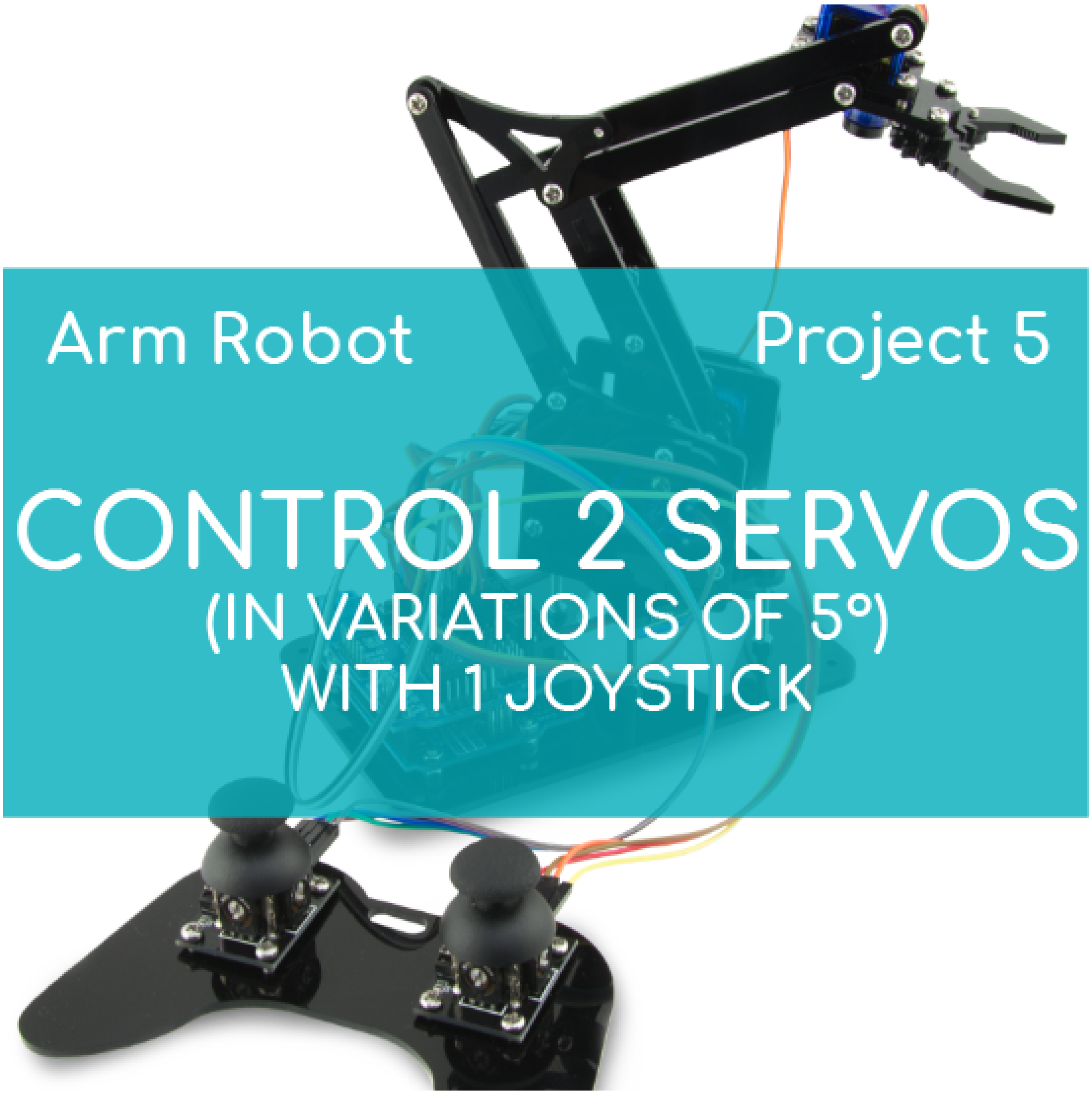


PROJECT Nº 5: Control 2 servo motors (in variations of 5º) with 1 joystick

Build and code a circuit to control the position of 2 servo motor using only 1 joystick.

Remember that a joystick has 2 potentiometers, so each potentiometer will be related to a different servomotor. When moving the joystick up and down you will control one servomotor, and when moving the joystick to the left and to the right you will control the other servomotor.

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.

