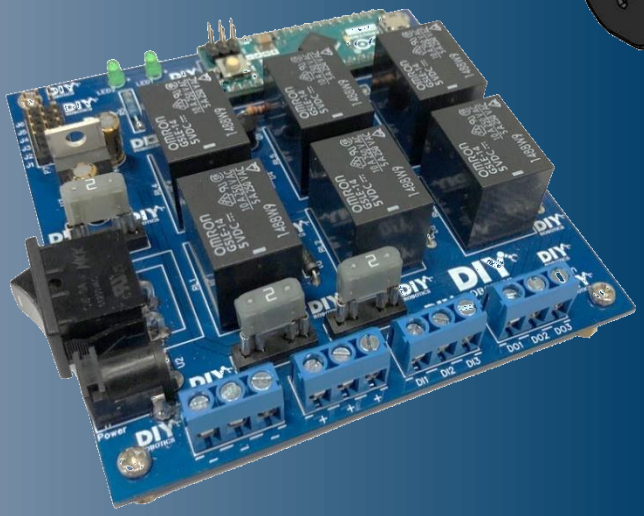
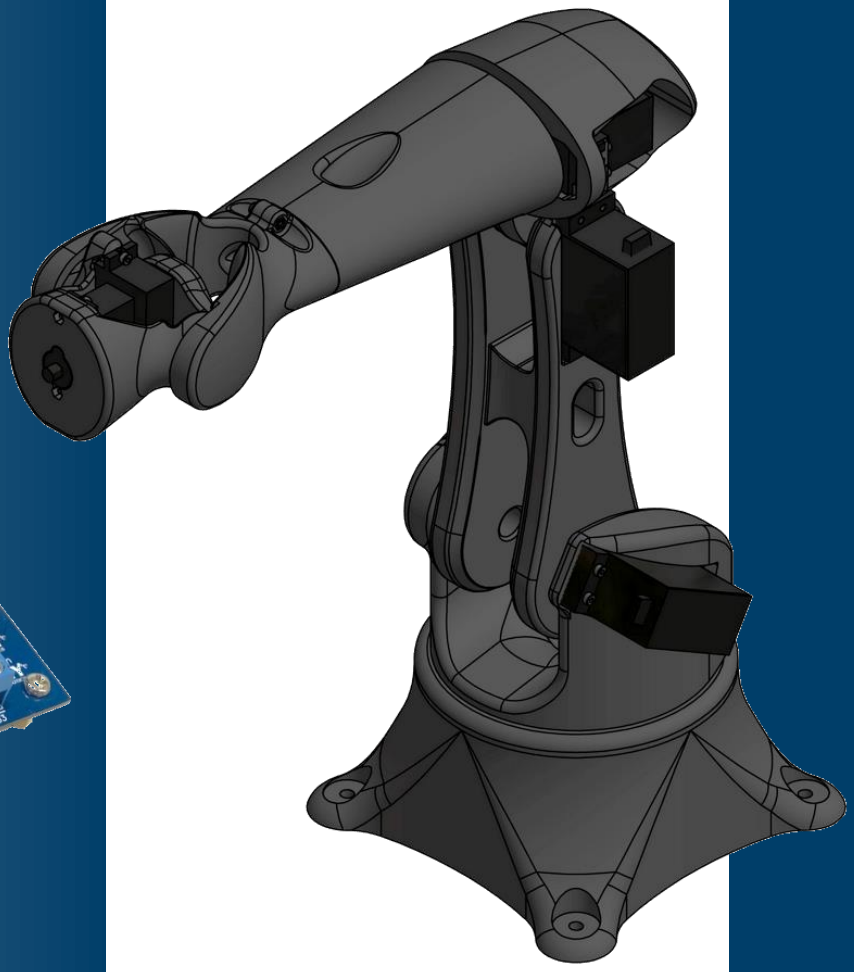
USER MANUAL

ROBOTIC EDUCATIVE CELL

V1.0





APRIL 2020

DIY ROBOTICS

www.diy-robotics.com

DIY ROBOTICS
www.diy-robotics.com

TABLE OF CONTENTS

INTRODUCTION

The DIY-Robotics educative cell is a platform that includes a 6-axis robotic arm, an electronic control circuit and a programming software. This platform is an introduction to the world of industrial robotics. Through this project, **DIY-Robotics** wishes to offer an affordable but quality solution to all those who would like to learn more about this fascinating field. This project is an excellent opportunity to develop various knowledge and skills in the fields of mechanics, electrical as well as computer science. With the **DIY-Robotics educative cell**, robotics is within everyone's reach.

This manual shows the various steps for mechanical assembly, electrical assembly and the installation and use of the **DIY-Robotics Educational Cell V1.0 software**.

You will find all files related to the development of the educational robotic cell in the compressed folder. It includes the 3D drawings of the robot, the electrical diagrams of the controller, the Arduino code, the software source codes as well as the bill of materials required.

Before getting started, be sure that you have access to a 3D printer and to buy all the components required. You will find a list of all the components required along with their price and where to order them in the bill of materials (bill-of-materials.pdf).

If you get stuck or need help make sure to check the **DIY-Robotics Forum**. You can create an account for free and ask your question to our community of accredited specialists and robotics enthusiasts.

Let’s begin! (and have fun!)

SOFTWARE

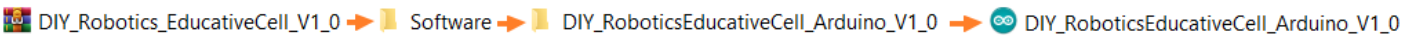
1. Overview

It is first necessary to program the microcontroller (Arduino Micro) of the robot controller. Then, the **Educative Cell V1.0** software developed by **DIY Robotics** must be installed on your computer in order to control the robot and create robot programs. The following sections will guide you through the software part of this project.

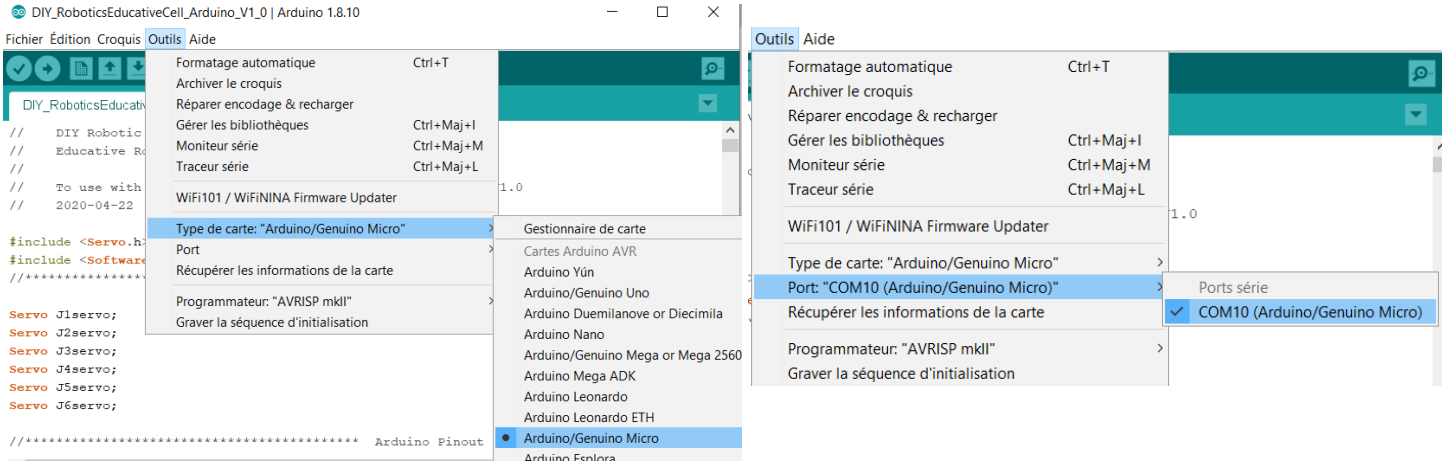
2. Arduino programming

Download the Arduino IDE software directly from the Arduino website:
https://www.arduino.cc/en/Main/Software

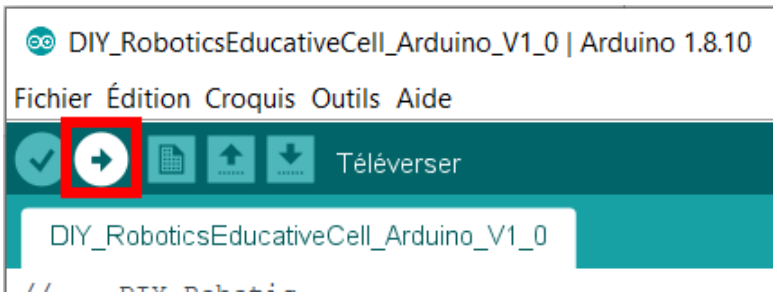
Open the **DIY_ROBOTICSEDUCATIVECELL_Arduino_V1_0.ino** file included in the **DIY_ROBOTICS_EDUCATIVECELL_V1_0.zip** compressed folder:



Connect the Arduino Micro to your computer with the USB cable.
Select the type **Arduino / Genuino Micro** and the correct communication port:

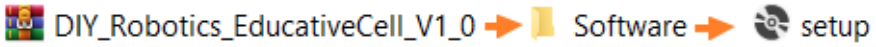


Program the Arduino Micro by pressing the **Upload** button.



