Arduino based robotic object pick arm

Objectives:

-Here in this project, **Arduino Uno is programmed to control servo motors** which are serving as joints of Robotic arm for picking an object.

-This setup also looks as a **Robotic Crane** or we can convert it into a Crane by doing some easy tweaks.

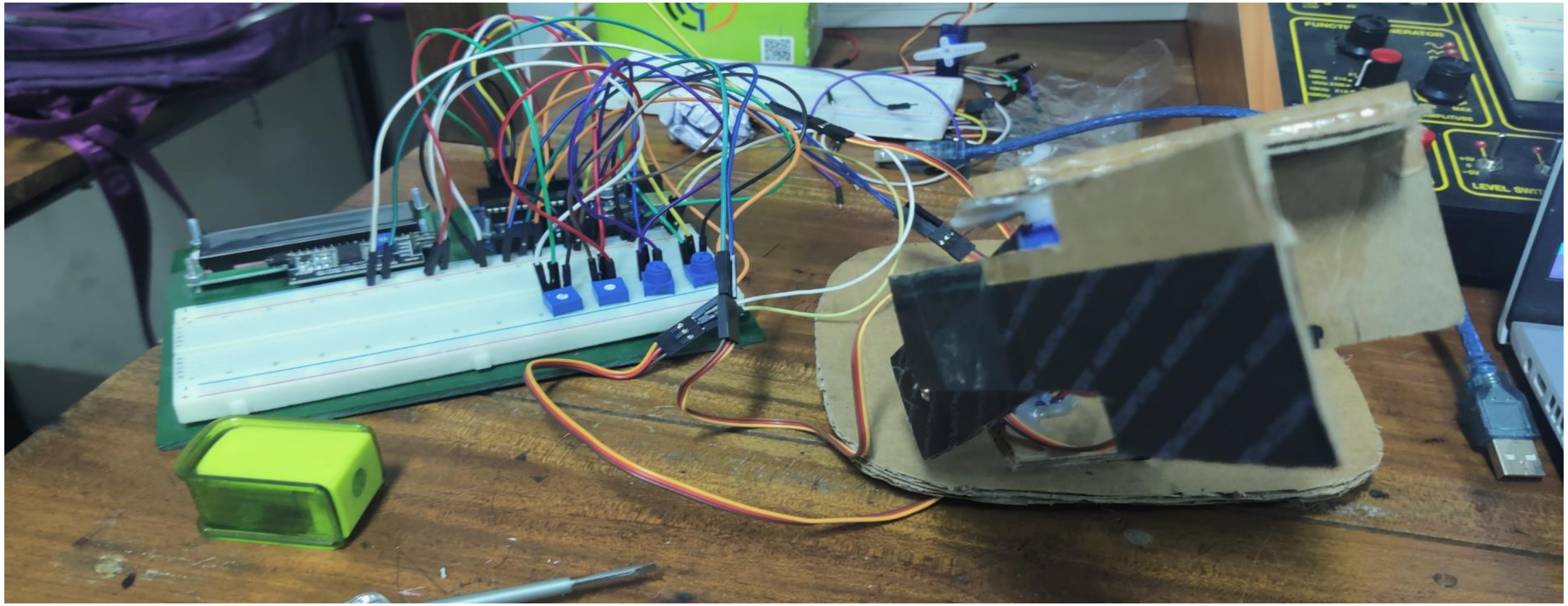
-This project will be helpful for beginners who want to learn to develop a Simple Robot in low cost or just want to learn working with Arduino and servo motors.

-This **Arduino Robotic Arm** is controlled by **four Potentiometer** attached to it, each potentiometer is used to control each servo. We can move these servos by rotating the pots to pick some object.(Low torque servos)

Servo Motor

Servo Motors are mainly used when there is a need for shaft movement or position

- Here four SG90 servos were used



Applied Circuit Design:

As per as the practical circuit construction I have used low torque servo motors and to control individual servo we used individual potentiometer.