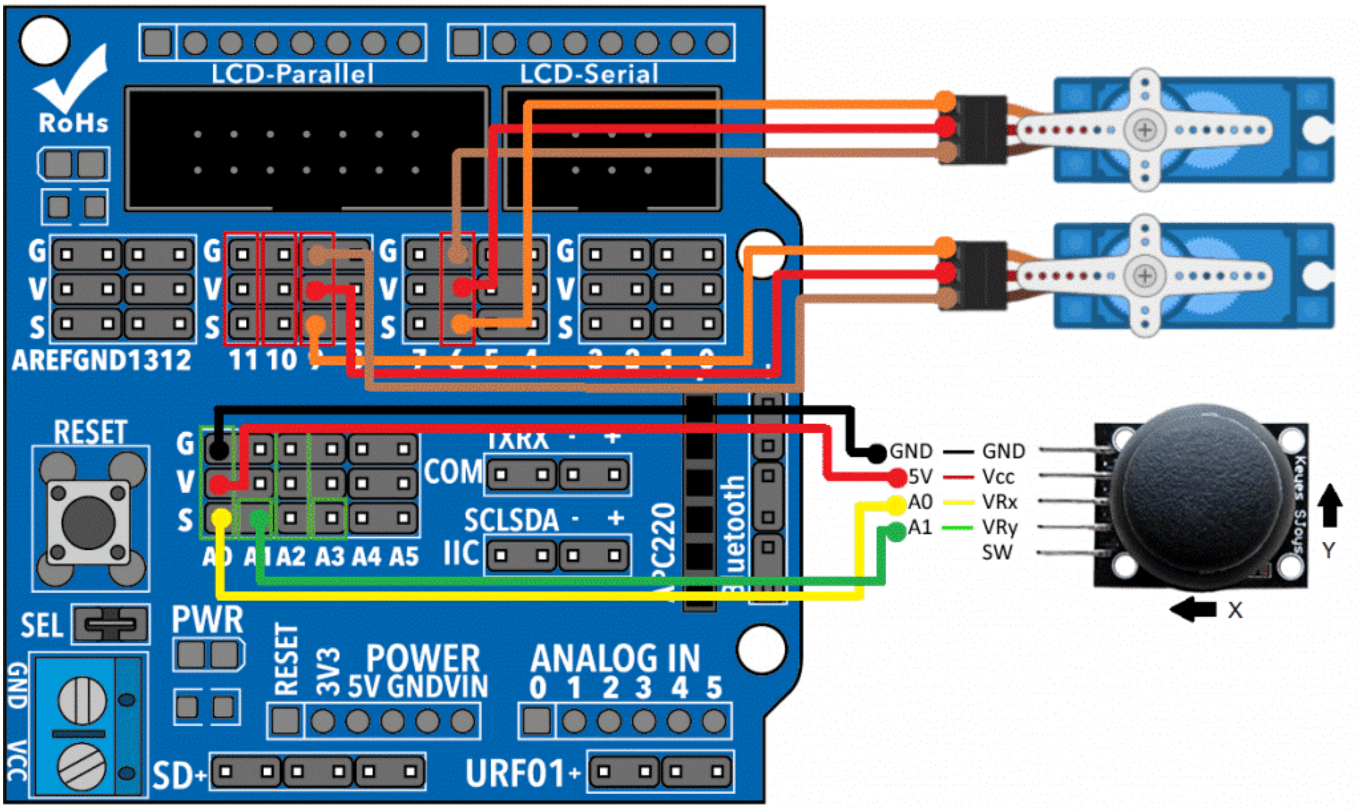
   Connections to the analog port A1:
   - The VRy to the S pin.

   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, with 5º variations. In this case, when you release the joystick, the servomotor will stay in that 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 0º a 180º, so you will have to relate the analog reading values with the servomotor movement values, using the “map” instruction.
- To make the servomotor move to one position or the other, use the conditional instruction “if/else”.
- If the reading value of the X or Y axis or the joystick is higher than 550, you will be moving the joystick forward, so you will have to relate this condition to a positive movement of the servomotor. This means that the position of the servomotor will increase 5º.
- If the reading of the joystick on the X or Y axis is lower than 480, you will be moving the joystick backwards, so you will have to associate this condition to a negative movement of the servomotor. This means that you will be subtracting 5º to the current position of the servomotor.
- If the reading of the servomotor is between 480 and 550, this means you are not moving the joystick. In this case, you don’t have to variate the degree’s variable that modifies the position of the servomotor.