3. DIY Robotics Educative Cell V1.0 software setup

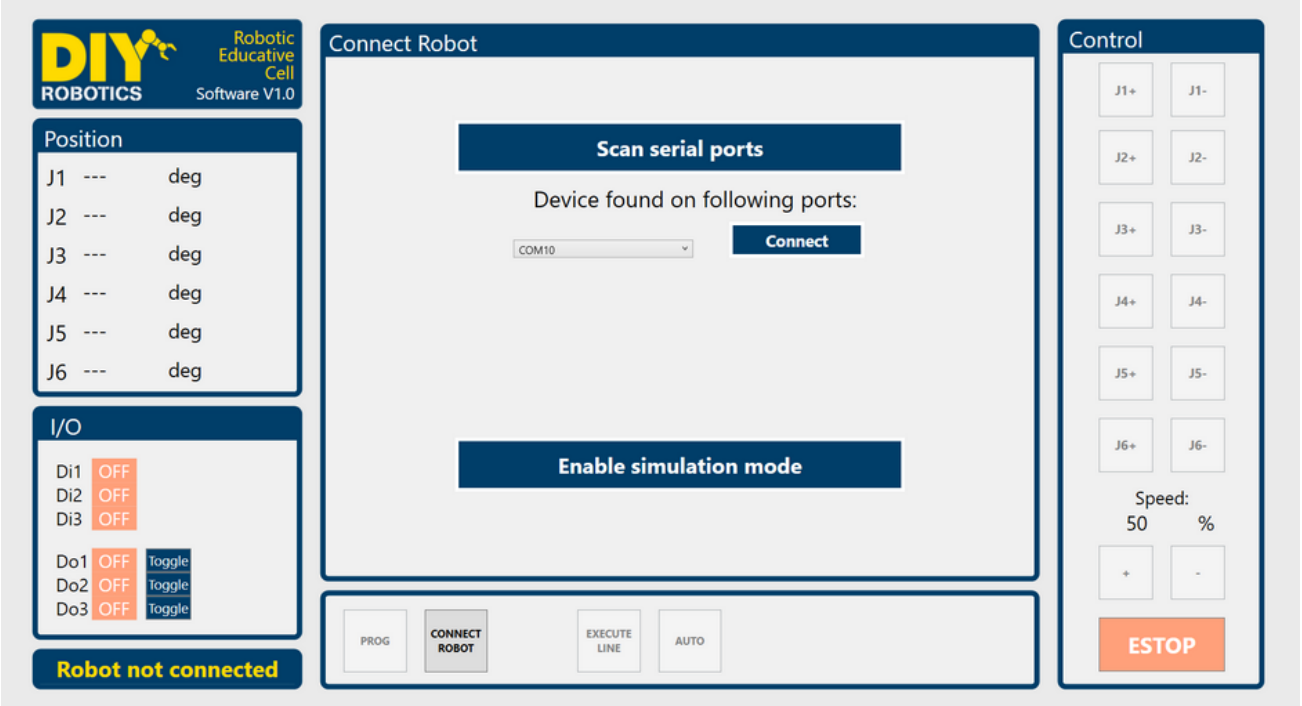
Open the **setup** file included in the **DIY_ROBOTICS_EDUCATIVECELL_V1_0.zip** compressed folder:



Follow the installer instructions to complete the installation.

When the installation is complete, run the software by clicking on the **DIY Robotics** icon on your desktop.

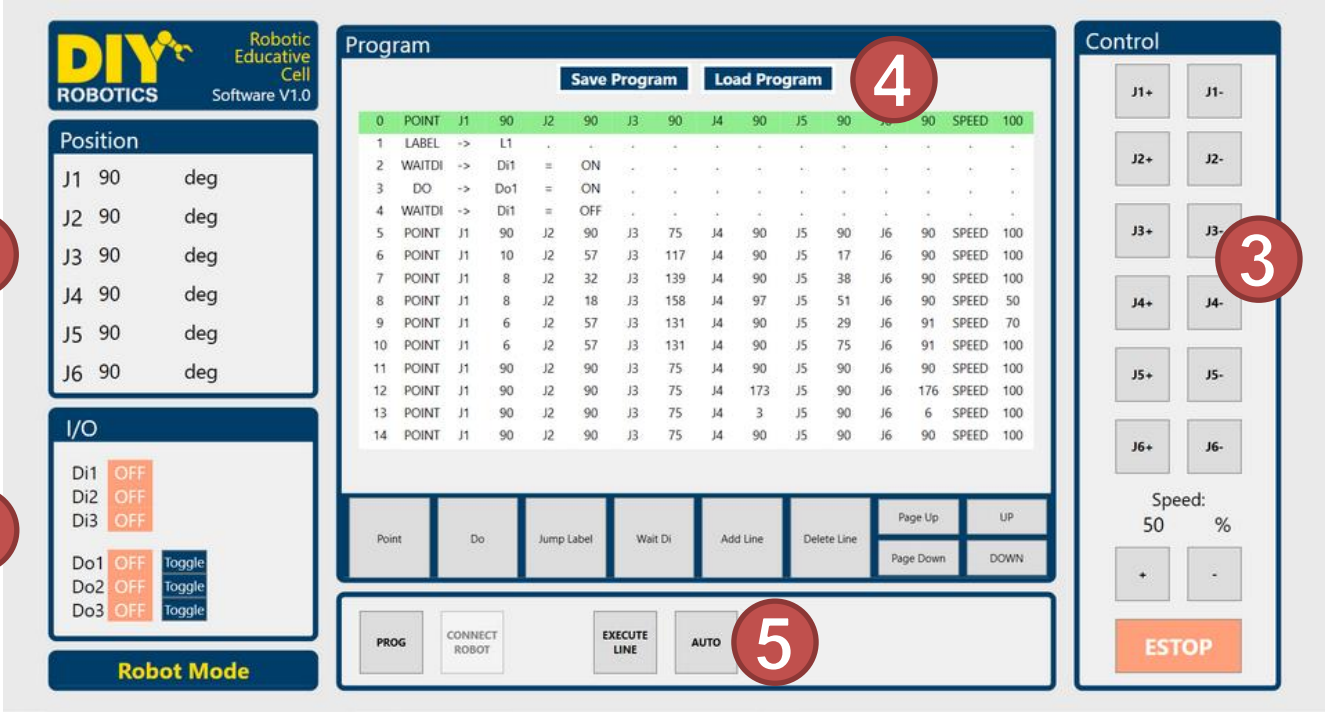
4. Navigating the interface



When the software opens, the robot connection panel is displayed.

It is then possible to search for the device corresponding to the robot controller by pressing the **Scan serial ports** button. If the robot controller is connected to the computer by its USB cable, its communication port will appear in the drop-down list. Select the correct port then press the **Connect** button. If the connection to the robot is successful, the robot position values will be displayed and the robot control buttons will be enabled. At the bottom of the software page, the **PROG** button then gives access to the programming panel.

If you want to navigate in the software without connecting to the robot, press the **Enable simulation mode** button to activate the simulator mode. You will then have access to the programming panel.

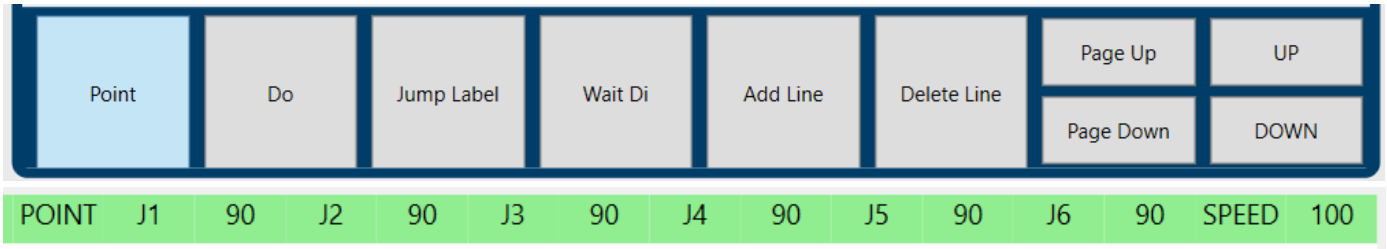


1	Position panel	This panel displays the position of each robot joint in degrees. **Jx** stands for "Joint x".
2	I/O panel	This panel displays the status of the I/O (input/output) signals from the robot controller **P3** and **P4** terminal blocks. Terminal block **P3** is used to connect 3 input signals (**Di**). The states of these signals are displayed in this panel. Terminal block **P4** is used to connect 3 output signals (**Do**). The states of these signals are displayed in this panel. **Do** output signals can be switched by pressing the **Toggle** buttons in this panel.
3	Control panel	This panel contains the buttons that allow you to individually control each joint of the robot. This panel contains two buttons per joint: a positive rotation button **Jx +** and a negative rotation button **Jx-**. **+** And **-** buttons are used to adjust the **speed** of movement. At the bottom of the panel, an **ESTOP** emergency stop button is used to stop the movement commands sent to the robot from the software.
4	Programming panel	The programming panel allows you to create robot programs. At the top of the panel, a **Save program** button allows you to save a robot program as a **.diy** file on your computer. A **Load program** button opens a **.diy** robot program previously saved on your computer. In the center of the panel are displayed 15 lines of the program being created. Each line corresponds to an instruction. The active line is highlighted in green. A running line will be displayed in yellow. At the bottom of the panel, there are buttons for adding instruction lines. It is also possible to add and delete a new line. The **UP**, **DOWN**, **Page Up** and **Page Down** buttons are used to navigate in the robot program.
5	Pages panel	This panel gives access to the programming and connection panels by the **PROG** and **CONNECT ROBOT** buttons. The **EXECUTE LINE** button is used to execute the active line (highlighted in green) of the robot program. The **AUTO** button is used to execute the entire robot program.

