

Arm Robot

Project 4

# CONTROL 2 SERVOS (FROM 0° TO 180°) WITH 1 JOYSTICK

## **PROJECT N° 4: Control 2 servo motors (from 0° to 180°) with 1 joystick**

Learn how to build and code a circuit to control the position of 2 servo motors using only 1 joystick.

Remember, a joystick has 2 potentiometers, so each potentiometer will be related to a different servomotor. The position of the servomotor will vary from 0° to 180° depending on the movement of the joystick, from left to right.

**DIFFICULTY LEVEL:** Intermediate.

**DURATION OF THE EXERCISE:** 30 min.

**MATERIALS:**

- 1 Joystick
- 2 Servomotors
- 1 Build&Code UNO board
- 1 Build&Code sensor shield
- 1 USB - USB type B cable
- Computer

**CONNECTIONS:**

1. Connect the joystick to the analog ports A0 and A1 of the Build&Code 4in1 board.

**Connections to the analog port A0:**

- Connect the GND to the G pin.
- Connect the Vcc to the V pin.
- Connect the VRx to the S pin.

**Connections to the analog port A1:**

- The VRy to the S pin.

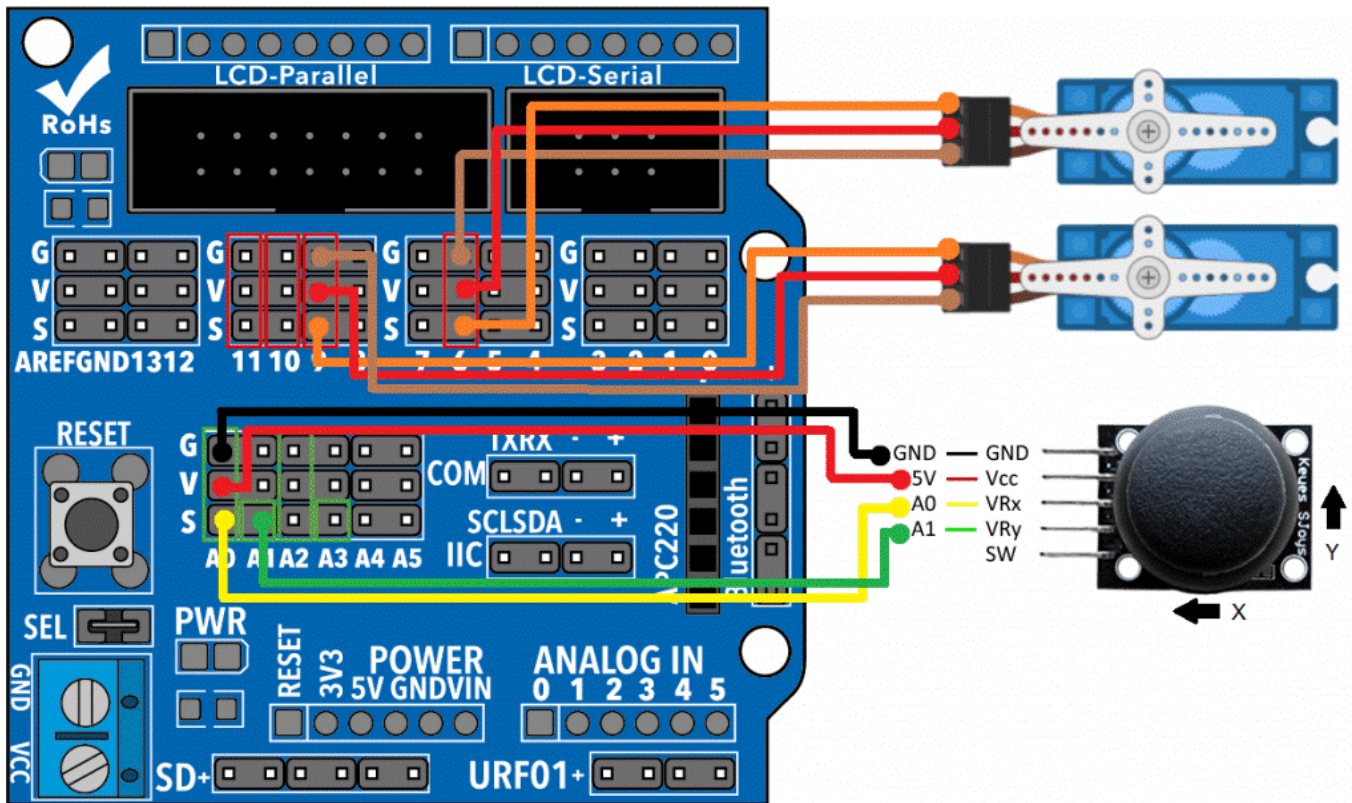
As you remember, we've connected the joystick to the analog ports A0 and A1 because it has two potentiometers, which are analog sensors.

