PROJECT Nº 1: Control a servomotor

Build and code a circuit to control the position of a servomotor using a potentiometer.
DIFFICULTY LEVEL: Beginner.

DURATION OF THE EXERCISE: 30 min.

MATERIALS:

- 1 Servomotor
- 1 Potentiometer
- 1 Build&Code UNO board
- 1 Protoboard

What is a servomotor?

A servomotor is a DC motor that instead of making a continuous rotation, it is designed to turn to a certain angle according to a control signal, and to stay in that position.

To control a servomotor, pulses must be sent to it every 20 ms, which means 50Hz. The width of the pulse determines the angle of the rotation, which is known as PWM. This width is usually between 0.5 and 2.5 ms, although it can vary in different servomotors. This allows the servomotor to have an operation range between 0º (0.5 ms pulse width) and a maximum of 180º (2.5 ms pulse width).

The Build&Code servomotors are small and they function with 5V. They are controlled by a PWM digital pin of the Build&Code UNO. The pulse width will determine the rotation angle of the servomotor.
A servomotor has 3 connectors: 1 red (5V), 1 black or brown (GND) and 1 yellow or white (control).

**CONNECTIONS:**

1. The protoboard receives the electricity from the 5V pin of the Build&Code UNO and then goes back to the protoboard from the GND pin of the Build&Code UNO. All grounds of the circuit must be connected to each other so they have the same GND value. In the image, the GND is represented with a **black** cable, where all the components are connected to each other and to the GND board. The **red** cable represents the 5V, that feeds both the servomotor and the potentiometer.
2. Connect the servomotor pin that is soldered to the **yellow** cable to a PWM digital pin of the Build&Code UNO board. In the image, you will see that the servomotor to the PWM digital pin 6, with the **orange**.
3. Connect the central pin of the potentiometer to an analog pin of the Build&Code UNO board, because the potentiometer is an analog sensor. This way you will make the readings of the sensor information. In the image you will see that the potentiometer is connected to the analog pin A0 using the **green**.

**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 write a program that will be continuously reading the information of the potentiometer. According to the information sent by the potentiometer, the servomotor will turn certain angle to one side or another.

You will need to use an Arduino Library that is already installed in the software. To load it you must write “#include”. The potentiometer will make readings from 0 to 1023, depending on how much
The Build&Code servomotor can rotate from $0^\circ$ to $180^\circ$, so you will have to make a lineal relation between the 1024 possible potentiometer states and the $180^\circ$ of the servomotor. To achieve this you will use the “map” instruction and the result of this mapping is the information you will use to move the servomotor.

1. Download and install the Arduino IDE program. It is available for Windows, Mac OS and Linux.
2. Open the Arduino program and copy the following program in it:

```cpp
#include <Servo.h> //Load the servo control Library

Servo myservo; //Assign the myservo variable to work with the servomotor

int sensor = A0; //Potentiometer connected to analog pin A0
int value; //create variable named value to store the sensor values
int Degrees; //create variable named Degrees to store the value in grads of the potentiometer (from 0 to 180)

void setup() {
  myservo.attach(6); //Assign the servo motor to Pin 6
}

void loop() {
  value = analogRead(sensor); //Instruction to read and store the analog value (from 0 to 1023)
  Degrees = map(value, 0, 1023, 0, 180); // Scales the sensor readings value with the servo motor rotation degrees
  myservo.write(Degrees); //Introduce the Degrees value in the servo motor to rotate it
}
```

Code for the visual programming software by blocks compatible
1. Download and install the program.
   1.1 Open the software.
   1.2 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. Use the following image as a guide:

RESULT OF THE EXERCISE

The LED board integrated in the Build&Code UNO board will turn on and off according to the
defined time intervals. This way, you will make sure that the Build&Code UNO board is working
correctly.