5. Creation of a robot program

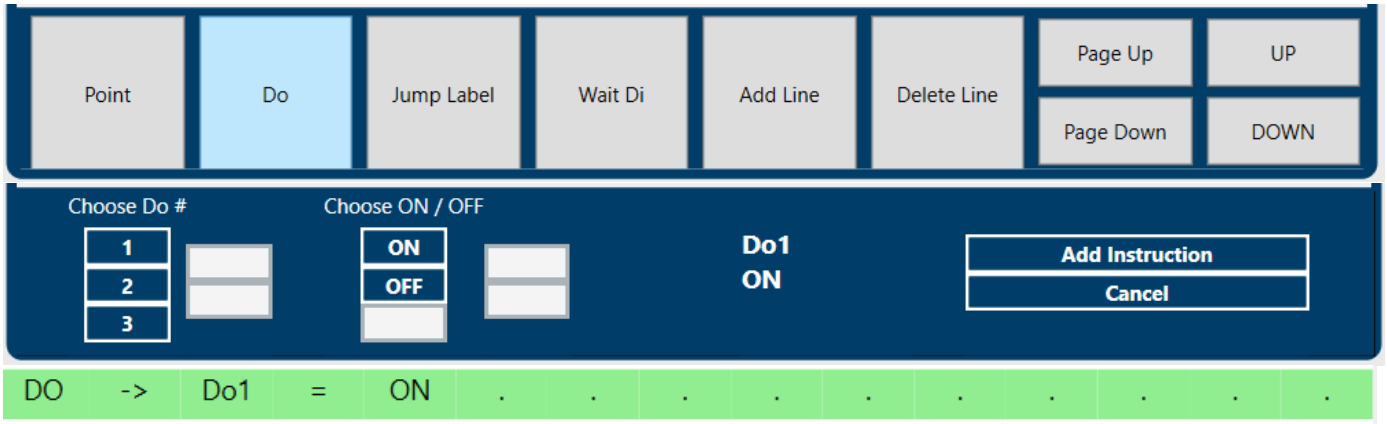
The programming panel allows you to create a robot program with up to 200 instruction lines. Here is a description of each type of instruction:

POINT instruction



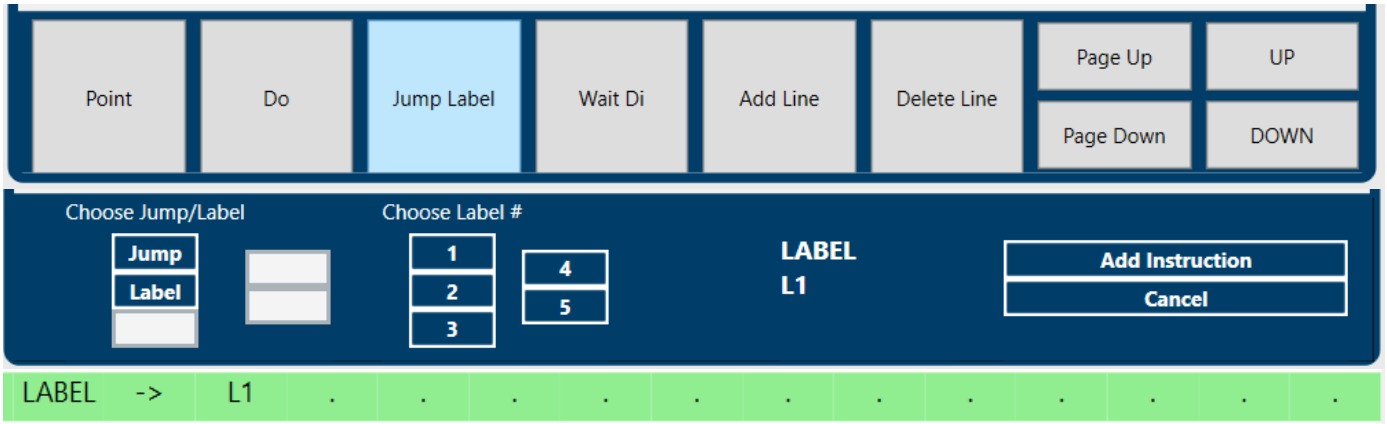
Saves a robot point (position).
Executing this instruction will move the robot according to the saved position and speed.
To save a robot point in an instruction, manually move the robot at the desired position and select the desired movement speed using the buttons in the control panel. Press the **Point** button. An instruction line is then added in the programming panel. The instruction line shows the value in degrees of each joint as well as the speed of movement.

DO instruction



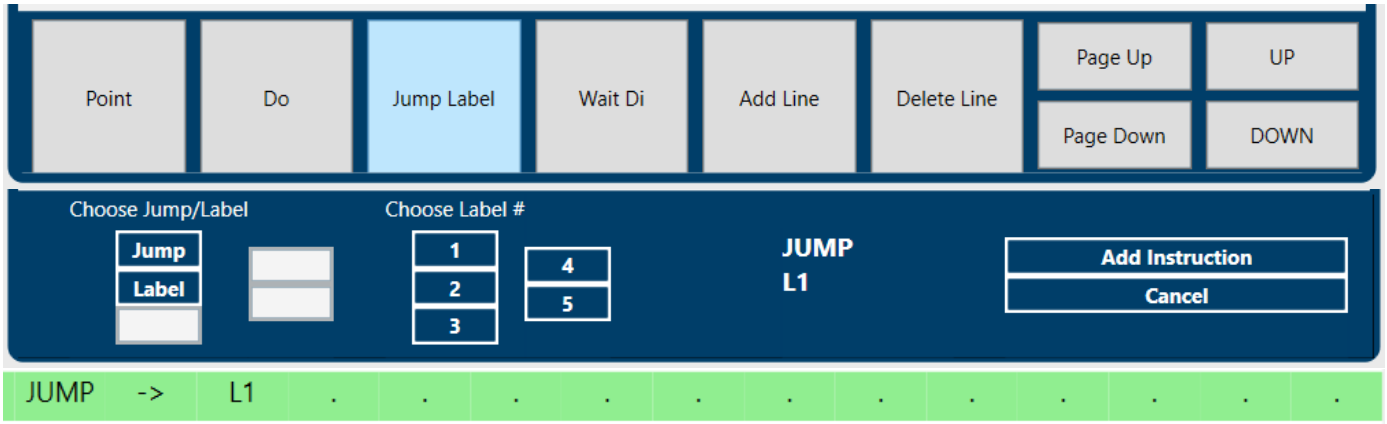
Changes the state of a **Do** output signal.
Executing this instruction will change the state of one of the output signals **Do** (ON/OFF).
To create a **DO** instruction, press the **Do** button. A parameter panel is displayed. Choose the **Do** output signal number (1, 2 or 3) as well as the desired state (ON or OFF). Press the **Add Instruction** button to add the instruction.
An instruction line is then added in the programming panel. The instruction line shows the **Do** signal number and the change of state.

LABEL instruction



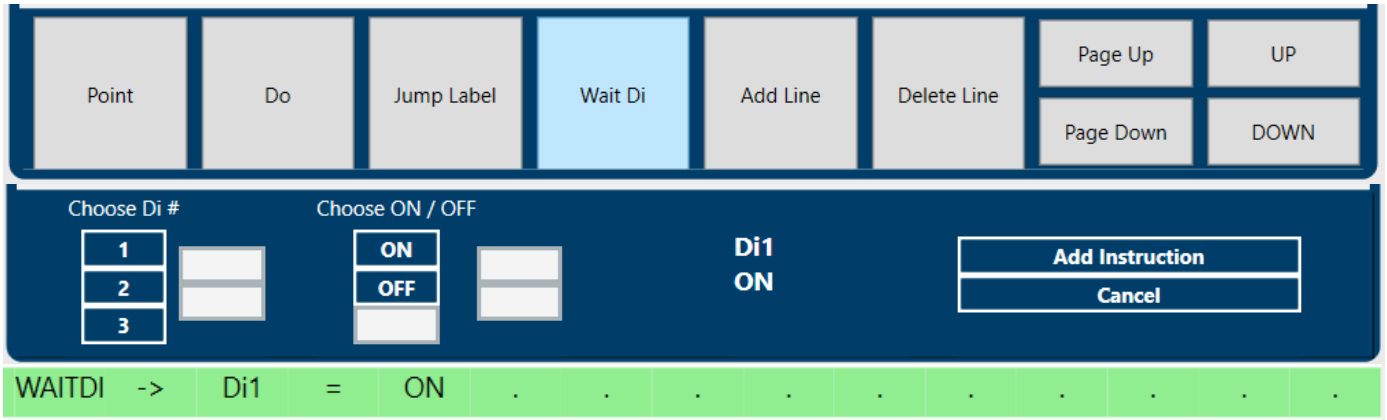
Adds a label in the robot program.
Executing this instruction will have no effect. This line is a label that will allow the **JUMP** instruction to jump to this **LABEL** instruction line.
To create a **LABEL** instruction, press the **Jump Label** button. A parameter panel is displayed. Choose the Label option and the number of the desired label (1 to 5). Press the **Add Instruction** button to add the instruction.
An instruction line is then added in the programming panel. The instruction line shows the label number.