-A servo motor basically controls the rotation of a DC motor by controlling a circuit that synthesizes its angle.

-The angle of a servo motor is between 0 degree and 180 degrees .To move the servo, I used Write() function with the required angle of rotation as argument.

-Then pulse width sent to the motors then changes according to angle change and it demarks the extent of rotation.

-By varying potentiometer I moved the cardboard attached with servo motors in between angle and pick my object.

CIRCUIT OPERATION:

- Used four potentiometer with 10 k resistance.
- Every pot will work on 5 volt which divided into 1023 division.
- Each division will carry 5-6 mV.
- For 5-6 mV it rotates 1degree.
- To control this robotic arm four servo were applied for four parts to move around the robotic arm as my wish.
- It can rotate 360 degree.

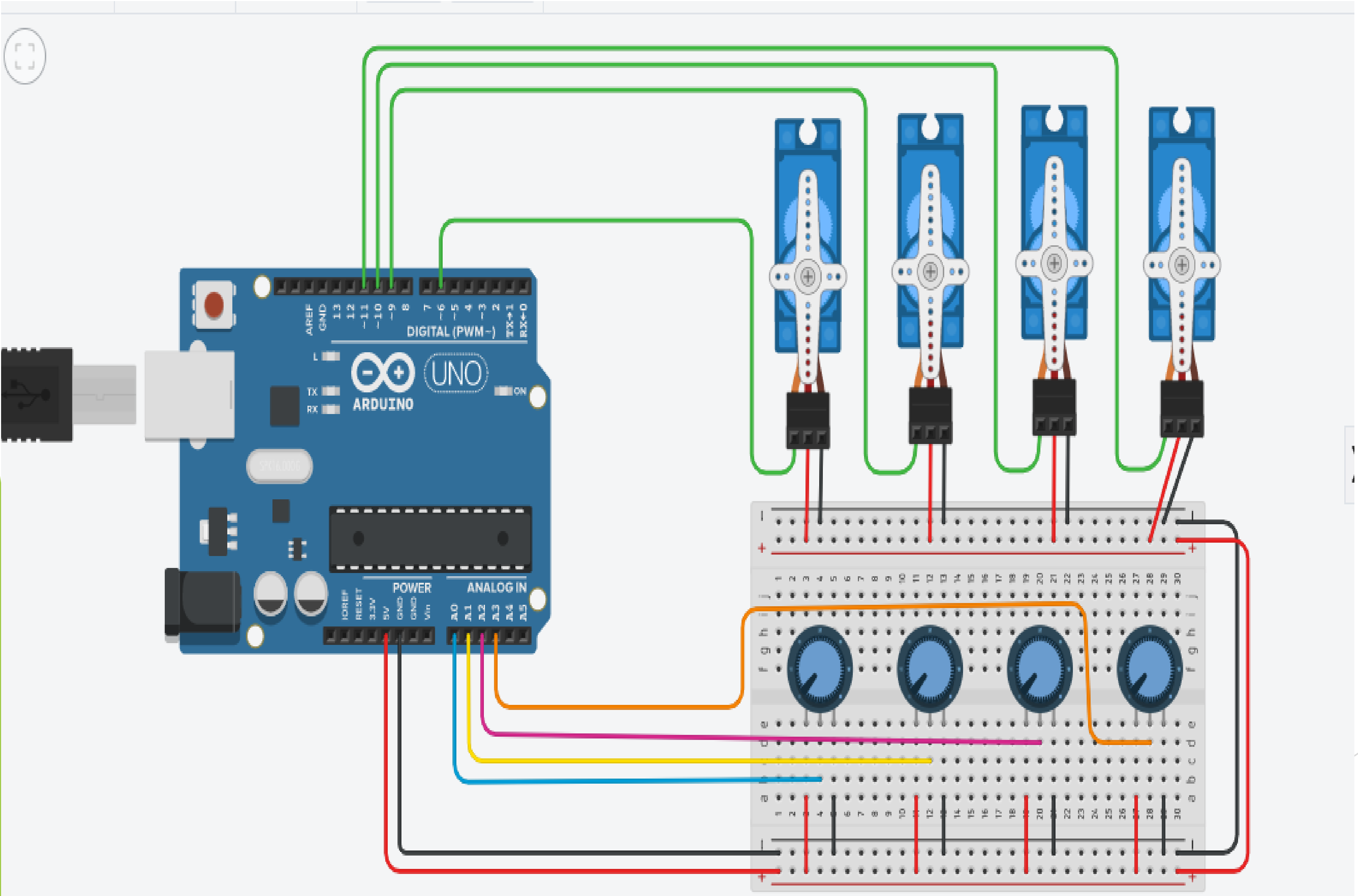
Functions of Servo Motor:

