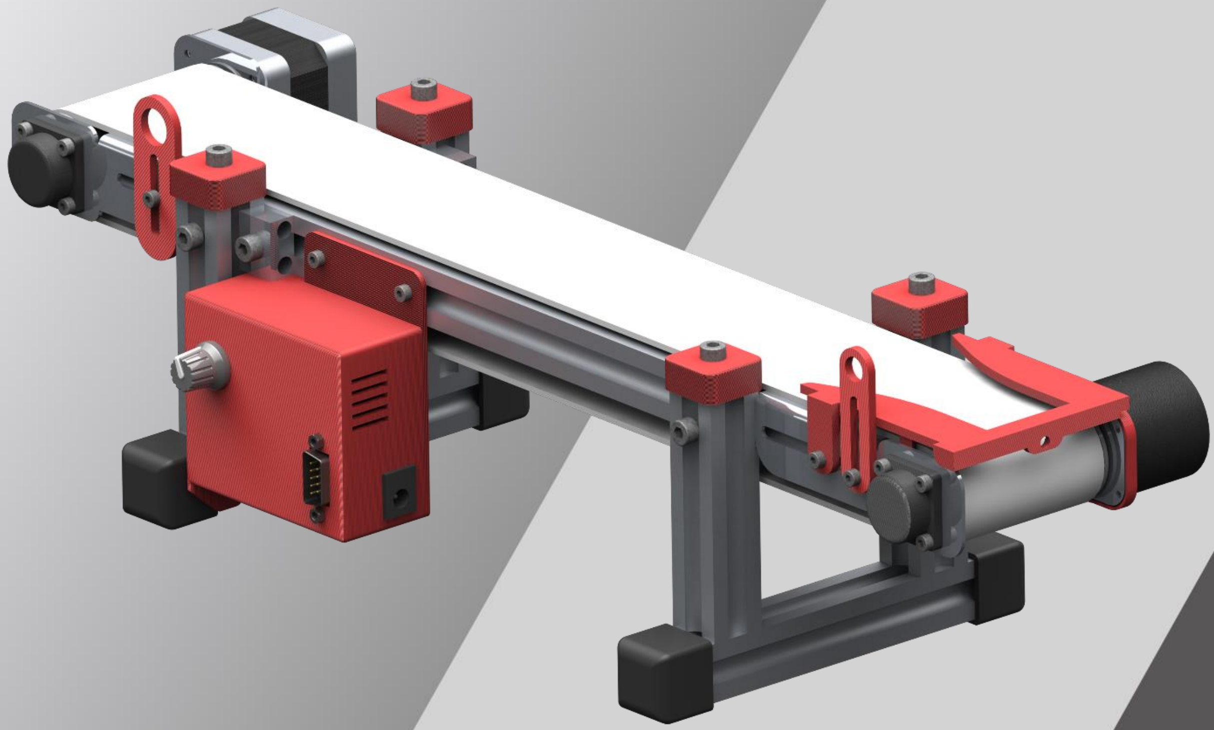
Kawasaki Robotics

Astorino

Conveyor Operation Manual



Preface

This manual describes the handling of the 6-axis robot “astorino" conveyor option.

The ASTORINO is a learning robot specially developed for educational institutions. Pupils and students can use the ASTORINO to learn robot-assisted automation of industrial processes in practice.

1. The "astorino" software included with the ASTORINO is licensed for use with this robot only and may not be used, copied or distributed in any other environment.
2. Kawasaki shall not be liable for any accidents, damages, and/or problems caused by improper use of the ASTORINO robot.
3. Kawasaki reserves the right to change, revise, or update this manual without prior notice.
4. This manual may not be reprinted or copied in whole or in part without prior written permission from Kawasaki.
5. Keep this manual in a safe place and within easy reach so that it can be used at any time. If the manual is lost or seriously damaged, contact Kawasaki.

Symbols

Items that require special attention in this manual are marked with the following symbols.

Ensure proper operation of the robot and prevent injury or property damage by following the safety instructions in the boxes with these symbols.

⚠ WARNING

Failure to observe the specified contents could possibly result in injury or, in the worst case, death.

[**ATTENTION**]

Identifies precautions regarding robot specifications, handling, teaching, operation, and maintenance.

⚠ WARNING

1. **The accuracy and effectiveness of the diagrams, procedures and explanations in this manual cannot be confirmed with absolute certainty. Should any unexplained problems occur, contact Kawasaki Robotics GmbH at the above address.**
2. **To ensure that all work is performed safely, read and understand this manual. In addition, refer to all applicable laws, regulations, and related materials, as well as the safety statements described in each chapter. Prepare appropriate safety measures and procedures for actual work.**

Paraphrases

The following formatting rules are used in this manual:

- For a particular keystroke, the respective key is enclosed in angle brackets, e.g. <F1> or <Enter>.

- For the button of a dialog box or the toolbar, the button name is enclosed in square brackets, e.g. [Ok] or [Reset].

- Selectable fields are marked with a square box □.
  If selected a check mark is shown inside the symbol ☑.

List of contents

1 Nomenclature in this manual

The author of the manual tries to use generally valid terminology while achieving the greatest possible logical sense. Unfortunately, it must be noted that the terminology is reversed depending on the point of view when considering one and the same topic. Also it is to be stated that in the course of the computer and software history terminologies developed in different way. One will find therefore in a modern manual no terminologies, which always satisfy 100% each expert opinion.

2 Overview of ASTORINO

The ASTORINO is a 6-axis learning robot developed specifically for educational institutions such as schools and universities. The robot design is based to be 3D printed with PET-G filament. Damaged parts can be reproduced by the user using a compatible 3D printer.

Programming and control of the robot is done by the "astorino" software.

The latest software version and 3D files can be downloaded from the KAWASAKI ROBOTICS FTP server:

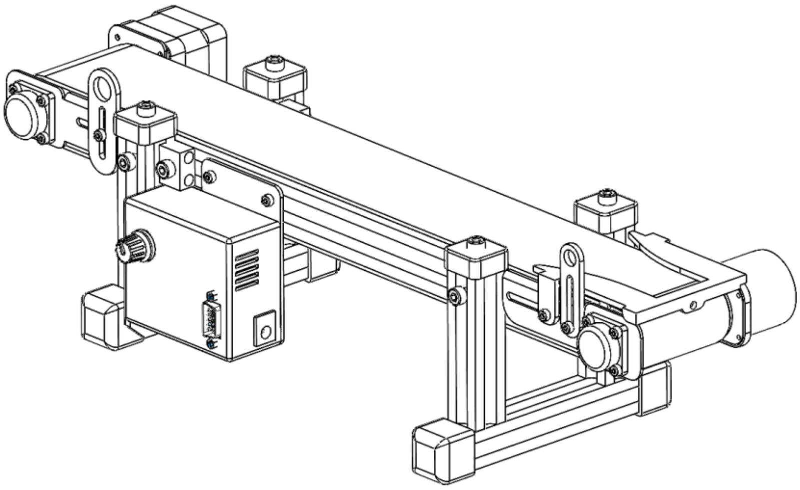
https://ftp.kawasakirobot.de/Software/Astorino/

Just like Kawasaki’s industrial Robots the ASTORINO is programmed using AS language. Providing transferable programming skills from the classroom to real industrial applications.