JUMP instruction



Jumps to the program line that contains the corresponding label.
Executing this instruction will cause a jump in the program to the line that contains the corresponding label.
To create a **JUMP** instruction, press the **Jump Label** button. A parameter panel is displayed. Choose the Jump option and the number of the desired label (1 to 5). Press the **Add Instruction** button to add the instruction.
An instruction line is then added in the programming panel. The instruction line indicates the number of the target label.
If several labels have the same number, the **JUMP** instruction will jump to the first corresponding label from the top of the program.
If there is no label corresponding to the **JUMP** instruction number, the program will jump to the last line of the program.

WAITDI instruction



Waits for a specific state of a **Di** input signal.
Executing this instruction will put the robot controller on hold as long as the state of the **Di** input signal is different from the expected state.
To create a **WAITDI** instruction, press the **Wait Di** button. A parameter panel is displayed. Choose the **Di** input signal number (1, 2 or 3) as well as the desired state (ON or OFF). Press the **Add Instruction** button to add the instruction.
An instruction line is then added in the programming panel. The instruction line indicates the Di input signal number and the expected state.

Add Line button

Add a blank line in the robot program above the active line (highlighted in green).

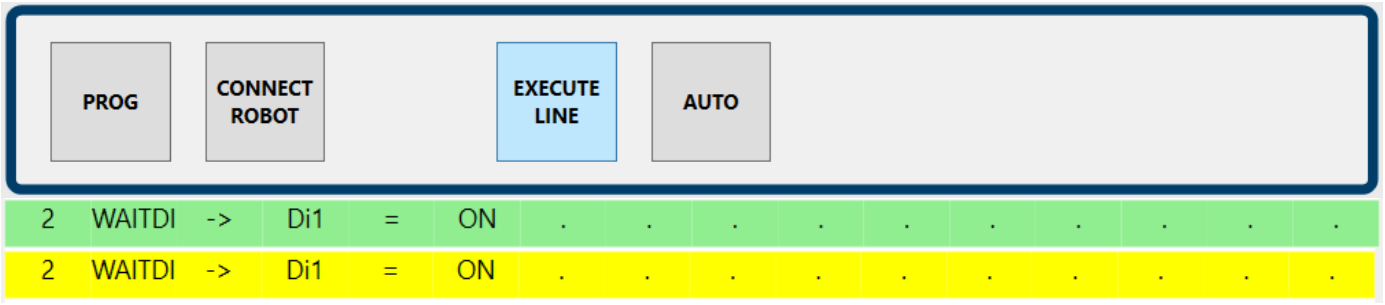
Delete Line button

Delete the active line (highlighted in green).

Page Up, Page Down, UP, DOWN buttons

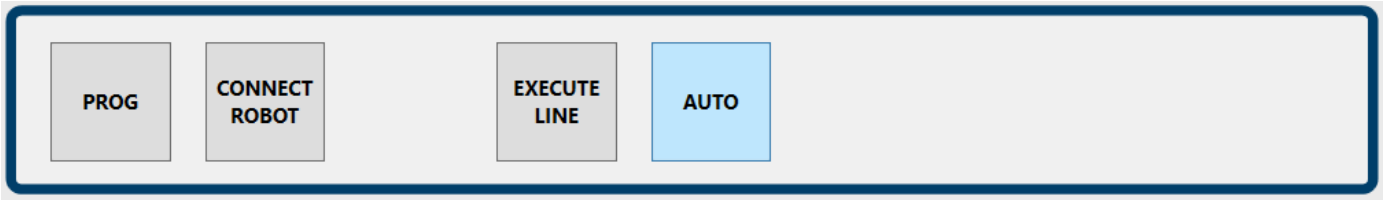
These buttons allow you to navigate in the robot program. The **Page Up** and **Page Down** buttons increment and decrement the active line of the program by 10. The **UP** and **DOWN** buttons increment and decrement the active line of the program by 1.

EXECUTE LINE button



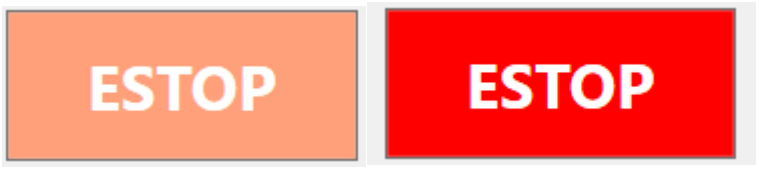
The **EXECUTE LINE** button is used to execute the active line of the robot program (highlighted in green).
The active line turns yellow when it is running.
When the execution is finished, the next line becomes active.

AUTO button



The **AUTO** button successively executes all the lines of the robot program. By pressing this button, the active line automatically switches to the first line of the robot program and then the execution of the lines begins.

ESTOP button



The **ESTOP** emergency stop button is used to stop the commands sent to the robot from the software. When a line in the program is running, press this button to stop program execution. When the emergency stop is activated, its button turns dark red.
The control panel buttons and the navigation in the software are disabled when the emergency stop button is activated.
To deactivate the emergency stop, press the button a second time. It turns pale again.

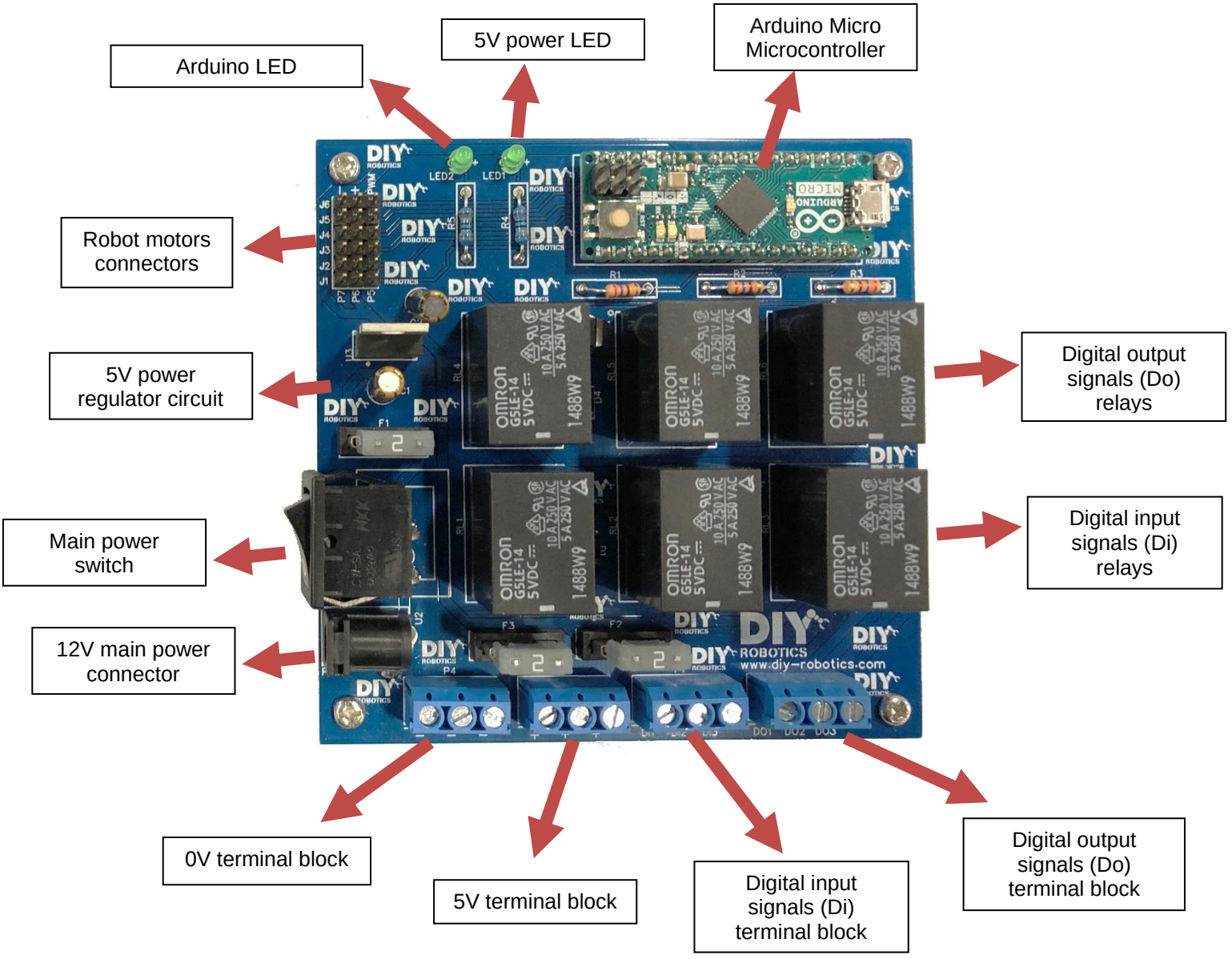
This software ESTOP button has no impact on the power supplied to the motors and to the input/output signals of the robot controller. If you need to turn off the robot controller power, use the physical switch SW1 on the robot controller instead and disconnect the USB cable.