- The first servo is in the base and it controls the robotic arm to rotate 180 degrees.
- Second servo was used as the elbow of the robotic arm and it helps to uplift the arm left to right and right.
- Third servo is in wrist position and it helps to pick up any object.
- The fourth servo is in the gripper position and it helps to grip any object as human fingers.

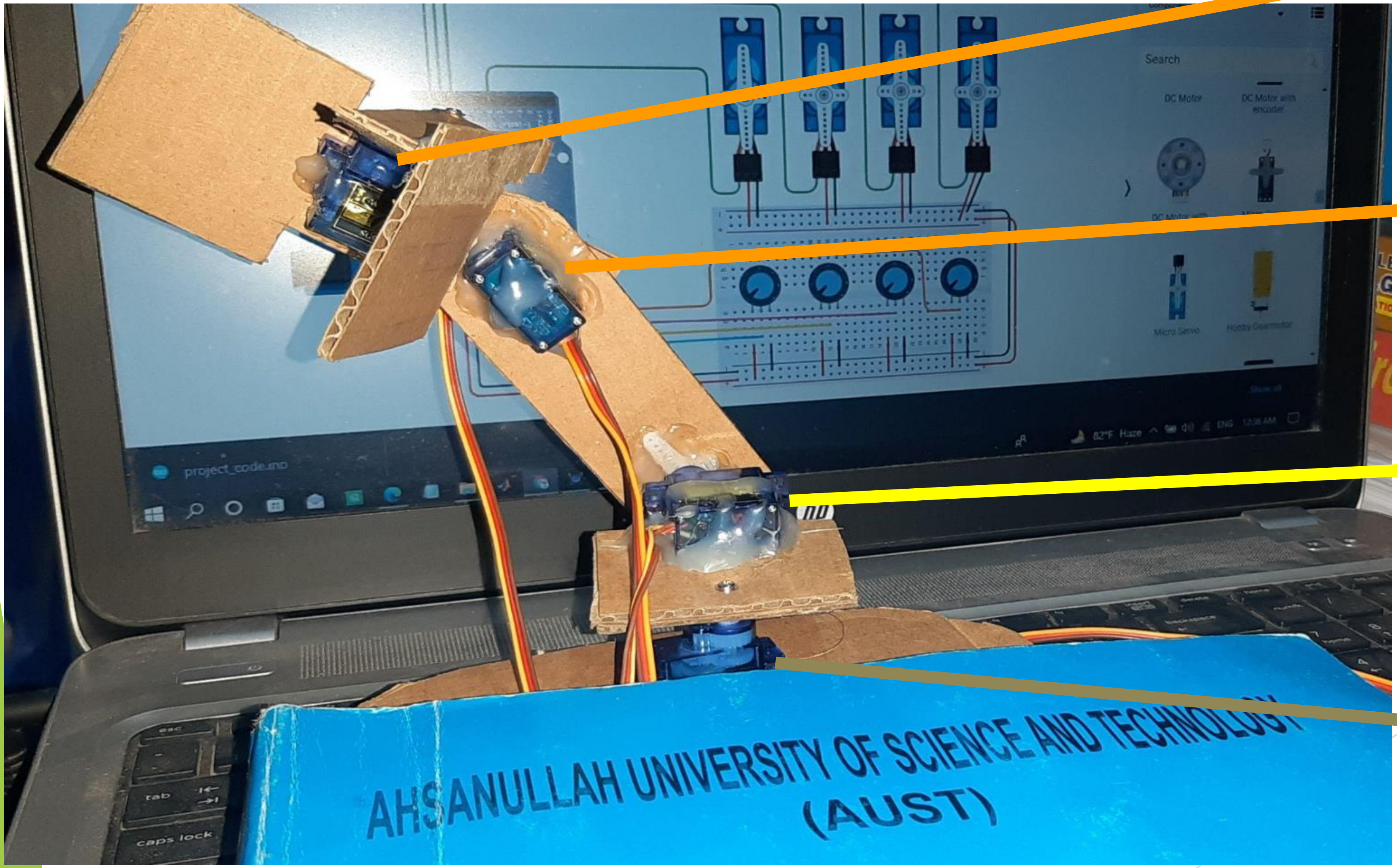
CIRCUIT SETUP

- First I connected the servo motor here first servo is connected to digital pin 6, and the second servo is connected to digital pin 9; the third servo is connected in digital pin 10; the fourth servo is connected in digital pin 11. and other two pin of each servo middle pin is connected in 5 volts and other pin is connected in ground pin. This digital pin can act as an analog pin.
- Then I connected the potentiometer to control the servo, here first potentiometer is connected to analog pin A0; the Second potentiometer is connected in analog pin A1; Third potentiometer is connected in analog pin A2; Fourth potentiometer is connect in analog pin A3;and other two pin of each potentiometer middle pin is connected in 5 volts and other pin is connected in ground pin.