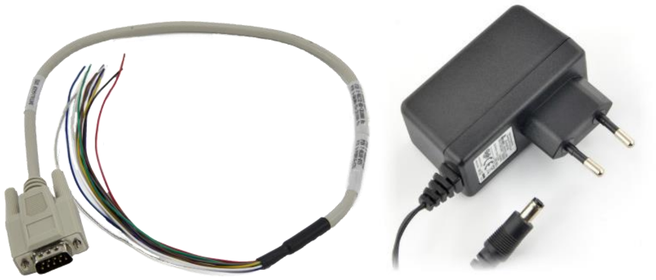
3 Technical specifications

Characteristics		Astorino Conveyor
Working environment	Temperature	0–40°C
	Humidity	35–80%
Controller		Arduino UNO
Max. motor speed		50 rpm
Max. current consumption		2000 mA
Size		450x150x120mm
Power supply		12V
Belt speed control		10 – 50 mm/s
Belt length		450 mm
Belt width		60 mm
4 INPUTS		2x24V, 2x5V
Weight		2 kg
Material		Aluminium, PET-G
Colour		Silver/Red
Incremental encoder		360 ppr, 5-30V

4 Conveyor package contents

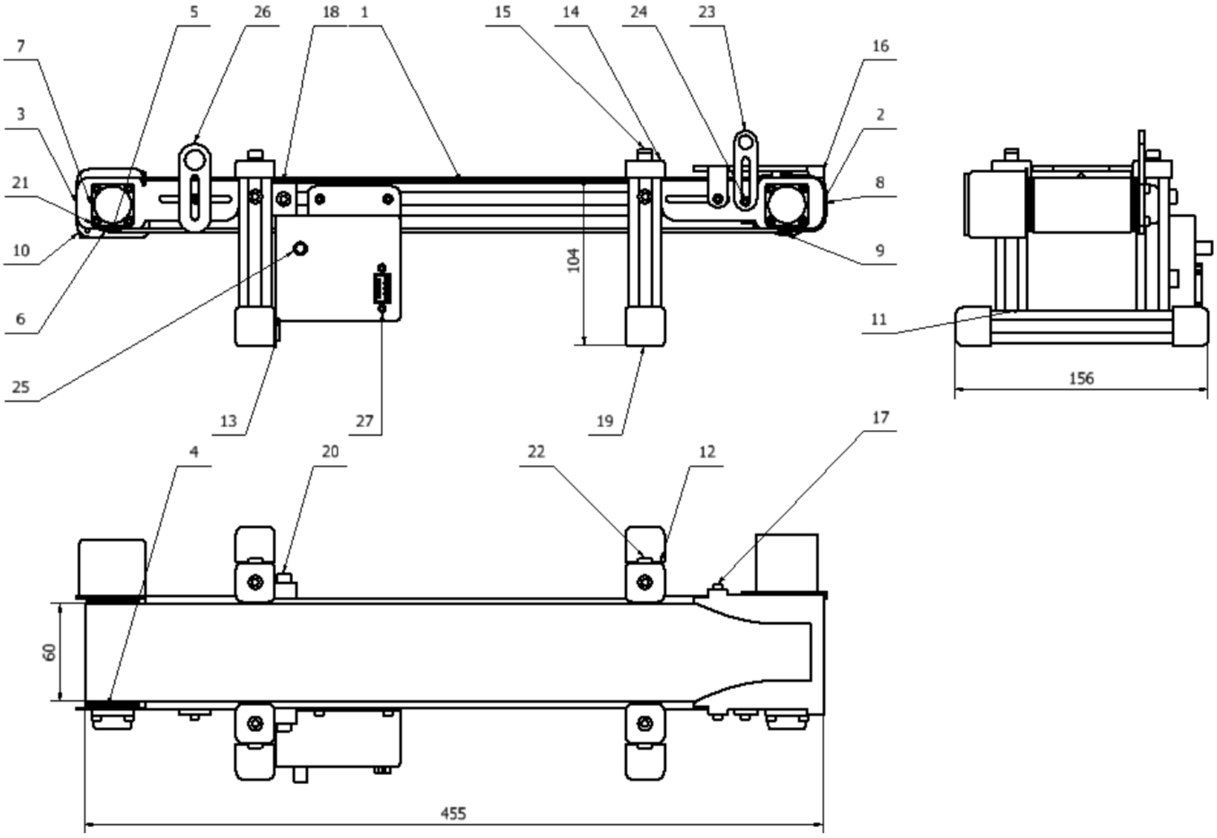


Conveyor



DSUB9 cable, power supply

5 Dimensions



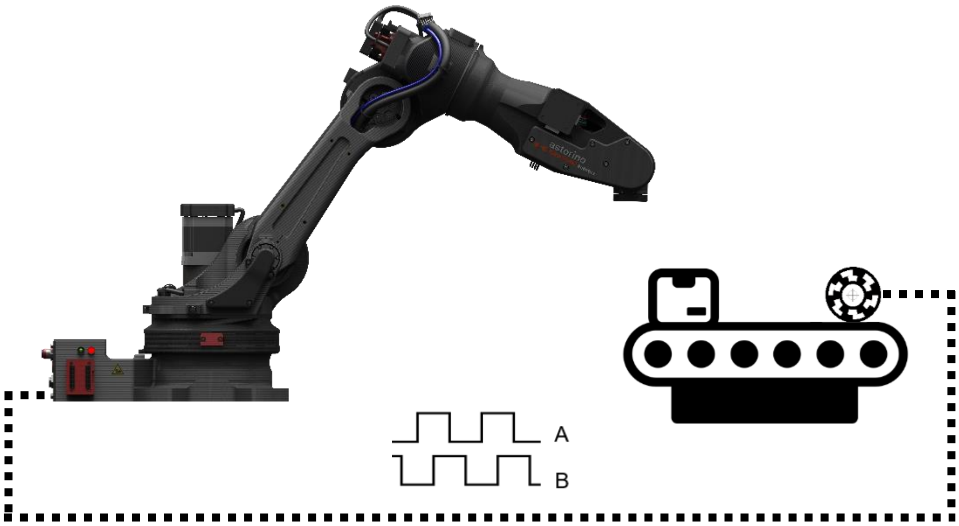
Item	quantity	Name
1	2	2020
2	1	Encoder mount
3	1	Motor mount
4	2	Drive shaft
5	2	Friction pad
6	1	Belt
7	2	Bearing housing
8	1	Encoder holder
9	1	Encoder
10	1	Stepper motor
11	4	Legs
12	2	Support beam
13	1	Controller
14	4	Dummy plug
15	8	DIN912 M8x8
16	1	Positioner
17	2	DIN912 M3x5
18	2	Belt tensioner
19	4	Beam dummy plug
20	2	DIN912 M5x10
21	11	DIN912 M3x6
22	4	DIN912 M5x20
23	1	Sensor holder
24	1	DIN912 M3x6
25	1	Rotary knob
26	1	Sensor holder
27	1	Signal connector

6 Cooperation with external encoder

In standard robot operations, the workpiece/workpiece remains stationary during operation. The conveyor synchronization function allows operations on objects moving on the conveyor belt.
Using the function of cooperation with an external encoder, the robot moves, synchronizing its movement with a moving object on the conveyor belt. To synchronize with the moving workpiece, the robot can use up to two external incremental encoders.
Taking into account the sequence of movements and the flow of the program should be avoided:

- movements that will cause going beyond the working range of the robot,
- unnecessary pause of work (stopping the robot while the object passes by the robot)

Before using the synchronous conveyor function, the parameters of the resolution and direction of movement of the conveyor belt must be set. Set the data in the Astorino software.

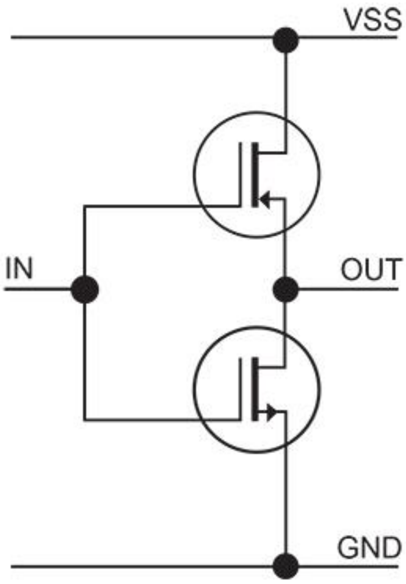


7 Supported encoders

Astorino is able to handle up to two additional incremental encoders at the same time.

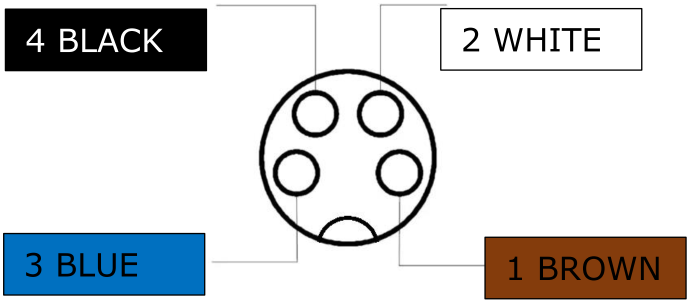
The basic parameters of the supported encoders are:

- Operating voltage 24V,
- Signal outputs A and B,
- Recommended resolution of no more than 300 PPR (pulses per revolution),
- Outputs operating in PUSH-PULL configuration.