ELECTRICAL

1. Overview

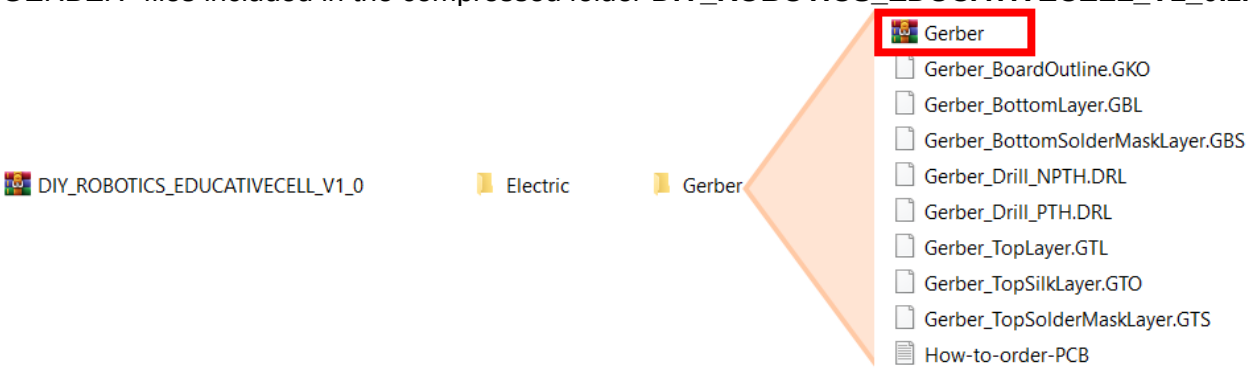
The electronic controller of the robotic educative cell is the bridge between the programming software and the robot. The microcontroller used on the printed circuit, the Arduino Micro, performs the following tasks:

- Communication between the electronic controller and the programming software
- Control of the 6 robot motors (5V servo motors)
- Control of 3 digital output signals (0-5V logic levels)
- Reading of 3 digital input signals (0-5V logic levels)



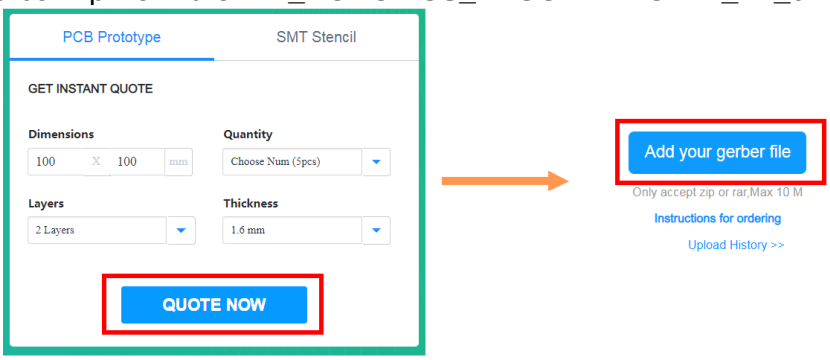
2. *Printed circuit board (PCB) order*

The printed circuit board (PCB) of the robot controller can be ordered from any PCB manufacturer with the "GERBER" files included in the compressed folder **DIY_ROBOTICS_EDUCATIVECELL_V1_0.zip**:

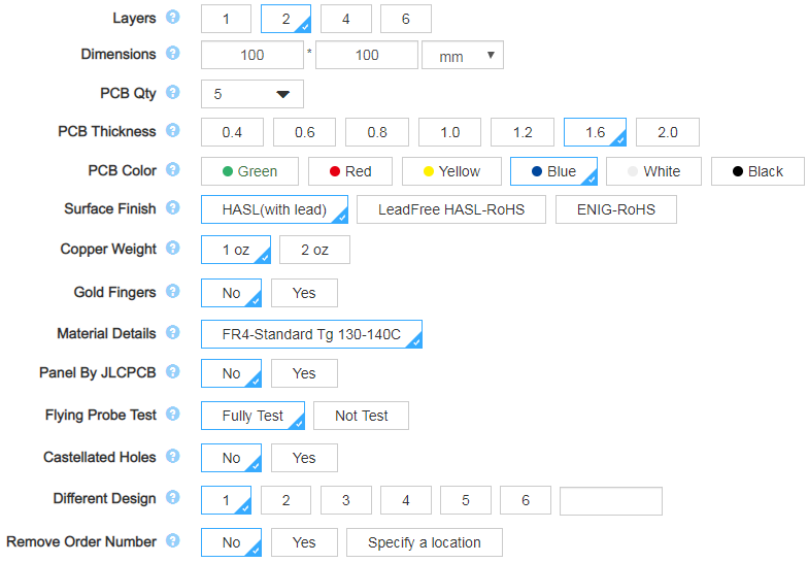


We suggest you to order from the manufacturer **JLCPCB** (jlcpcb.com) which offers a fast, simple service at a very low price. Follow the following steps to order the PCB:

1. On the jlcpcb.com home page, select **QUOTE NOW**, then **Add your gerber file**.
   Select the Gerber.zip file in the **DIY_ROBOTICS_EDUCATIVECELL_V1_0.zip** compressed folder.



2. Select the following parameters:



3. Select **Save to cart** and proceed with payment to complete the order.

3. Printed circuit board (PCB) assembly

Once the robot controller PCB in hands, proceed to its assembly. You will have to solder all the components.

Each component of the PCB is identified.
Refer to the **bill-of-materials.pdf** material list included in the **DIY_ROBOTICS_EDUCATIVECELL_V1_0.zip** compressed folder:

DIY_Robotics_EducativeCell_V1_0 → bill-of-materials

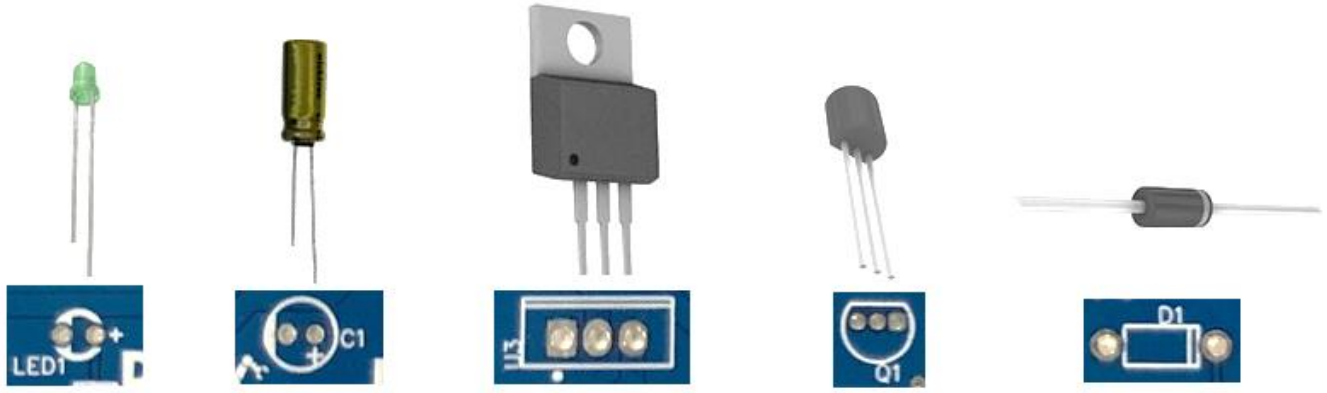
BOM - ServoPCB-1

ID	Name	Designator	Quantit	Manufacturer Part	Manufa
1	ARDUINO MICRO	U1	1	Arduino Micro	Arduinc
2	RELAY GEN PURPOSE SPDT 10A 5V	RL2,RL1,RL3,RL4,RL5,RL6	6	G5LE-14 DC5	Omron
3	Diode 1N4007	D2,D1,D3,D4,D5,D6	6	1N4007	Micro C
4	TERM BLK 3P SIDE ENT 5.08MM PCB	P1, P2, P3, P4	4	OSTTC032162	On Sho
5	TRANS NPN 40V 0.2A TO-92	Q1,Q2,Q3	3	2N3904BU	ON Ser
6	RESISTOR 270 OHM 1/4W 5% AXIAL	R1,R2,R3	3	CF14JT270K	Stackpc
7	FUSE BLOCK BLADE 500V 20A PCB	F1,F2,F3	3	3544-2	Keystor
8	CONN JACK R/A PCB 5.5X2.1MM	U2	1	54-00133	Tensilit
9	CAP ALUM 10UF 20% 50V RADIAL	C1,C2	2	UFW1H100MDD1TD	Nichico
10	IC REG LINEAR 5V 3A TO220-3	U3	1	LM1085IT-5.0	TI
11	CONN HEADER VERT 12POS 2.54MM	P5,P6,P7	2	TSW-104-07-L-T	Samtec
12	RES 140 OHM 1/4W 1% AXIAL	R4,R5	2	MFR-25FBF52-140R	Yageo
13	LED GREEN DIFFUSED 3MM ROUND T/H	LED1,LED2	2	151031VS06000	Wurth I
14	SWITCH ROCKER SPST 6A 250V	SW1	1	CWSA11AAN1H	NKK Sw
15	FUSE AUTO 2A 32VDC BLADE MINI	F1,F2,F3	3	0297002.WXNV	Littelfus
16	9G METAL GEAR MICRO SERVO (1.8KG)	NA	2	SER0039	DFRobo
17	AC/DC WALL MOUNT ADAPTER 12V 25V	NA	1	SWI25-12-N-P5	CUI Inc.
18	USB A MALE TO MICRO B MALE 3'	NA	1	3025010-03	Qualtek
19	RES 10K OHM 1/4W 5% AXIAL	NA	3	CF14JT10K0	Stackpc
20	Digital Metal Gear Servo For Robot	NA	4	MG996R	NA
21	DIY-Robotics PCB	NA	1	NA	NA

