A0 on e-Series robots

9.3.1 Problem

If you are using an old qb SoftHand or an old URCap version it may happen that you experience the error `"!Tool!A C740A0: Hardware monitoring: Critical error"` when using the device on e-Series robots.

This is particularly annoying because the robot set its protection mode and you need to restart it every time the error triggers.



NOTE:

CB-Series are not affected by this problem.
Anyway, we recommend to update the URCap to the last version available. You can find it in our download section at the bottom of the qb SoftHand page[11].

9.3.2 Solution

First of all, check your installed URCap version and update it to the last available on our website (see the above link) if required.

This may be enough to solve the problem if you have a hardware compliant device. In such a case, please flag this post as helpful and enjoy our device from now on.

On the other hand, old devices miss a hardware fix that cause the fault in e-Series robots.

11. https://qbrobotics.com/it/download/qb-softhand-industry-downloads/

If you have updated the URCap, but the `C740A0` error still appears on your Teach Pendant, you have two possible solutions:

- Get rid of this problem using an external 24V isolated power supply — e.g. http://www.deltapsu.com/products/din-rail-power-supply/DRC-24V60W1AZ — instead of the UR Control Box.

> **WARNING:**
>
> Be aware that depending on the environment, the power supply needs also to filter out potential electrical noise.

- Raise a *"Service Request"* in our portal with at least the following details at our help desk portal:
  - The device serial number;
  - The robot system type and software version;
  - The installed qbrobotics URCap version;

  And wait for a response.
- If you are suitable for upgrading we will do our best to ensure that the assistance goes as quickly and as smoothly as possible.

9.4 Grasping with a custom finger position

9.4.1 Problem

The qb SoftHand adapts the finger position to envelop the object shape as much as possible — i.e. automagically.

However the mechanical synergy of the hand always goes from a fully open *"slap"* configuration to a fully closed *"punch"*.

If you are wondering whether it is possible to command the hand to change this behavior or not, the answer is no.

9.4.2 Solution

As stated in the description (see Page 5) chapter of the manual *"The single-motor actuation makes the hand [...] simple-to-control (one single motor requires one single control signal to close and open the whole hand)"*. Thus there is no way to control each finger independently. For such a purpose you would need several motors with the consequent increase in complexity of the control.

Anyway, since the qb SoftHand can grasp a huge variety of objects with just one synergy, we prefer to have a simple and powerful device w.r.t. a bit more versatility alongside with a complex control system behind.

Quite for sure you can find another way to perform the task you have in mind. Have a look at our suggested articles and try to exploit those tips.

If you want some help with a specific task you will need to raise a "Feasibility Request" in our portal.

9.5 Simplified CAD models of the qb SoftHand

9.5.1 Problem

Where can I find CAD models or simplified meshes of the qb SoftHand Research?

9.5.2 Solution

We provides only simplified geometries of the qb SoftHand Research that you can find at our ROS description repository[12] (measures are expressed in millimeters).

As additional information, the phalanges are made of a polyamide 6 reinforced with 30% glass fiber (discontinuously placed), with the following main properties:

- density `: 1.36 [g/cm3]`;
- tensile yield strength `: 165 [MN/m2]`;
- tensile Young modulus `: 8500 [MN/m2]`;
- percent elongation at break `: 3%`;
- resilience (Izod method) `: 125 [J/m]`;
- melting point `: 220 [°C]`;
- Poisson ratio `: N/A` (for general polyamide 6, i.e. without glass fiber, it is approximately `0.4`).

9.6 The demo is not as good as expected

9.6.1 Problem

When dealing the very first times with a soft robotics device, it quickly appears intuitive and easy to use, but it may require some times to get rid of all of its power and secrets.

If you try to make a demo and it seems that it is not amazing as you have thought before, maybe is just because you need few tips to mastering in the use of our device.

Here are some tips — in no special order — to create a better demo with our hand:

9.6.2 Tips

- If you are not enthusiast with a given grasp, try to adjust the robot pose. You'll get surprisingly different outcomes with just few changes: e.g. tilting a bit the "pitch" of the wrist would result in a more natural pose of the hand when it comes close to an object. Moreover, pay attention to the robot wrist configuration, i.e. you should often be able to choose distinct kinematic configuration when approaching an object, so try to use the one that gives you more freedom while moving during the task.