Output in PUSH-PULL configuration

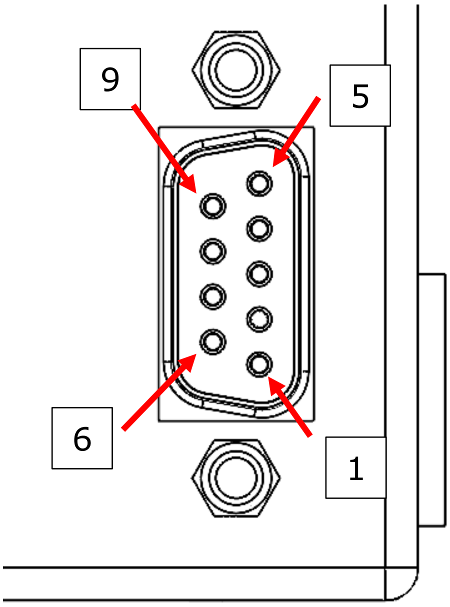
Every encoder in the robot is connected on the M8 four-pole plug.



Entry No.	1 BROWN	2 WHITE	3 BLUE	4 BLACK
1	A	B	24V	GND
2	A	B	24V	GND

8 Electrical connection

8.1 DSUB9-pinout



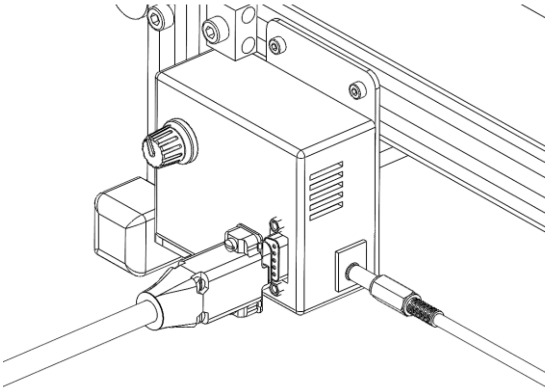
Pin	Function	Color
1	N/C	-
2	ENABLE 5V	PINK
3	DIR 5V	GREY
4	ENABLE 24V	YELLOW
5	DIR 24V	GREEN
6	N/C	-
7	N/C	-
8	N/C	-
9	GND	WHITE

9 Unboxing and starting-up

Once the conveyor is removed from the packaging, place it on a solid surface.

9.1 Connection

- Connect D-SUB 9 cable and power plug to the conveyor

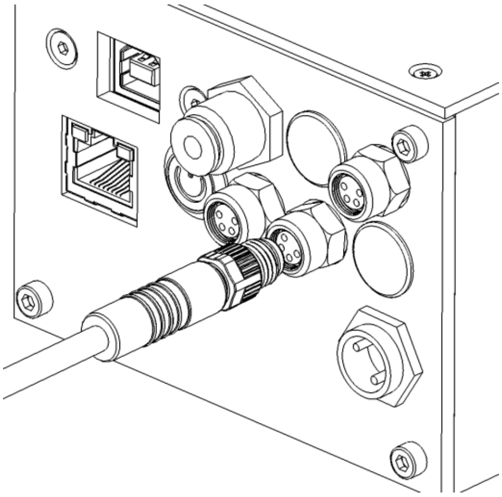


[ATTENTION]

When the power is switched on, the motor controller is activated. At this point, the belt is blocked and it's hard to move it manually.

9.2 Connection to astorino robot – encoder

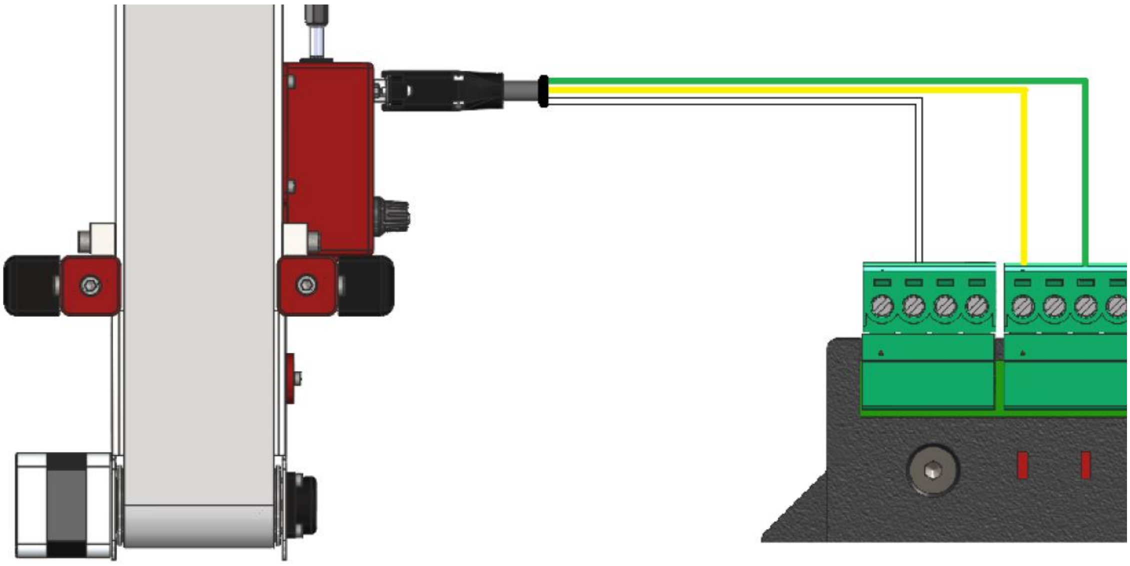
Connect M8 cable from encoder to one of conveyor encoder inputs on the robot base. If robot do not have dedicated M8 plug for encoder see Attachment A – installation of conveyor tracking option of this document.



9.3 Connection to astorino robot - signals

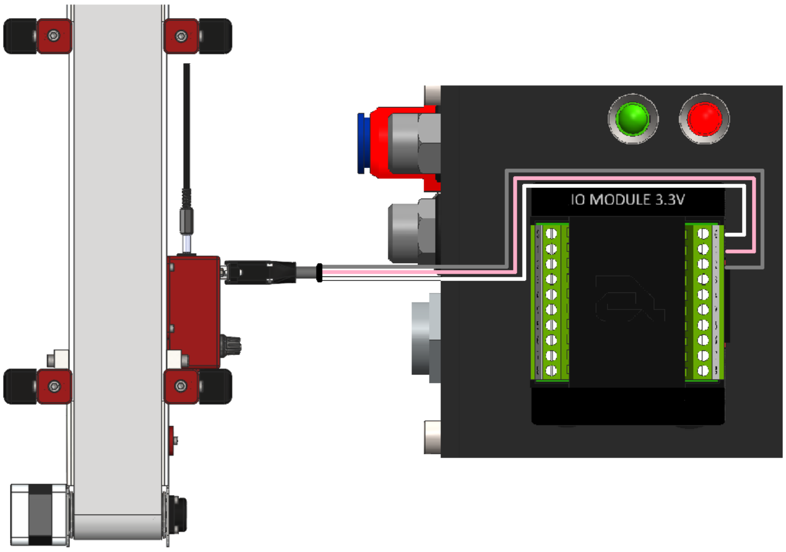
9.3.1 24V IO module

Connect 3 wires from DSUB9 connector to the 24V IO Module. In this example ENABLE signal is connected to 1st OUTPUT, DIR signal is connected to 2nd OUTPUT.



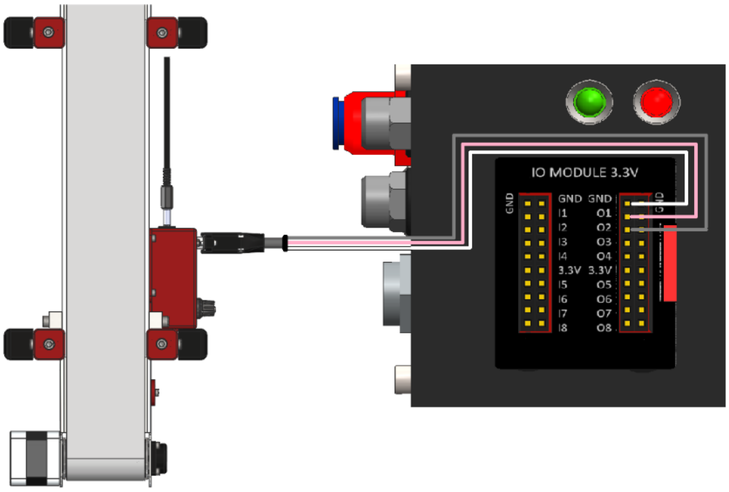
9.3.2 3.3V IO Adapter

Connect 3 wires from DSUB9 connector to the 3.3V IO Adapter. In this example ENABLE signal is connected to 1st OUTPUT, DIR signal is connected to 2nd OUTPUT.