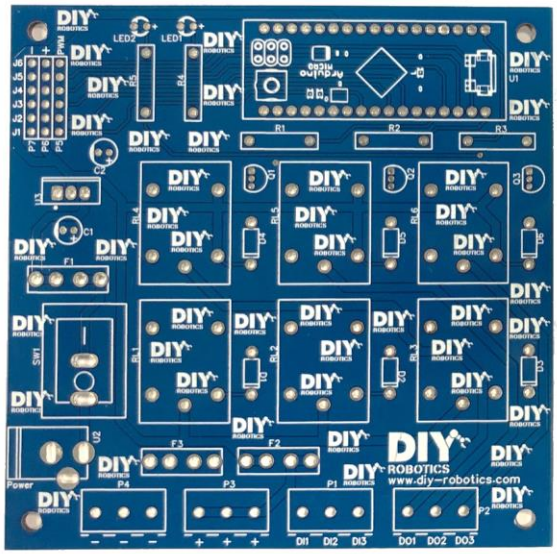
Pay particular attention to the polarity of the following components:

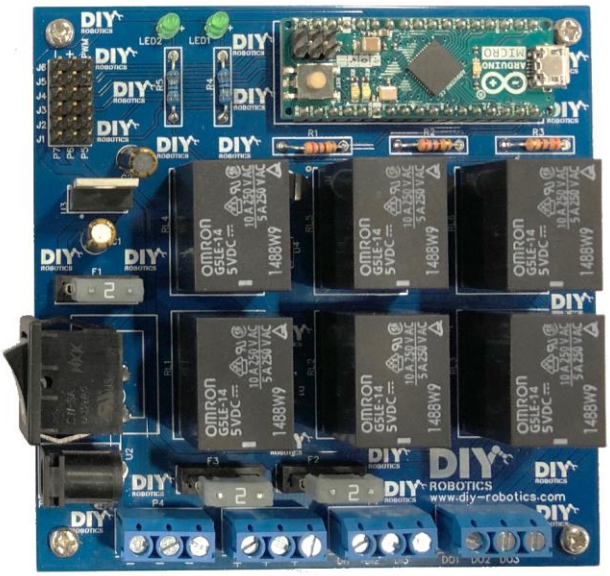
LED1, LED2, U1, U3, C1, C2, D1, D2, D3, D4, D5, D6, Q1, Q2, Q3

These components must be soldered the right way, otherwise they will burn. For example, notice that the light emitting diodes (LEDs) and the capacitors (C) have a long pin and a short pin. The long pin, the anode, must be inserted and soldered into the hole identified by a **+**.
Use the following image to solder these components the right way:



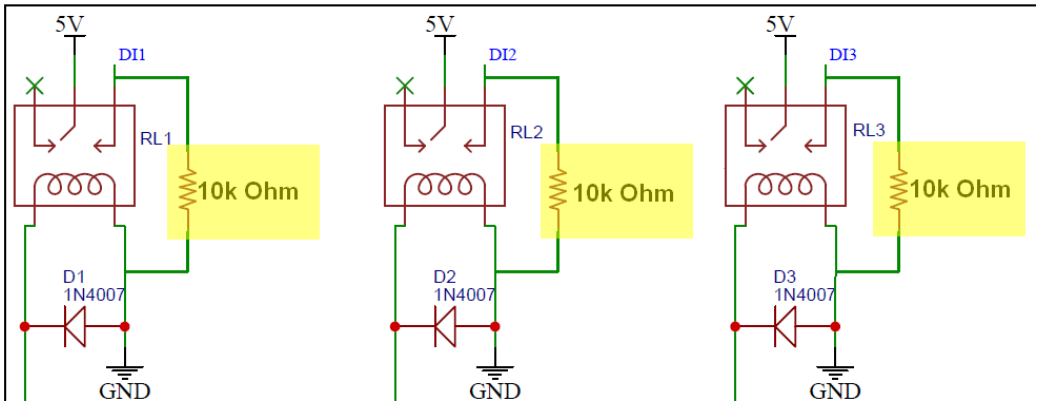
Use the following image as example:

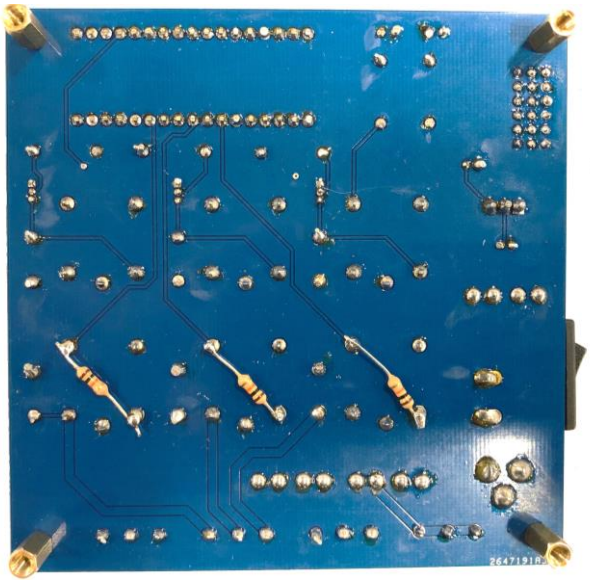


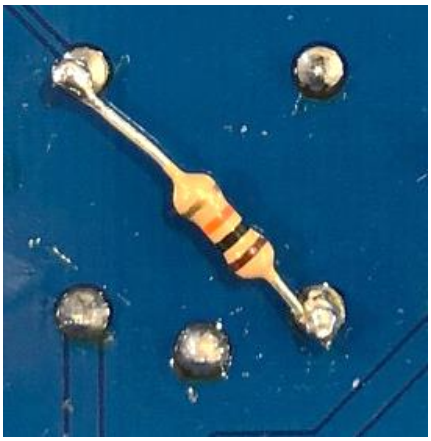


Finally, 3 resistors of 10k Ohm must be added to the circuit in order to make the digital input signals (**Di**) functional. These resistors are described as follow in the material list:

RES 10K OHM 1/4W 5% AXIAL







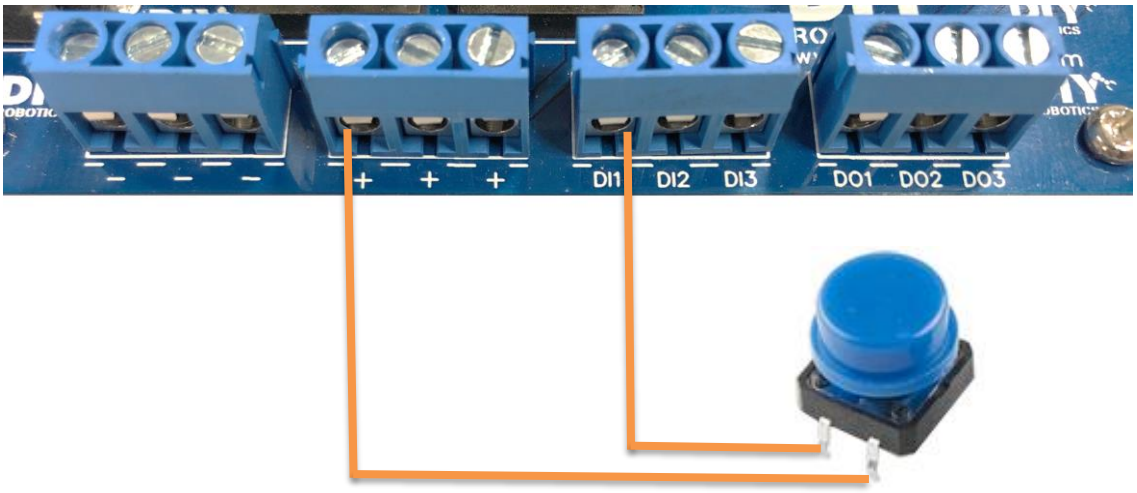
4. I/O signals - Examples

P1, **P2**, **P3** and **P4** terminal blocks of the controller allow devices (buttons, LEDs, sensors) to be connected to the robot using 0-5V logic levels.

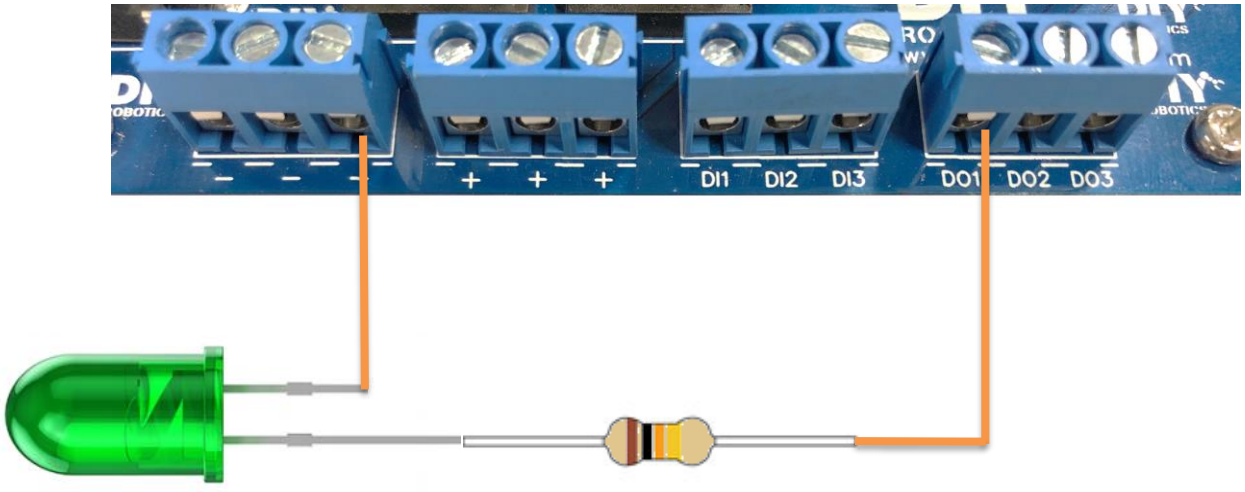
The 0V **P1** terminal block is protected by a 2A fuse.
The 5V **P2** power supply terminal block is protected by a 2A fuse.
The inputs of the **P3** digital input signals terminal block are directly connected to 5V relays. If a voltage greater than 5V is applied to this terminal block, the relays may be damaged.
The outputs of the **P4** digital output signals terminal block are not directly protected. A short circuit on this terminal block could damage the 5V voltage regulator of the controller.