Circuit Diagram



Practical Position of Servo



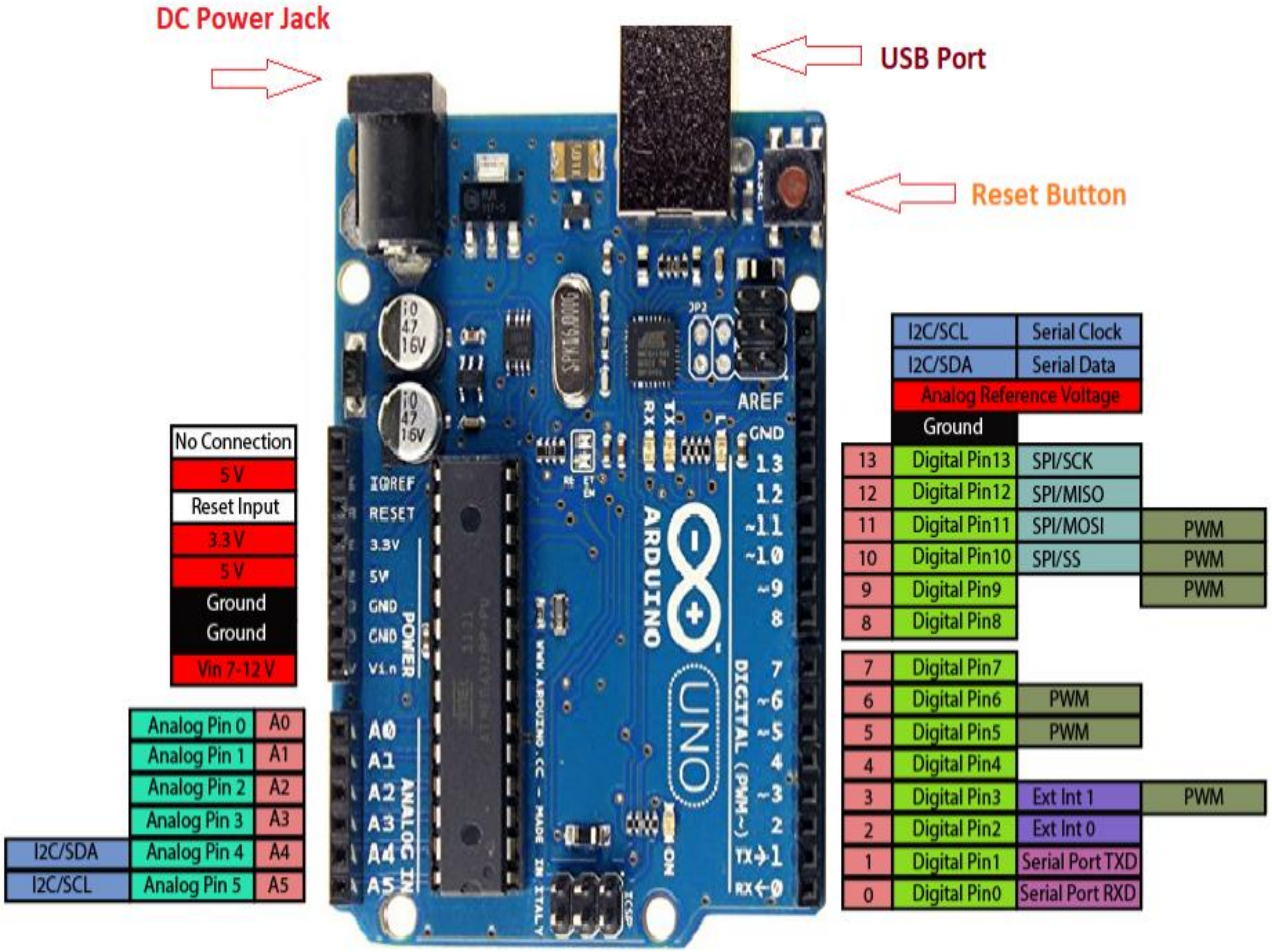
4th Servo

3rd Servo

2nd Servo

1st Servo

Arduino UNO Board