9.3.3 3.3V IO Module

Connect 3 wires from DSUB9 connector to the 3.3V IO Adapter. In this example ENABLE signal is connected to 1st OUTPUT, DIR signal is connected to 2nd OUTPUT.

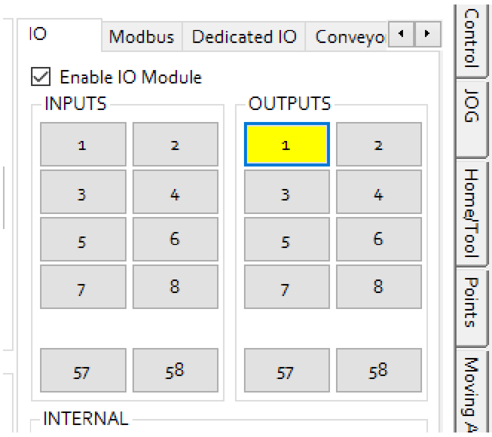


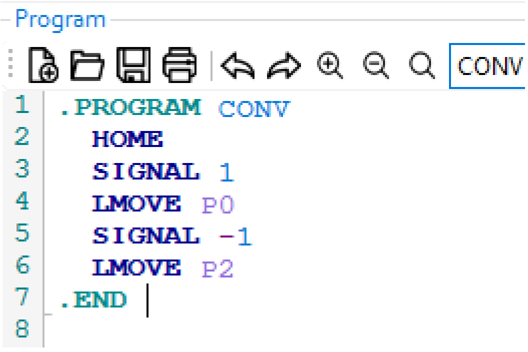
[ATTENTION]

This connection require some more work. DSUB-9 cables are prepared as naked wires and without any connector will not fit to the 3.3V IO module plug.

9.4 Controlling conveyor via OUTPUTS

To control conveyor motion and direction please use astorino software or Teach Pendant to turn ON or OFF OUTPUTS that are connected to the conveyor, or use SIGNAL command in your program



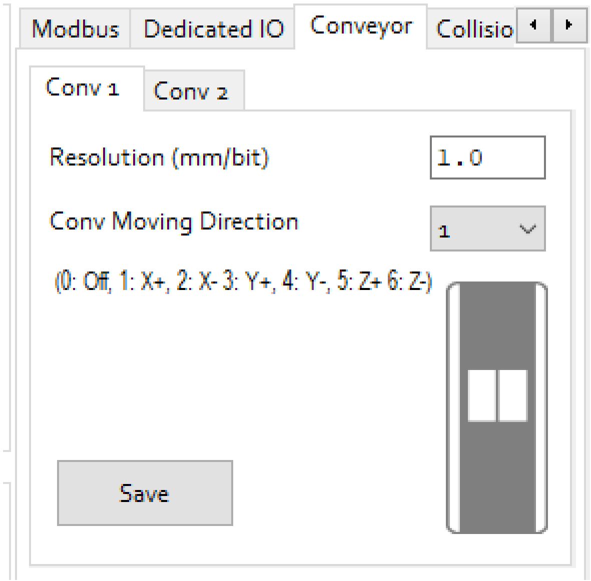


```
.PROGRAM CONV
  HOME
  SIGNAL 1
  LMOVE P0
  SIGNAL -1
  LMOVE P2
.END
```

9.5 Setting encoder data for conveyor tracking

Before running any programs with conveyor tracking you need to set the correct resolution and motion direction of the belt. It is done in the astorino software, on the Sys. Set. Page in the Conveyor section.

- Resolution is an information how many millimetres are travelled on single encoder pulse,
- Moving direction is an information how the conveyor is placed according to the robot BASE coordinate system. For example X+ means that the belt is moving in a positive direction along X axis



Resolution (mm/bit) 1.0

Conv Moving Direction 1

(0: Off, 1: X+, 2: X- 3: Y+, 4: Y-, 5: Z+ 6: Z-)

Standard values:

- Resolution: 0.2967 mm/bit
- Moving Dir: 1 (X+)

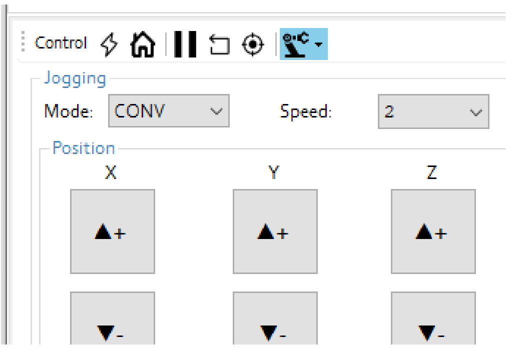
After setting the data press [SAVE]

9.6 Checking conveyor synchronization

To check if the conveyor tracking is working corectly you can use CONV mode in JOG movements.

[**ATTENTION**]

Before make sure that the correct cooperation is on by typying CVCOOPJT 8 or 9 in the Terminal, this will turn on cooperation with 1st or 2nd external encoder. Normally after powering up the robot it is set to 1st encoder.



Control

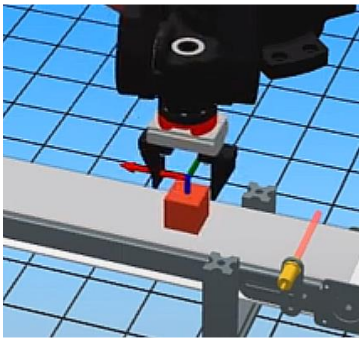
Jogging

Mode: CONV

Speed: 2

Position

Move astorino robot that it is above conveyor.



Then switch mode to CONV, turn on the conveyor so the belt is moving and press one of the JOG buttons. Robot should move in synchronization with the belt.

10 Movement instructions in cooperation with the conveyor belt

CVLMOVE	Linear motion in cooperation with conveyor belt
CVLAPPRO	Approaches the target point linearly in cooperation with the conveyor belt
CVLDEPART	Leaves the current position linearly in cooperation with the conveyor belt
CVDELAY	Stops the movement of the robot for a specified period of time in cooperation with the conveyor belt
CVWAIT	Stops the robot until the conveyor reaches the set value
CVRESET	Overwrites the current position of the conveyor belt
CVPOS	Reads the current position of the conveyor 1
CVPOS2	Reads current position of the conveyor 2
CVCOOPJT	Turns on robot cooperation with conveyor 1 or 2

CVLMOVE position variable

Function

The movement of the robot to a specific position in linear interpolation with synchronization with the conveyor.

Parameter

Position variable

Specifies the target position of the robot's movement. (It can be in transformation values, composite transformation values, or joint values.)

Explanation

TCP follows a linear trajectory from the start to the end position, synchronizing with the conveyor.

Example

CVLMOVE #pick	Linear motion to the defined by the values of the joints angles (#pick) during synchronization with the conveyor.
CVLMOVE Place	Linear motion to the position defined by the transformation variable "place" when synchronizing with the conveyor.

CVLAPPRO position variable, distance

Function

Movement in linear interpolation to a specified distance from a certain position, synchronizing with the conveyor.

Parameter

Position variable

Specifies the target position (in transformation values or axis angles)

Distance

Determines the distance in the direction of the tool's Z axis between the target position specified above and the position that the robot actually achieves. (Unit: mm)

Providing a positive distance value moves the robot away from the target position (negative direction of the tool's Z-axis). Entering a negative value moves the robot towards the target position (positive direction of the tool's Z-axis).

Explanation

In this manual, the orientation of the tool in the position in which the robot actually reaches is determined by the position variable given. The position of the tool becomes a position away from the specified position by a specified distance in the positive or negative direction of the Z axis of the tool.

Example

CVLAPPRO Place,100 The robot synchronizes with the conveyor and moves in linear interpolation to a position 100 mm away in the direction of the tool Z axis from the defined by the values of the "Place" transformation.

CVLDEPART distance

Function

Movement in linear interpolation to a specified distance from the current position, synchronizing with the conveyor.

Parameter

Distance

Determines the distance in the direction of the tool's Z axis between the current position and the position that the robot actually achieves. (Unit: mm)

Providing a positive distance value moves the robot away from its current position (negative direction of the tool's Z-axis). Entering a negative value moves the robot towards the current position (positive direction of the Z axis of the tool).

Explanation

In this manual, the orientation of the tool in the position in which the robot actually reaches is determined by the current position of the robot. The position of the tool becomes a position away from the current position by a specified distance in the positive or negative direction of the tool Z-axis.

Example

CVLDEPART 100 The robot synchronizes with the conveyor and moves in linear interpolation to a position 100 mm away in the direction of the Z-axis of the tool from the current position.