Now, follow this steps:

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

```
#include <Servo.h>

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

void setup() {
Serial.begin(9600); // ACTIVATE THE COMMUNICATION WITH THE BUILD&CODE UNO
motor1.attach (9); // PWM DIGITAL PIN 9 WHERE THE SERVO 1 IS CONNECTED
```

```
motor1.write (0);
motor2.attach (10); // PWM DIGITAL PIN 10 WHERE THE
SERVO 2 IS CONNECTED

motor2.write (0);
}

void loop() {

valueX = analogRead ( pinJX); // A0 ANALOG PORT
READING OF THE X-AXIS

valueY = analogRead ( pinJY); // A1 ANALOG PORT
READING OF THE Y-AXIS

// INCREASE MOTOR1 DEGREES

if ( (valueX > 550 ) && (valueX <= 1023) ) //
INCREASE DEGREES WHEN THE X-AXIS READING IS HIGHER
THAN 550

{
degree1 = degree1 + 1;
}

// DECREASE MOTOR1 DEGREES

if ( (valueX >= 0 ) && (valueX < 480 )) // DECREASE
DEGREES WHEN THE X-AXIS READING IS LOWER THAN 480

{
degree1 = degree1 - 1;
}

else // KEEP THE DEGREE VALUE STABLE MEANWHILE YOU
ARE NOT USING THE JOYSTICK

{
degree1 = degree1;
}

// INCREASE MOTOR 2 DEGREES

if ( (valueY > 550 ) && (valueY <= 1023) ) //
```

```
INCREASE THE DEGREES WHEN THE X-AXIS READING IS
HIGHER THAN 550

{
degree2 = degree2 + 1;

}
// DECREASE MOTOR 2 DEGREES

if ( (valueY >= 0 ) && (valueY < 480) ) // DECREASE
THE DEGREES WHEN THE X-AXIS READING IS LOWER THAN 480

{
degree2 = degree2 - 1;
}

else // KEEP THE DEGREE VALUE STABLE MEANWHILE YOU
ARE NOT USING THE JOYSTICK

{
degree2 = degree2;
}

motor1.write (degree1); // SEND THE DEGREES TO SERVO
1

motor2.write (degree2); // SEND THE DEGREES TO SERVO
2

delay (500);

// PRINT ON THE SCREEN THE VALUES WHICH YOU ARE
WORKING WITH

Serial.print (valueX); // PRINT ON SCREEN X-AXIS
VALUES FROM 0 TO 1023

Serial.print("\t");
Serial.print (valueY);  // PRINT ON SCREEN Y-AXIS
VALUES FROM 0 TO 1023

Serial.print("\t");
Serial.print (degree1);  // PRINT ON SCREEN THE
SERVO1 DEGREES VALUE FROM 0 TO 1023
```