DC Power Jack

USB Port

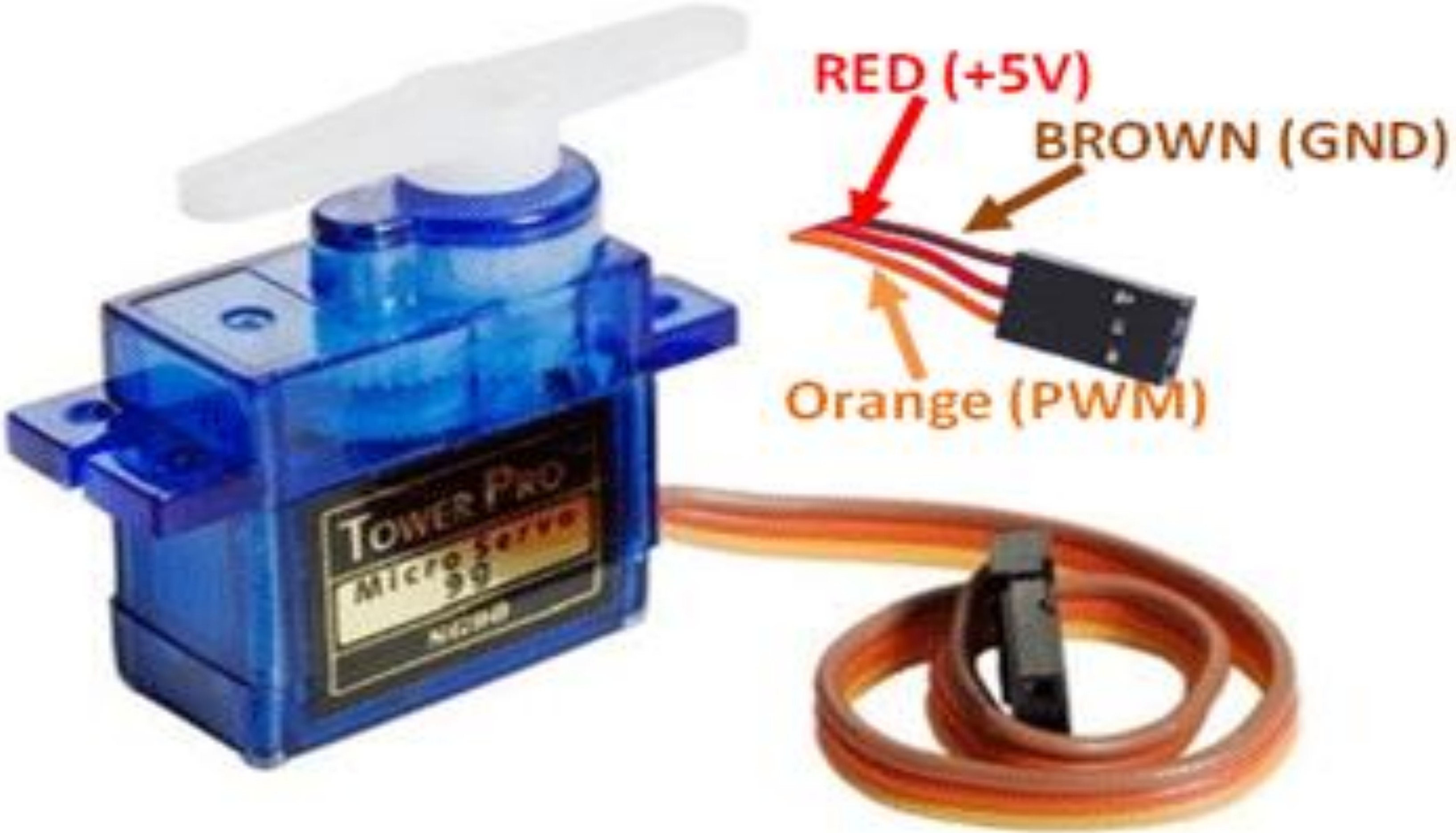
Reset Button

No Connection
5 V
Reset Input
3.3 V
5 V
Ground
Ground
Vin 7-12 V