CVDELAY time

Function

It "stops" the movement of the robot as seen from the conveyor reference view point for a specified period of time.

Parameter

Time

Specifies the amount of time during which the robot must remain "still" as seen from the conveyor's perspective (Unit: seconds)

Explanation

The **CVDELAY** instruction is a robot movement instruction. After following this instruction, the movement of the robot is controlled so as to maintain the same position relative to the moving conveyor, and therefore, when viewed from the conveyor side, the robot seems to stop. (Looking from externally, the robot moves in accordance with the conveyor, so that the same position is maintained in relation to the workpiece/workpiece on the conveyor).

Example

CVDELAY 2.5	The robot remains stationary for 2.5 seconds from the conveyor's perspective.

CVWAIT starting position

Function

Pauses the execution of the program until the conveyor reaches the specified position. (Unit: mm) .

Parameter

Starting position

The robot starts again when the conveyor reaches this position.

Explanation

When this instruction is executed in the program, the robot stops and waits (does not perform the next step in the program) until the conveyor reaches the specified position. The robot resumes work when the conveyor reaches a certain position.

Example

CVWAIT 50	Further execution of the program is suspended until the 50mm conveyor belt position is reached.

CVRESET axis number

Function

Resets the position value of the currently cooperating conveyor belt.

Parameter

Axis number

Specifies the conveyor number (axis number) which needs to be reset. This function overwrites the value of conveyor to 0. Possible parameter values are 8 (conveyor 1) and 9 (conveyor 2)

Explanation

The CVRESET statement resets the value of a specified conveyor

Example

CVRESET 8	Resets the Conveyor 1 to 0mm

CVPOS

Function

Assigns the current position of the first conveyor to a variable.

Example

CONV1 = CVPOS	Assigns to the variable CONV1 the current value of the position of the first conveyor

CVPOS2

Function

Assigns the current position of the second conveyor to a variable.

Example

CONV2 = CVPOS2	Assigns to the variable **CONV2** the current value of the position of the second conveyor

CVCOOPJT axis number

Function

Enables the robot to cooperate with the conveyor belt.

Parameter

Axis number

Specifies which robot conveyor belt is currently to work with. Possible values 8 and 9

Explanation

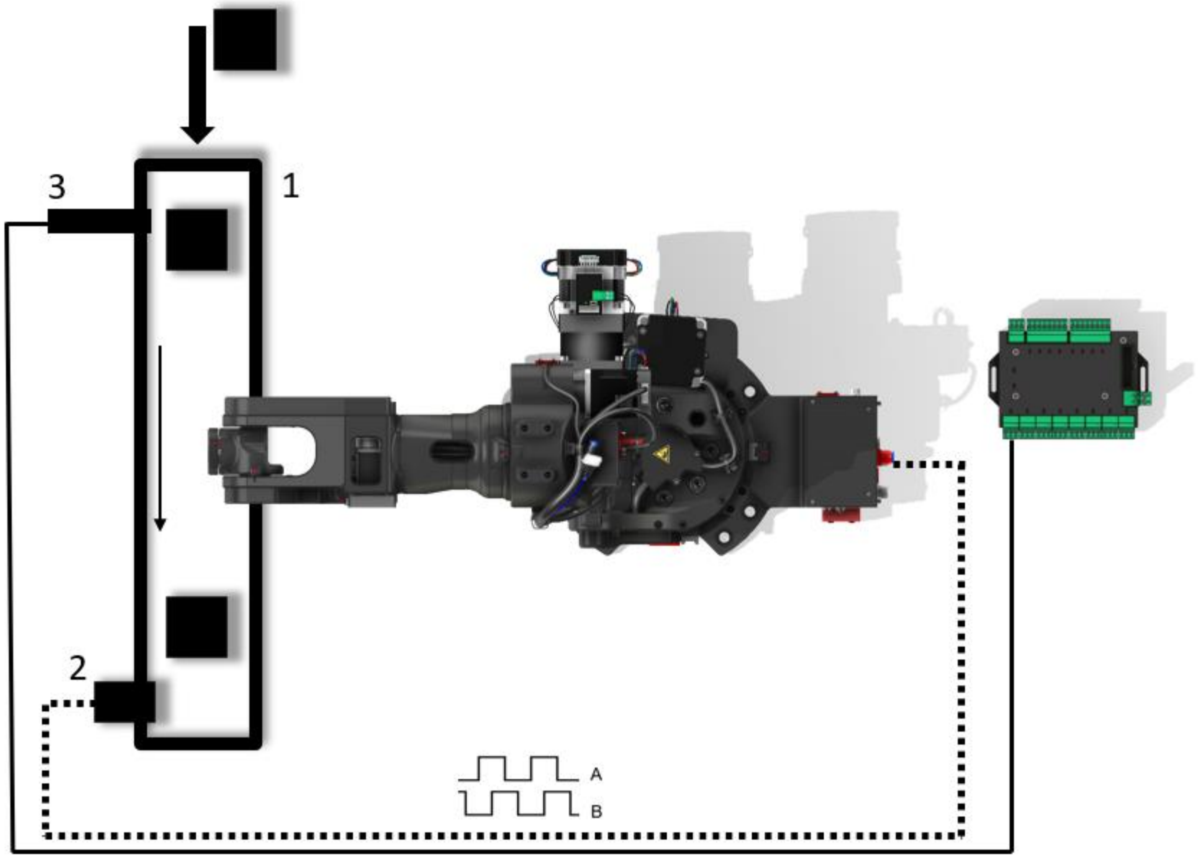
If the function has not been used, the robot cooperates with conveyor number one.

Example

CVCOOPJT 8	Activates the cooperation of the robot with the first conveyor

11 Example of a conveyor belt application

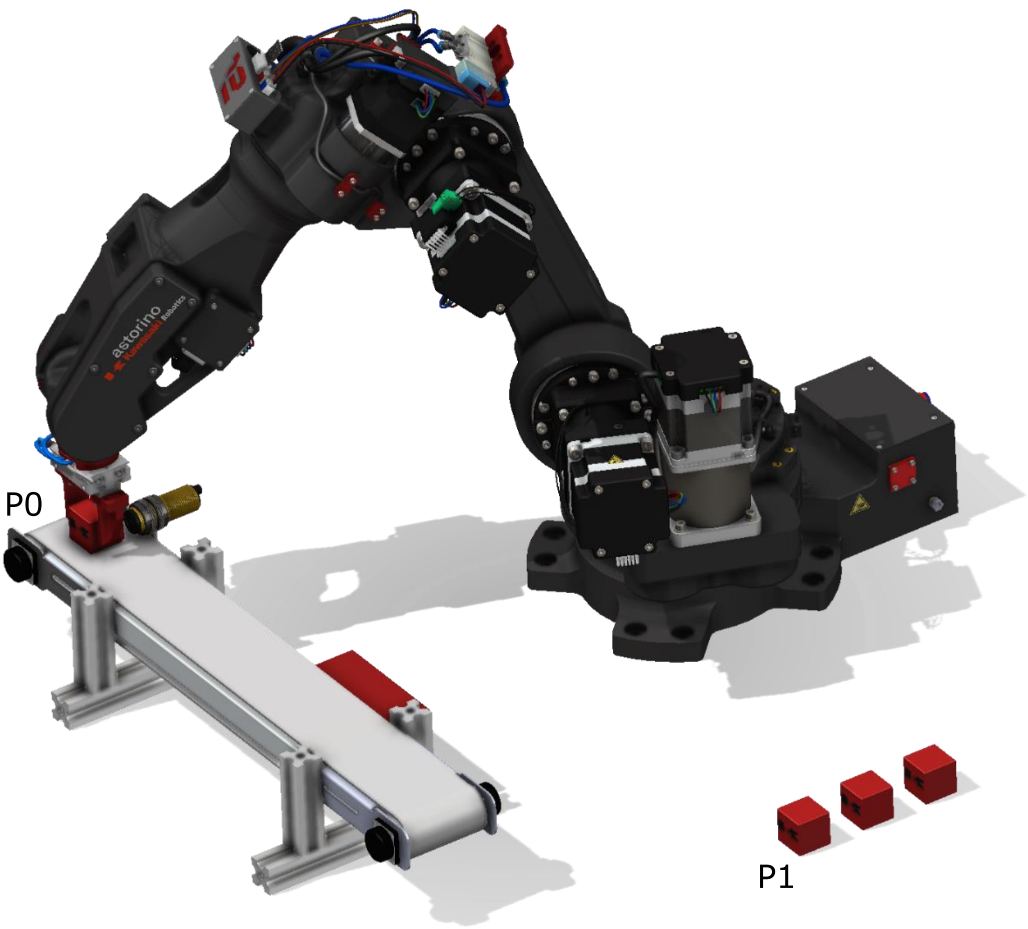
In the example below, the robot is equipped with a conveyor belt (1), a 24V pulse encoder (2) and a proximity sensor (3). The encoder has been connected to the input of the first encoder and the proximity sensor to the first input in the 24V I/O module. In order to clarify the diagram, the connection of the 24V IO module with the robot is not shown. The following example assumes that the conveyor has its own control and its movement matches the arrow in the drawing.