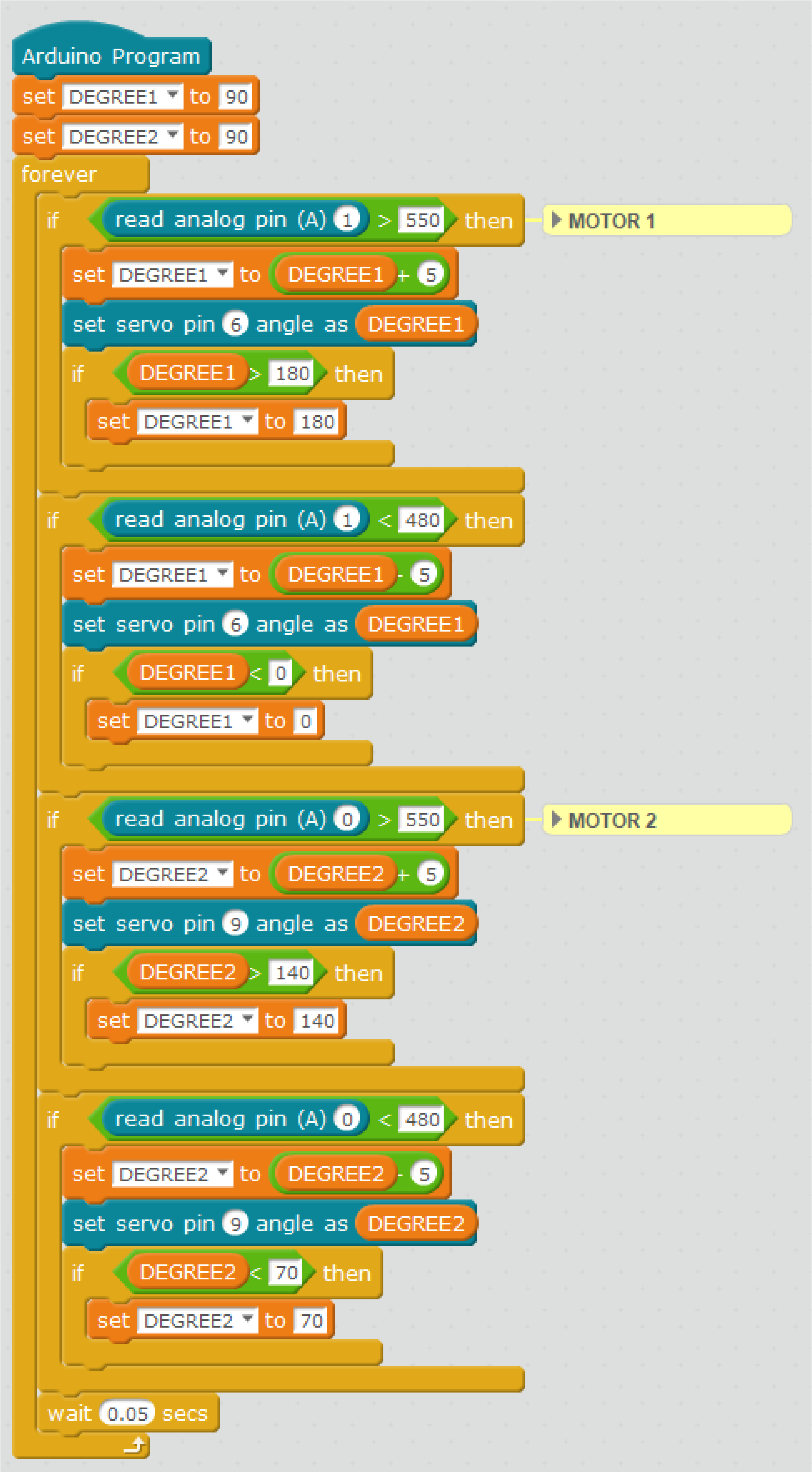
```
Serial.print("\t");

Serial.println (degree2); // DISPLAY ON SCREEN THE SERVO2 DEGREE VALUE FROM 0 TO 180

}
```

Code for the visual programming software by blocks compatible

1. Download and install the program.
2. To create this code you will use variables. The variables are a block to which you will assign a name. In that block you will save values.
3. The variable blocks are located in Data and Blocks. To create a variable click on Create a variable, name it and you will have blocks available for that variable.
4. The variables you will create are called DEGREE1 and DEGREE2. They will have the degree’s values from servomotors 1 and 2, which will be modified basing on the movement of the joystick.
5. Open the program and copy the following code:



```
Arduino Program
set DEGREE1 to 90
set DEGREE2 to 90
forever
    if read analog pin (A) 1 > 550 then          MOTOR 1
        set DEGREE1 to DEGREE1 + 5
        set servo pin 6 angle as DEGREE1
        if DEGREE1 > 180 then
            set DEGREE1 to 180
    if read analog pin (A) 1 < 480 then
        set DEGREE1 to DEGREE1 - 5
        set servo pin 6 angle as DEGREE1
        if DEGREE1 < 0 then
            set DEGREE1 to 0
    if read analog pin (A) 0 > 550 then          MOTOR 2
        set DEGREE2 to DEGREE2 + 5
        set servo pin 9 angle as DEGREE2
        if DEGREE2 > 140 then
            set DEGREE2 to 140
    if read analog pin (A) 0 < 480 then
        set DEGREE2 to DEGREE2 - 5
        set servo pin 9 angle as DEGREE2
        if DEGREE2 < 70 then
            set DEGREE2 to 70
    wait 0.05 secs
```

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

RESULT OF THE EXERCISE

The program will be constantly reading information from the 2 joystick outputs: VRx and VRy.

If there is an increase to a value higher than 550, the servomotors will increase 5º, and when there is a decrease to a number lower than 480, the servomotors will subtract 5º, until reaching the highest and lowest values: 180º and 0º.

The increment and decrement of the values of the joystick correspond to the forward, backward, left and right movements. If the joystick remains static, the servomotor will keep its last position.