Here are two examples of device connections on the input/output signal terminal blocks:

1. **Button on Di1**



2. **LED on Do1**

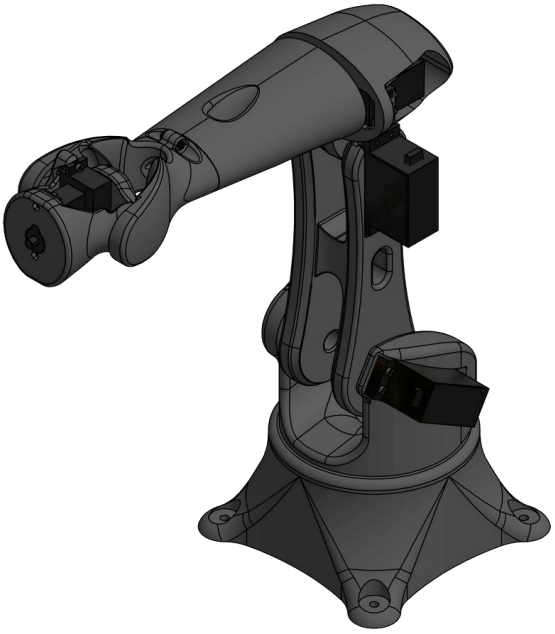


MECHANICAL

1. Overview

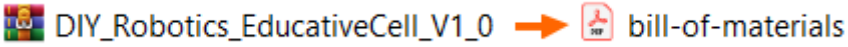
In order to mechanically assemble your robot, you will need the following components and tools:

- 4 MG966R servo motors
- 2 9g Micro servo motors
- 8 3D printed robot parts
- 24 metric M2 nuts
- 24 metric M2 bolts
- 2 metric M2.5 bolts
- 4 metric M3 bolts



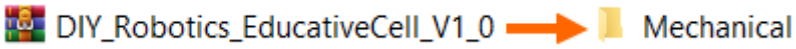
- 3D printer
- Soldering iron
- Lighter
- Hex keys

Refer to the **bill-of-materials.pdf** material list included in the **DIY_ROBOTICS_EDUCATIVECELL_V1_0.zip** compressed folder:



2. 3D printing

You will find the 3D files of the 8 robot parts in the compressed folder **DIY_ROBOTICS_EDUCATIVECELL_V1_0.zip** :



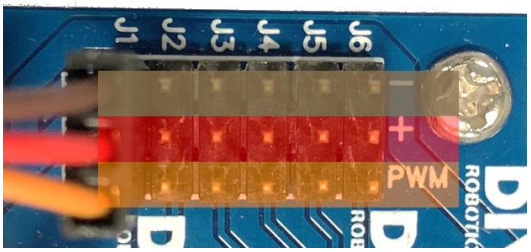
Print the parts using a 3D printer. We recommend that you use the following settings:

Top layer	4 layers
Bottom layer	4 layers
Wall	4 layers

3. Align the servos

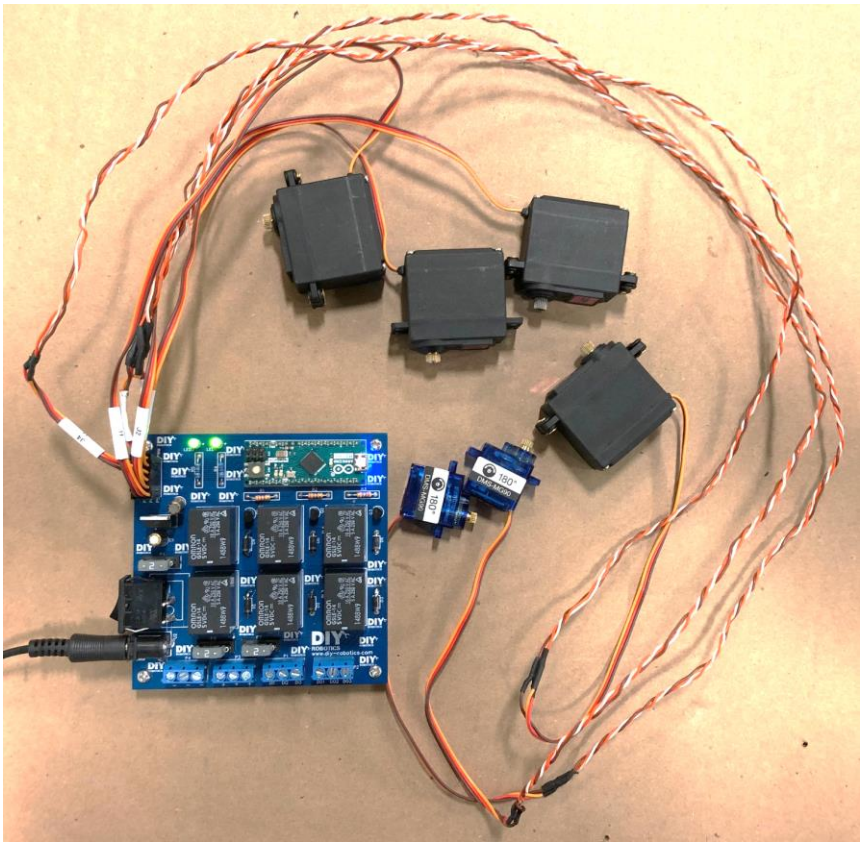
Before assembling the robot, it is important to make sure that all the servo motors are at midpoint. In order to align the servos, make sure you have previously programmed the Arduino microcontroller and assembled the robot controller. Follow the instructions below to align the servo motors:

Connect the 6 servo motors to the robot controller.
Make sure the connectors are plugged in the correct way.



Brown wire : **0V (-)**
Red wire : **5V (+)**
Orange wire : **PWM**

Connect the 12V regulator to your 120V AC wall outlet.
Connect the 12V regulator to the power connector of the robot controller.
Activate the power switch SW1.
The LED1 light should turn on and the LED2 light should flash.
The robot should position all of its servo motors at 90 degrees.



You can turn off the power on the robot controller and disconnect the servo motors.

4. Insert the nuts

Before assembling, insert a M2 x 0.4mm nut into each hexagonal hole of the 3D printed parts to allow assembly.
Use a soldering iron to facilitate insertion:

5. Cast the gears in the junction holes

The mechanical junction between the servo motors and the 3D printed robot parts are direct: the gear must be inserted directly into the hole.
To ensure a good mechanical junction, the holes are slightly smaller than the gears after 3D printing.
With a lighter, slightly heat the hole, then insert the gear of a servo motor (as straight as possible). The melted plastic will take the form of a gear. Complete the insertion by gently tightening a bolt. Repeat this step for each junction.

Be careful, overheating 3D printed parts could deform them and make them unusable.