In the above application, the user gives the workpieces (cubes) at the beginning of the conveyor belt, when the sensor triggers, a point is saved, to which the robot then goes and picks the detail at the same time synchronizing with the conveyor belt. It then puts the items in a different location.

The first thing to do is to configure the conveyor setting in the robot settings. In this example, the resolution is 0.1mm/bit and the direction is set to X-.

Before starting, record the point where the workpiece is located at the time of detection by the sensor (3) and the deposit point P1



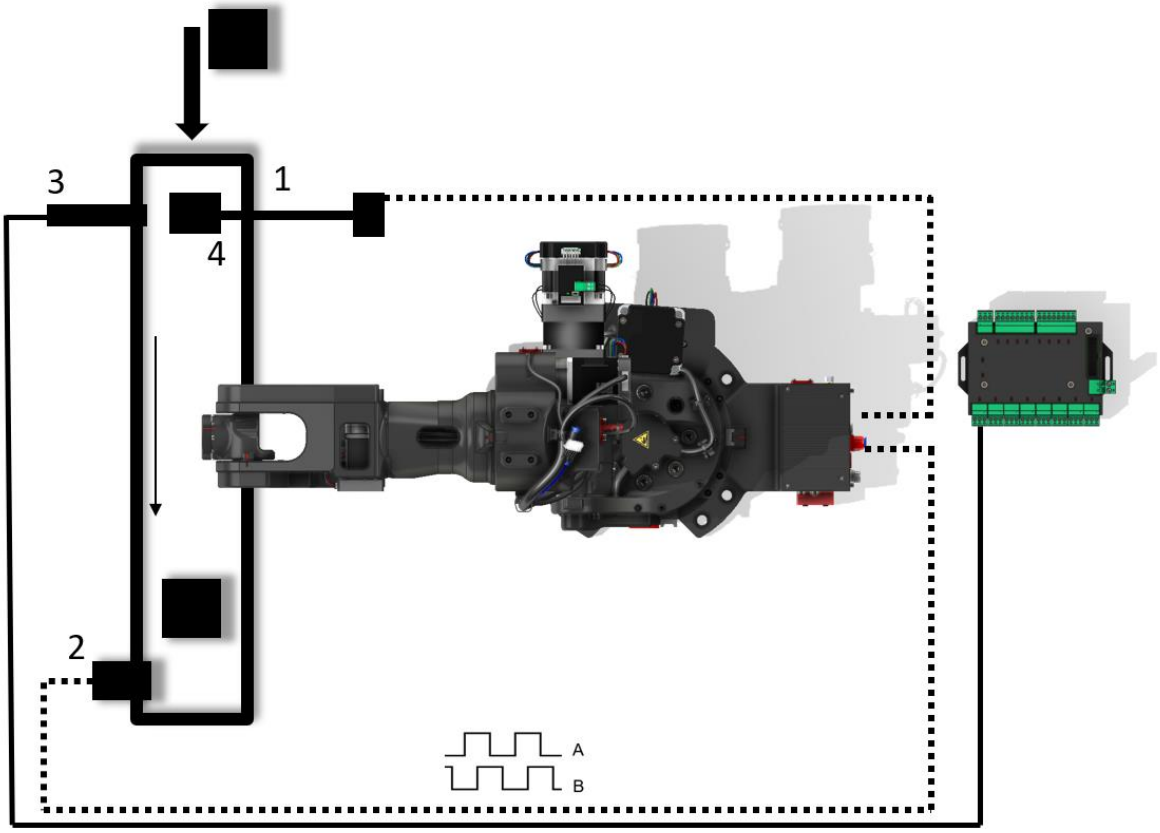
In order for the above application to work properly, the workpieces should be given in the same orientation and position on the conveyor belt (relative to the width), this can be done by designing appropriate bumpers that will automatically position the detail in the middle of the conveyor belt

Example program:

```
.PROGRAM CONV
  SPEED 100 MM/S ALWAYS
  TOOL 1
  POINT PICK = P0 ;P0 saved point at sensor
  POINT PLACE = P1 ;P1 saved put away point
  HOME
  CVCOOPJT 8; synch with 1st conv
  CVRESET 8
  WHILE SIG(1002) == TRUE DO
    SWAIT 1001 ;wait conv sensor signal
    ENC = CVPOS
    POINT/8 PICK = ENC ;store current encoder value to PICK
    CVWAIT 50 ; wait till conv moved 50 mm
    CVLAPPRO PICK,50
    SPEED 50 MM/S
    CVLMOVE PICK ;move to PICK
    CVDELAY 0.5 ;wait above conv 0.5s
    SIGNAL 1 ;close gripper
    CVDELAY 1 ;wait above conv 1s
    CVLDEPART 50
    JAPPRO PLACE, 50
    SPEED 20 MM/S
    LMOVE PLACE
    TWAIT 0.5
    SIGNAL -1
    TWAIT 1
    LDEPART 50
    POINT PLACE = SHIFT(PLACE BY 0,-50,0)
    IF CVPOS > 5000 THEN
      CVRESET 8 ; reset encoder if too big
    END
  END
.END
```

12 Example of a conveyor belt and vision system application

In the example below, the robot is equipped with a conveyor belt (1), a 24V pulse encoder (2), a proximity sensor (3) and a vision system (4). The encoder was connected to the first encoder input, the proximity sensor to the first input in the 24V I/O module, and the vision system to the Serial input in the robot base. In order to clarify the diagram, the connection of the 24V IO module with the robot is not shown. The following example assumes that the conveyor has its own control and its movement matches the arrow in the drawing.



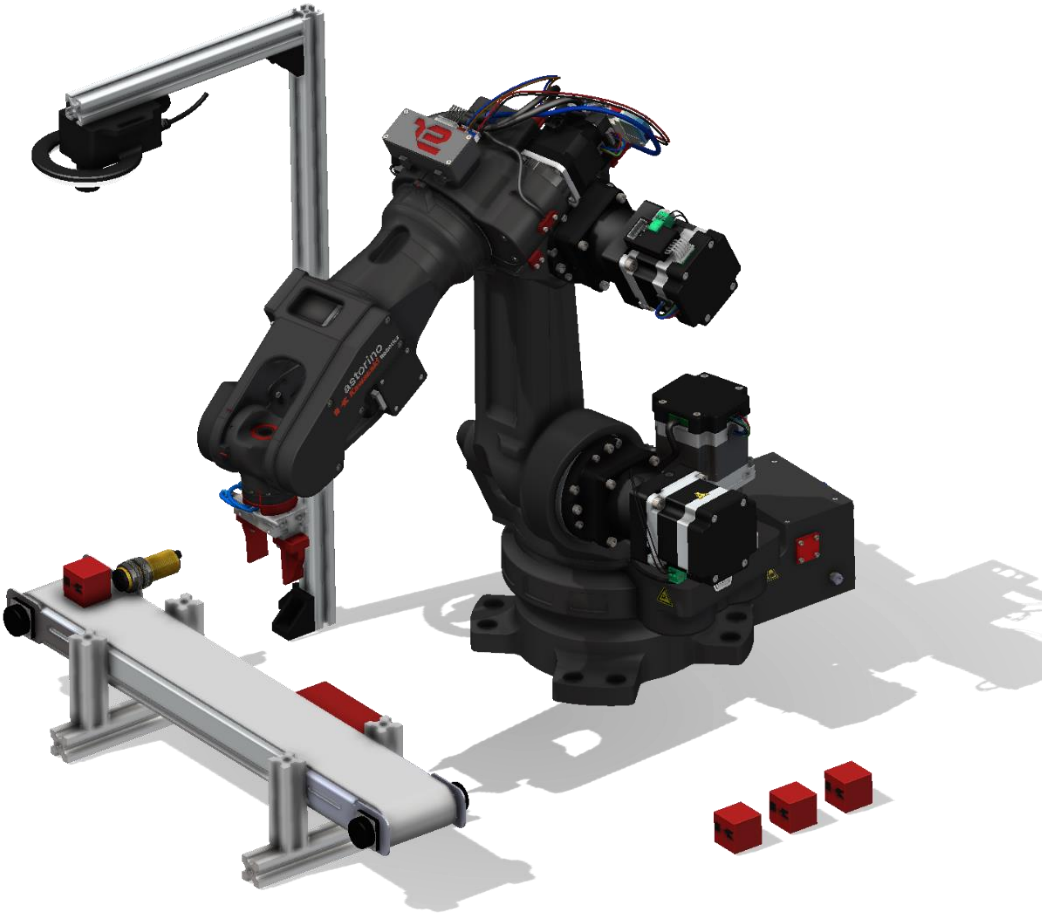
In the above application, the user puts workpieces (cubes) at the beginning of the conveyor belt, when the sensor triggers, the camera is activated, which detects the object on the conveyor belt and sends the coordinates to the robot. A saved point is used to determine the location of the workpiece. The robot then goes and picks up the cube while synchronizing with the conveyor belt. It then puts the picked up items in a different location.