2. Connect the servomotors to the Build&Code 4in1 board.

- Connect the servomotor S1 to the digital port 6. Connect the **brown** cable to the G pin, the **red** cable to the V pin and the **orange** cable to the S pin.
- Connect the servomotor S2 to the digital port 9. Connect the **brown** cable to the G pin, the **red** cable to the V pin and the **orange** cable to the S pin.

We've connected the servomotors to the digital ports 6 and 9 because they have a PWM output. This output will be in charge of making the servomotor rotate more or less depending on the values received from the joystick.

3. The following image shows the connections. Use it as a guide.



## PROGRAMMING CODE

You can do this project using the Arduino program or a visual programming software by blocks compatible. Below you will find the necessary code.

### Arduino Code

You will create a program that will be constantly reading information from the 2 joystick outputs, and basing on the reading received from the joystick, it will make the servomotor move to one position or the other. When you release the joystick, the servomotor will go back to its original position.

Before start, remember that:

- A potentiometer is an analog sensor, so it will make readings from 0 to 1023, basing on how much you have rotated it.
- The digital pins PWM will be working as output pins, but instead of giving a permanent high or low value, they will be able to make very quick changes in order to give medium values too. This will make possible to send different levels of output voltage.

- To work with servomotors, you will use the “#include” servomotor library, since it will make it much easier.
- A servomotor can rotate from 0° a 180°, so you will have to relate the analog reading values with the servomotor movement values, using the “map” instruction.

Now, follow this steps:

1. Download and install the Arduino IDE program. It is available for Windows, Mac OS and Linux.
  - Configure and upload the code, following the indications on the Arm Robot First Steps guide.
2. Open the Arduino program and copy the following code in it:

```
#include <Servo.h>

int valueX = 0; // X-AXIS LECTURE
int valueY = 0; // Y-AXIS LECTURE
int pinJX = A0; // X-AXIS CONNECTED TO ANALOG PIN A0
int pinJY = A1; // Y-AXIS CONNECTED TO ANALOG PIN A1
Servo motor1; // DECLARE SERVO 1
Servo motor2; // DECLARE SERVO 2
int degree1 = 0; // SERVO 1 DEGREES
int degree2 = 0; // SERVO 2 DEGREES

void setup() {
  Serial.begin(9600); // ACTIVATE THE COMMUNICATION
  WITH THE BUILD&CODE UNO BOARD
  motor1.attach (6); // PWM 9 DIGITAL PIN CONNECTED TO
  SERVO 1
  motor1.write (0);
  motor2.attach (9); // PWM 10 DIGITAL PIN CONNECTED TO
  SERVO 2
  motor1.write (0);
}

void loop() {

  valueX = analogRead (pinJX); // STORES THE READINGS
  OF THE X-AXIS OF THE ANALOG PORT A0
  valueY = analogRead (pinJY); // STORES THE READINGS
  OF THE Y-AXIS OF THE ANALOG PORT A1
```

```

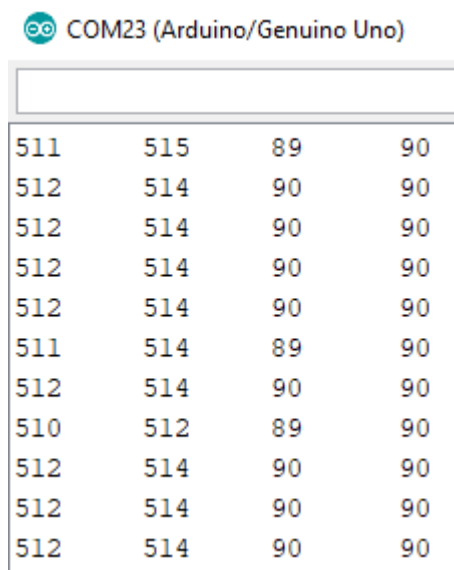
degree1 = map(valorX, 0, 1023, 0, 180);
degree2 = map(valorY, 0, 1023, 0, 180);

motor1.write (grados1); // SEND DEGREES TO SERVO 1
motor2.write (grados2); // SEND DEGREES TO SERVO 2

//Show on the screen the variables we are working
with
Serial.print (valueX); // SHOW ON SCREEN THE X-AXIS
VALUES FROM 0 TO 1023
Serial.print("\t");
Serial.print (valueY); // SHOW ON SCREEN THE Y-AXIS
VALUES FROM 0 TO 1023
Serial.print("\t");
Serial.print (degrees1); // SHOW ON SCREEN THE X-AXIS
VALUES FROM 0 TO 180
Serial.print("\t");
Serial.println(degrees2); // SHOW ON SCREEN THE Y-
AXIS VALUES FROM 0 TO 180