6. Assembly

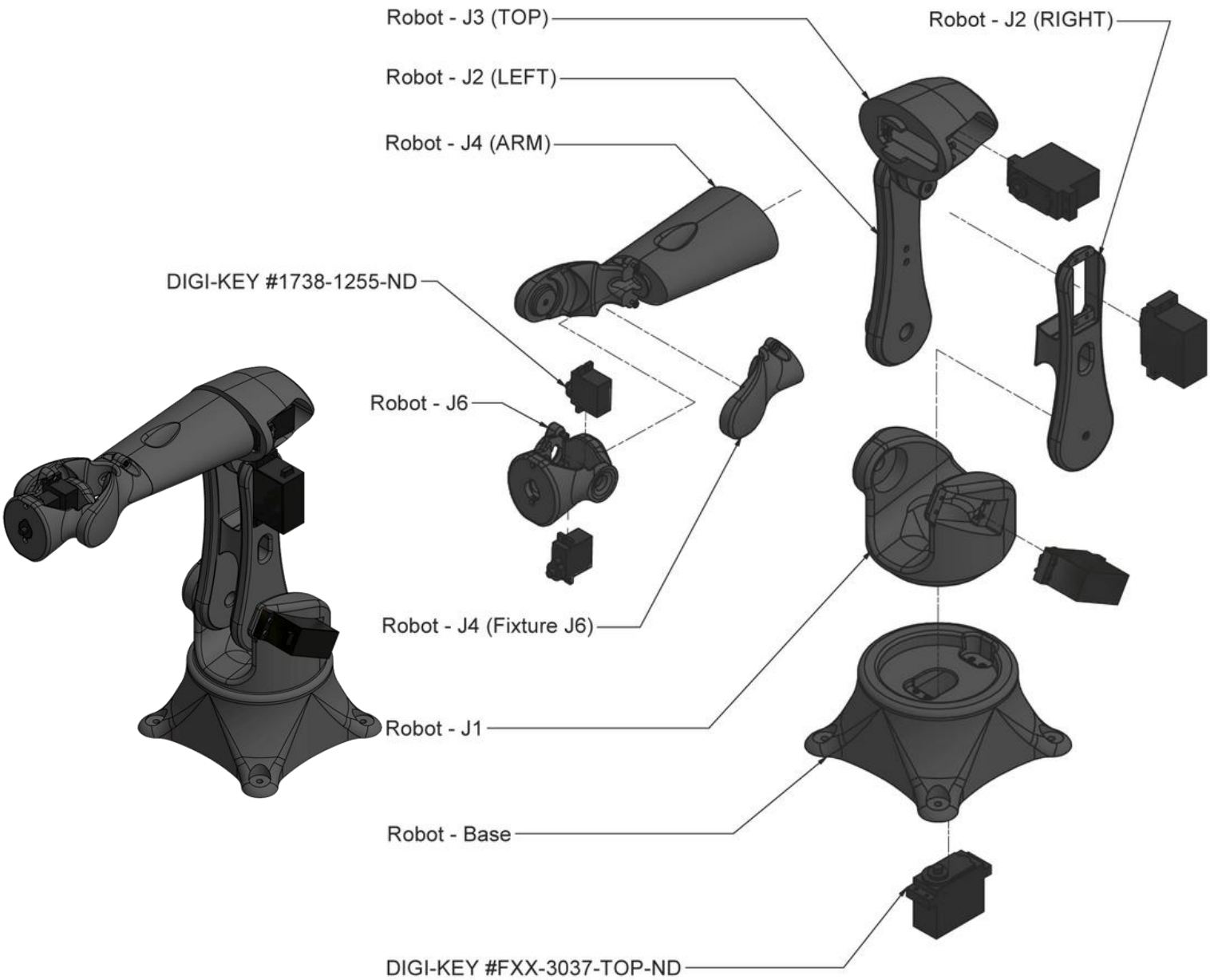
Use M3 metric bolts to attach the servo motor gears to the 3D printed robot parts.

Use M2 metric bolts to attach the servo motor housings to the 3D printed robot parts.

Use M2 metric bolts to assemble the two 3D printed robot parts from J2 to J4.

Assemble the robot so that each joint is at its mid-point (straight robot, as illustrated below).

Rely on the image below to assemble all the pieces of the robot.



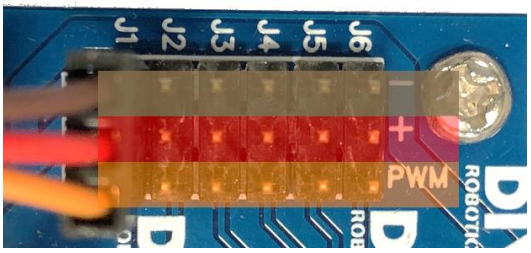
USING THE ROBOT

- Robot mechanical assembly
- Arduino Micro programming
- **DIY Robotics Educative Cell V1.0** software setup
- Robot controller (PCB) electrical assembly

If these steps have been completed, you are now ready to use your robot.

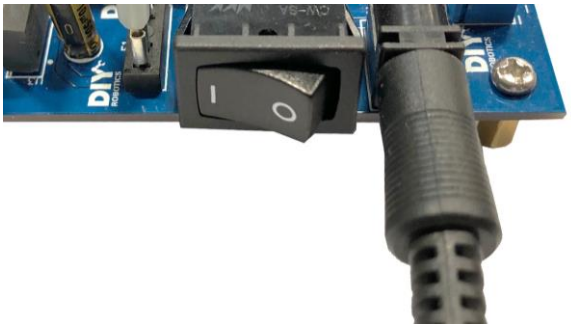
1. Electrical connections

Connect the 6 servo motors of the robot to the robot controller.
Make sure the connectors are plugged in the correct way.



Brown wire : **0V (-)**
Red wire : **5V (+)**
Orange wire : **PWM**

Connect the 12V regulator to your 120V AC wall outlet.
Connect the 12V regulator to the power connector of the robot controller.
Activate the power switch SW1.
The LED1 light should turn on and the LED2 light should flash.
The robot should position all of its servo motors at 90 degrees.

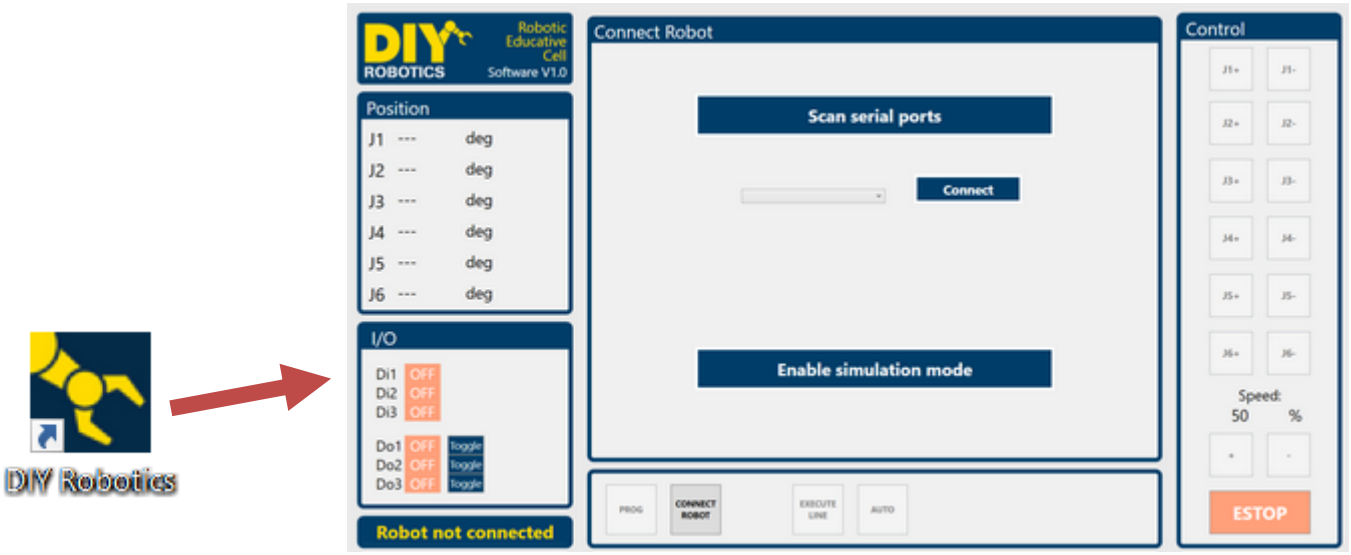


Connect the USB cable from the robot controller to your computer.

The robot controller can operate from the USB port 5V power from your computer. However, the power available from USB ports varies from computer to computer and is often limited to less than 1A. For proper operation of the robot controller, remember to connect the 12V regulator and activate the power switch SW1.

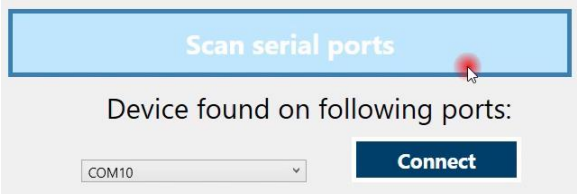
2. Run the software

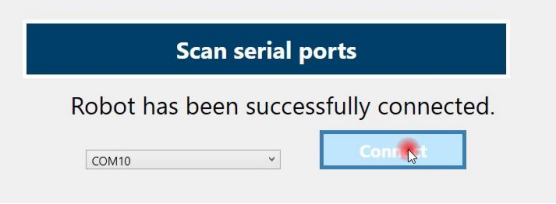
Run the **DIY Robotics Educative Cell V1.0** software by clicking on the **DIY Robotics** icon on your desktop.
The software opens on the connection panel.



3. Set PC <-> Robot serial communication

Press the **Scan serial ports** button.
Select the correct communication port from the drop-down list.
Press the **Connect** button.

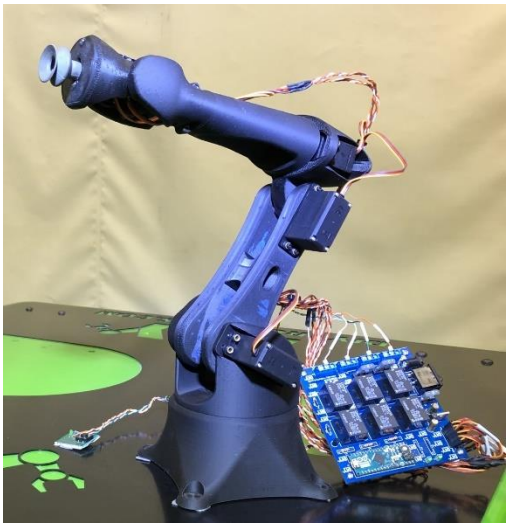




4. Let the creation begin

Control the robot from the control panel.
Create your robot program from the programming panel.

Have fun!



CONCLUSION

Want to go further?

You enjoyed learning about the world of industrial robotics? You're ready to pimp your new robotic arm? Join the **DIY-Robotics Forum** now! The **DIY-Robotics Forum** is a place to talk programming, share ideas and solutions, and work together to build cool stuff in a supportive, smart community.

Need help?

The **DIY-Robotics community** is there to help if you need some support as you build the **DIY-Robotics educative cell**. Subscribe to the **DIY-Robotics Forum** and ask your question to the community.