The first thing to do is to configure the conveyor setting in the robot settings. In this example, the resolution is 0.1mm/bit and the direction is set to X-.

Before starting, calibrate the camera according to the instructions of the vision system and teach the P1 place point. It is also necessary, as in the previous example, to reach the point P0 at any position of the conveyor belt so as to read the coordinate Z of the intake position. Then enter it in the program in the line

POINT PICK = TRANS(dataX, dataY, height, 0,0,0).

In the example program, the value is 100mm.



```
▪ .PROGRAM CONV
  SPEED 100 MM/S ALWAYS
  TOOL 1
  POINT PLACE = P1 ;P1 saved put down point
  HOME
  CVCOOPJT 8; synch with 1st conv
  CVRESET 8
  WHILE SIG(1002) == TRUE DO
    SWAIT 1001 ;wait conv sensor signal
    SEND "T"
    WHILE EXISTCOM == false DO
      TWAIT 0.05
    END
    $temp = RECEIVE
    $temp2 = $DECODE($temp,"/")
    $temp3 = $DECODE($temp, "/")
    $temp4 = $DECODE($temp, "/")
    dataX = VAL($temp2)
    dataY = VAL($temp3)
    dataA = VAL($temp4)
    IF ((dataX <> 0) OR (dataY <> 0)) THEN
      POINT PICK = TRANS(dataX,dataY,100,0,0,0)
      POINT/OAT PICK = P0
      POINT PICK = PICK + RZ(dataA)
      ENC = CVPOS
      POINT/8 = ENC
      CVWAIT 100 ; wait till conv moved 50 mm
      SPEED 100 MM/S ALWAYS
      CVLAPPRO PICK, 40
      SPEED 40 MM/S ALWAYS
      CVLMOVE PICK ;move to PICK
      CVDELAY 0.5 ;wait above conv 0.5s
      SIGNAL 1 ;close gripper
      CVDELAY 1 ;wait above conv 1s
      CVLDEPART 50
      JAPPRO PLACE, 50
      SPEED 20 MM/S
      LMOVE PLACE
      TWAIT 0.5
      SIGNAL -1
      TWAIT 1
      LDEPART 50
      POINT PLACE = SHIFT(PLACE BY 0,-50,0)
      IF CVPOS > 5000 THEN
        CVRESET 8 ; reset encoder if too big
      END
    ELSE
      PRINT "No workpiece"
      CVRESET 8
    END
  END
.END
```

13 Manufacturer information

For further questions, contact Kawasaki Robotics support.

Contact:

Kawasaki Robotics GmbH

tech-support@kawasakirobot.de

+49 (0) 2131 – 3426 – 1310

Kawasaki Robot
Conveyor Operation Manual

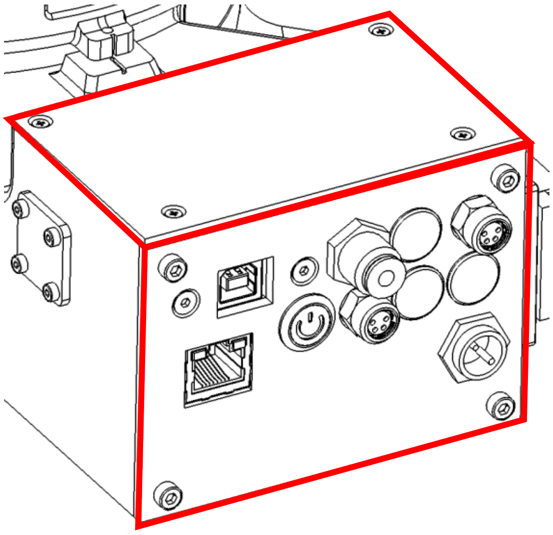
2024-01: 2nd Edition

Publication: KAWASAKI Robotics GmbH

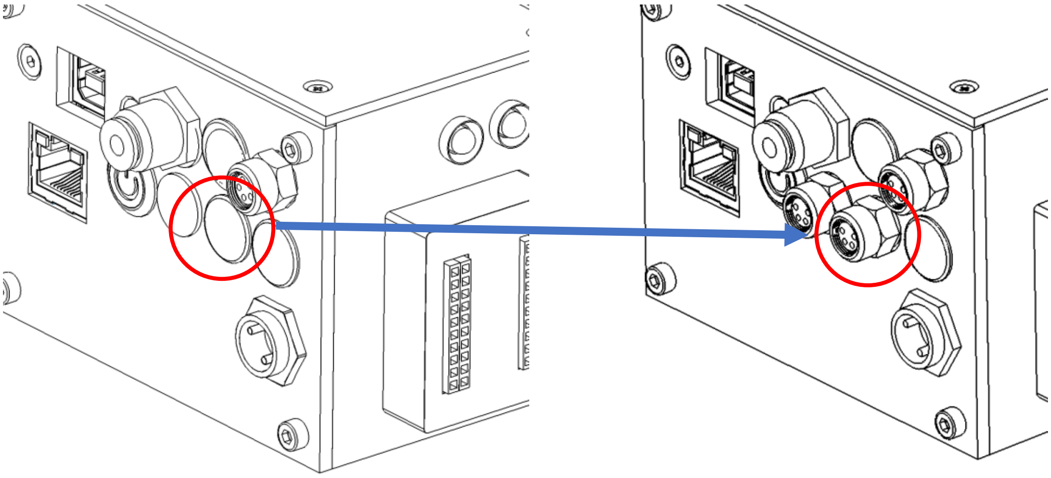
Appendix A – Conveyor tracking option installation (B - version)

To connect the conveyor M8 connector to the robot, follow these steps:

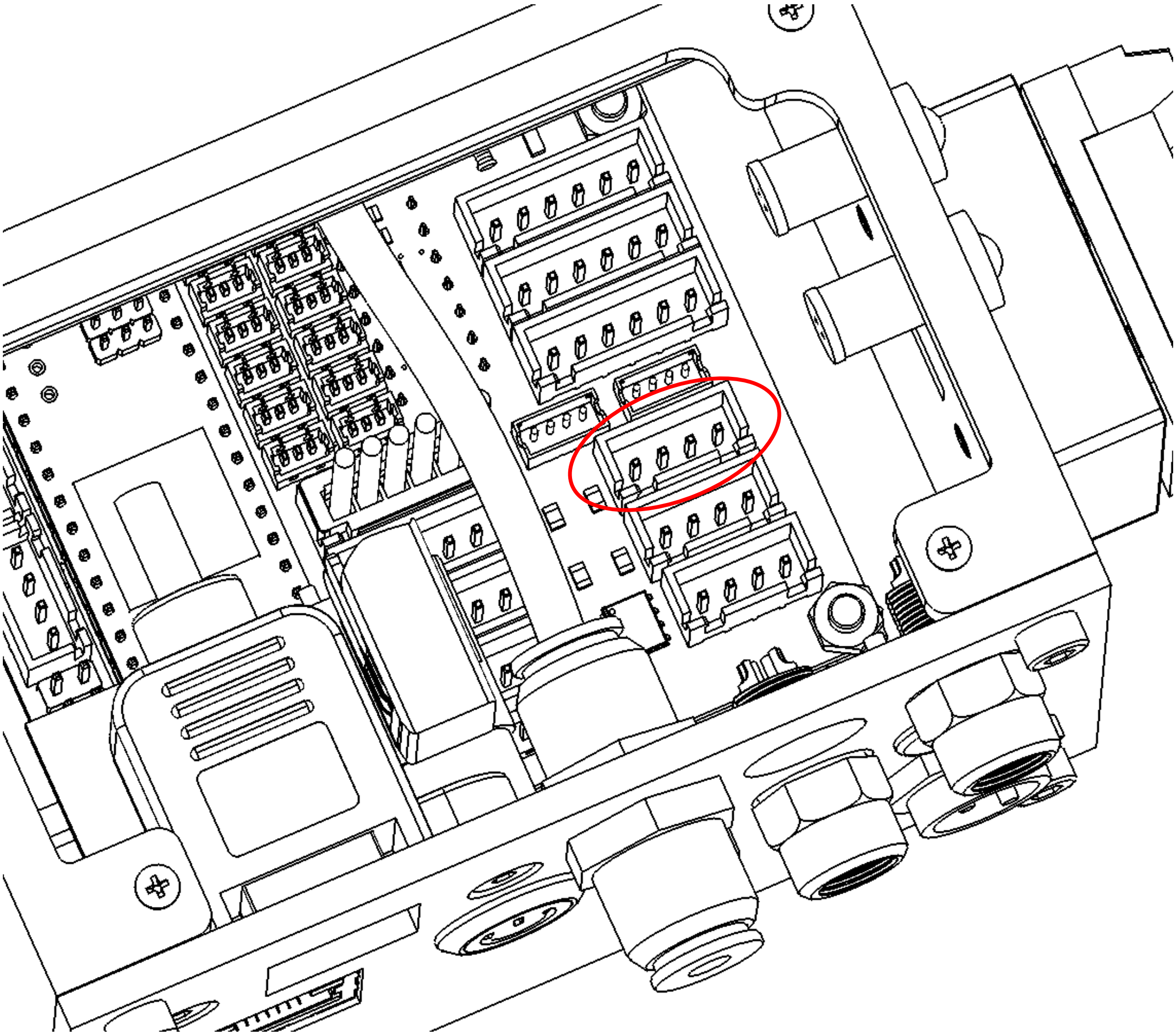
1. Disconnect the robot from the power supply,
2. Unscrew the top and back covers,



3. Insert the included M8 plug through the hole in the rear panel located next to SERIAL communication connector.



4. Plug the cable from the included M8 plug into the motherboard

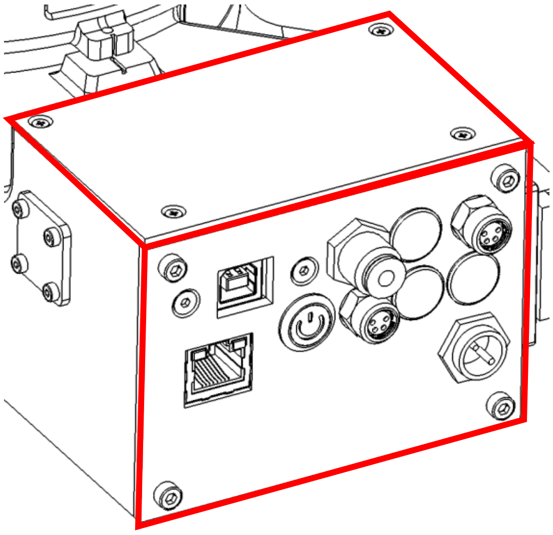


5. Close all covers.

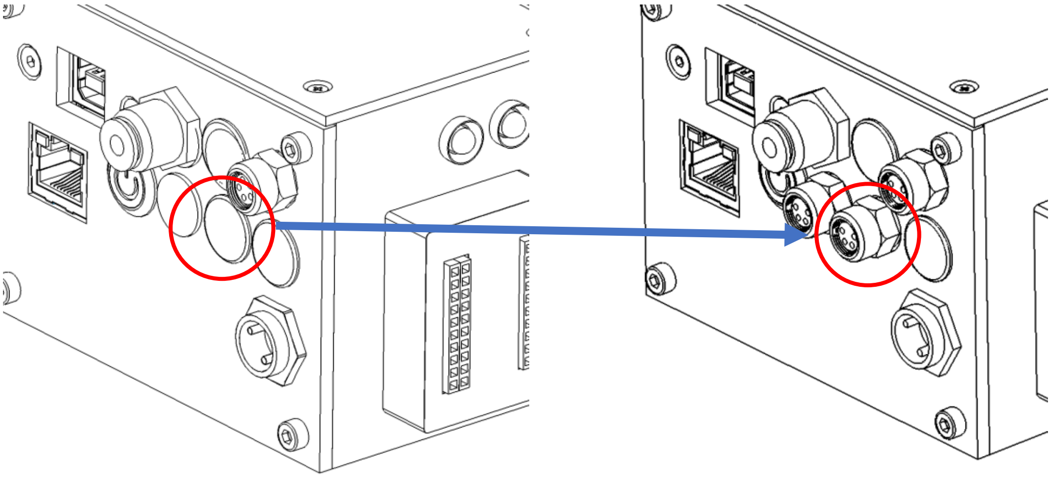
Appendix A – Conveyor tracking option installation (A- version)

To connect the conveyor M8 connector to the robot, follow these steps:

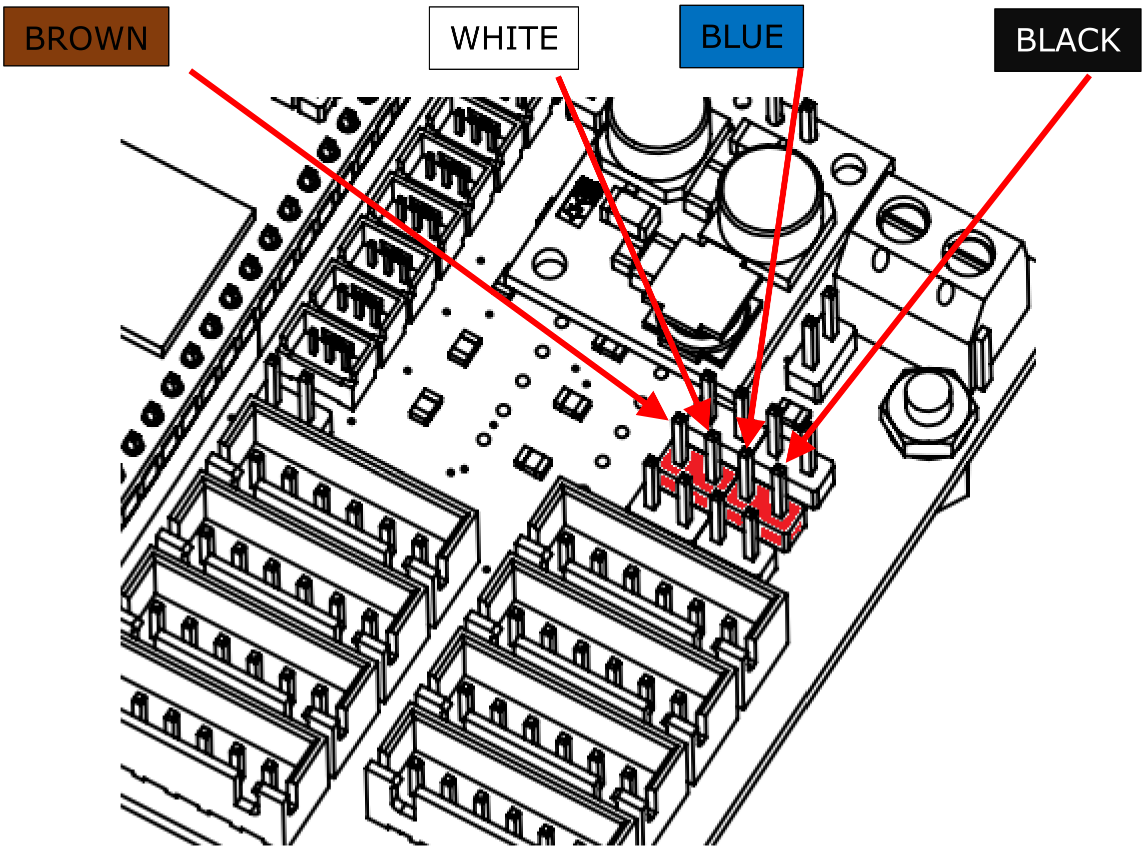
1. Disconnect the robot from the power supply,
2. Unscrew the top and back covers,



3. Insert the included M8 plug through the hole in the rear panel located next to SERIAL communication connector.



4. Plug the cable from the included M8 plug into the motherboard



5. Close all covers.