12. https://bitbucket.org/qbrobotics/qbhand-ros/src/dd75e2f70bbb470be4a2360177dbde6a7afbd638/qb_hand_description/meshes/?at=production-kinetic

- Of course you'll need some practice to get smoother behaviors, but when you have doubts just *"think human"*: reproduce the task yourself and even if you believe to know what you are doing, try to see whether your wrist is slightly tilted w.r.t. the expectation or if you approach the object a bit displaced from its center. If you place the robot hand in a very similar way, you should get a more natural outcome.
- And do this for all the motions in the given task. For example, grasping and moving a joystick involves linear displacement of the end-effector combined with a proper tilt of the hand pose. Varying the wrist angles produce a more natural behavior even if the hand seems to be located in a very similar pose. Try to believe!
- Try to not hardcode waypoints, at least at the beginning. Even if it seems a symmetrical task or whatever, just teach the robot by moving it to the pose you want, store that waypoint and goes to the next. Then iterate the process and correct the ones which are not as expected.
- Exploit the environment! One of the main reason to choose a soft robotics device relies in its adaptability. Our hand is able to pick even a paper from a table if you make human-like gestures; to do so, you need to being in contact also with the table instead of trying to avoid it as with usual industrial gripper.
- If you have trouble when choosing how much the hand needs to be closed, just go *"step-by-step"* with pre-grasp phases, like the following example:

UR program example

```
Move L:  # not Move J which is linear in joint space
 - approach_waypoint
 - hand close 40%  # this is a pre-grasp
 - pick_waypoint
 - hand close 75%  # this is the real grap on the object
   ...
```

NOTE:

Pre-grasp waypoints are very useful, especially when you need to enter in a narrow space, e.g. to grasp a door handle.
It is worth noting that you can add as many pre-grasps as you want!

- Additionally, if you want to smooth the robot motion between waypoints, use the *"Blend Radius"* option to the given ones. The more you blend, the less the end-effector reaches the teached waypoint, so be careful and do some trial and error to achieve the right behavior: start with no blend, i.e. 0, and increment the value until you are satisfied.
- Improving robot velocity to be far more impressive! When the hand is properly closed on an object, you can move the robot even quite fast without losing grip if you have set the waypoints accordingly (just avoid discontinuities and you are fine).
- Last, but not least, do not always close the hand at its maximum (`100%`): it is adaptable and will do its job in any way, but it is far better to close the hand around the minimum closure that it is suitable for the given object. This will lead to a more gentle grasp, which can be really helpful when dealing with fragile objects. For example, a bottle can be grasped with a closure of something about 60%; if it

is not full, higher values may result in an unwanted squashing of its plastic. A good approach for the very first times is to increment the closure value gradually until you are confident with our device and the given object.

WARNING:

As a rule of thumb, *"you should close the hand at the minimum value that lets you perform the task"*. Therefore go incrementally from 40-50% to 80-85% (values above 85% are to be considered only for very small objects).

9.7 The glove is deteriorated/broken

9.7.1 Problem

The glove is one of the component of the hand subject to wear and tear.

The planned maintenance already makes up for this, but it may happen that the glove simply breaks during the normal usage.

WARNING:

Be aware that you must never grasp sharp objects, nor have them close to the robot during its operation, cf. safety (see Page 2) .

A damaged glove can affect repeatability and grasping performance.

9.7.2 Solution

Qbrobotics can provide a new glove and let you proceed with its replacement, following the manual instruction.

NOTE:

Inner components of the han can be fixed by qbrobotics only, like the elastic components inside the fingers.

In such a case you should raise a *"Service Request"* in our portal as soon as you notice the problem. We will do our best to ensure that this goes as quickly and as smoothly as possible.

9.8 The hand does not reopen correctly

9.8.1 Problem

The hand does not fully open up after setting the 0% position command: i.e. some of the fingers still remain a bit closed even if the motor position is correctly commanded to 0. Some of the fingers may move a bit slower than expected. Moreover, closure and reopening cycles present the same unexpected behavior at each open.