```

Click on the magnifier to see how the variations of the joystick affect the position of the servomotor.

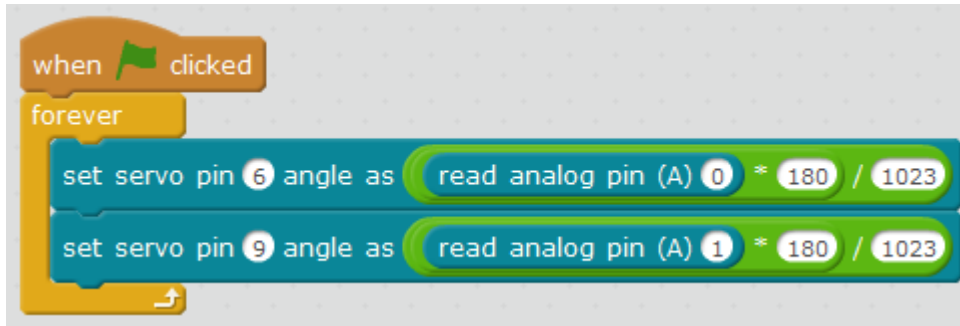


## Code for the visual programming software by blocks compatible

1. Download and install the program.

- Open the software.
- Configure the program to save code into the Build&Code UNO board. You will find the instructions in the Arm Robot First Steps guide.

2. Open the program and copy the following code:



3. To vary the position of the servomotor according to the movement of the joystick, you will do the following mathematical operation:

***Value of the servomotor = (Value of the joystick \* highest value of the servomotor) / highest value of the joystick***

$$\text{Servomotor } S1 = (\text{Reading of the analog port } A0 * 180) / 1023$$

The servomotor works with values from 0 to 180 and the joystick gives values from 0 to 1023. In order to move the servomotor, you have to relate the values of the joystick.

To activate the program you have to configure the control board in a different way. Follow these steps:

1. Go to the Boards section and select Arduino UNO.
1. Go to Connect and select Serial Port. Then, select the COM port to which the Build&Code UNO board is connected.
2. In Connect, select Update Firmware and wait until the communication program is completely downloaded.

Once the program is finished, click on the green flag, move the joystick and watch the Arm Robot moving.

## **RESULT OF THE EXERCISE**

The program will be constantly reading information from the 2 joystick outputs

and, basing on the reading received from the joystick, it will move the servomotors to one position or the other, from  $0^\circ$  to  $180^\circ$ . When you release the joystick, the servomotors will go back to their original position ( $90^\circ$ ).