WARNING:

Be aware that this behavior can lead to hardware malfunctions if it is not solved quickly.
If it happens too often, try to grasp the objects more gently (e.g. smaller closure values) or use another approach strategy which produce a more natural grasp.

9.8.2 Solution

USER-FRIENDLY:

This problem can be easily solved by the user in a couple of minutes!

The thumb phalanges are not properly placed: indeed it can happen that they don't always go back to the original position (this highly depends on how fingers envelop the object and how much each phalanx dislocate from its original place).

As you can see in the video I attached below, you can try to manually adjust the thumb configuration by pulling or stretching each phalanx.

If you make sure that thumb phalanges are aligned, you should have no more troubles.

If the thumb seems ok, just try to do the same to all the other fingers: pull and twist a bit each distal phalanx until it seems to be in the proper place (set the hand reference to 0%, i.e. fully open, when you do such maintenance).

NOTE:

It is a good practice to perform these manual adjustments once in a while to ensure that the hand works at its best.

Please, let us know whether this solve the problem or you need further assistance.

9.9 The hand is connected but does not move

9.9.1 Problem

The hand is completely open — i.e. it is in its resting configuration.

The user interface shows that the hand is properly connected and you can see the device status card (from either the installation tab or the toolbar on e-Series robots) working correctly.

However the hand does not move even if the *"Hand Closure"* status bar moves accordingly with user commands and the motor turns — i.e. you can hear its quite sound for something close to 1 second.

It may happen that the thumb (or the little) fingers move a bit from their rest configuration, but really far from what is expected.

Also, the *"Motor Current"* status bar has really low values, even when you hear the motor moving.

9.9.2 Solution

If all the above conditions match your status, there may be a hardware fault which requires a qualified service action.

To check whether you are on such a case or not you should perform this simple mechanical test:

1. Disconnect the hand from the power source;
2. Flex the fingers from index to little (all together) in the direction of the back of the hand — i.e. opposite to the usual closure direction;
3. Repeat point (2.) several times and take care whether the thumb moves while doing such a movement or not;

If the thumb does not move at all or move just a bit from its rest configuration, you will need to raise a *"Service Request"* in our portal. We will do our best to ensure that this goes as quickly and as smoothly as possible.



NOTE:

If all the previous conditions match the thumb should never move.
Anyway, if you think you have this problem but the thumb moves while flexing the other four fingers together, follow the same steps as this other topic (see Page 43) .

9.10 The hand often disconnects while the robot is moving

9.10.1 Problem

When the robot is stopped, the user interface shows that the hand is properly connected and you can use it with no problem for teaching grasp and set waypoints; the device status card (from either the installation tab or the toolbar on e-Series robots) display data correctly.

However, when you start a robot program it often happens that it stops saying that the hand is not connected any longer.

9.10.2 Solution

There may be a couple of reasons for this unexpected behavior:

1. Some cables are not properly connected or secured, and robot vibrations when it starts moving can cause faults in the communication channel;
2. If you are powering the device from an external power source, there may be some electrical noise in your network that is not properly filtered out.

If you are powering the device from an external power source, first try the same tasks with the UR Control Box as PSU. If this solves the disconnection problems you are for sure in case (2.): you need either to isolate the power source from your network noise or change the PSU.

On the other hand, if these steps do not solve the problem — i.e. the device still frequently disconnect from the robot system while it is operating — please raise a support request with at least the following details at our help desk portal:

- The device serial number;
- The robot system type and software version;
- The installed qbrobotics URCap version;
- The list of any other URCap installed in your robot;
- The power supply configuration;
- Any error reported in the UR logs tab;
- Any remarkable info;



WARNING:

Be aware that the device must not violate our warranty policy, cf. maintenance and warranty (see Page 37), in order to ensure the best support service for our customer.

9.11 The USB flash drive is not working

9.11.1 Problem

It is not so common, but it may happen that any of the files provided through the USB flash drive is corrupted.

9.11.2 Solution

You can find all the USB flash drive files in our download section at the bottom of the qb SoftHand page[13].

13. https://qbrobotics.com/it/download/qb-softhand-industry-downloads/

WARNING:

If any of the files is corrupted we recommend to format the flash drive and download again all the files from our website.