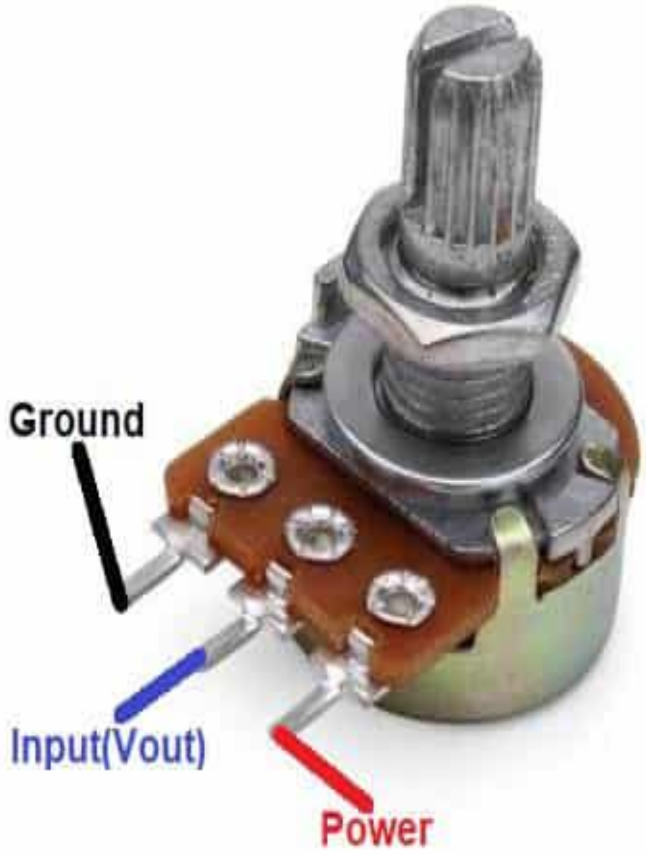
	Analog Pin 0	A0
	Analog Pin 1	A1
	Analog Pin 2	A2
	Analog Pin 3	A3
I2C/SDA	Analog Pin 4	A4
I2C/SCL	Analog Pin 5	A5

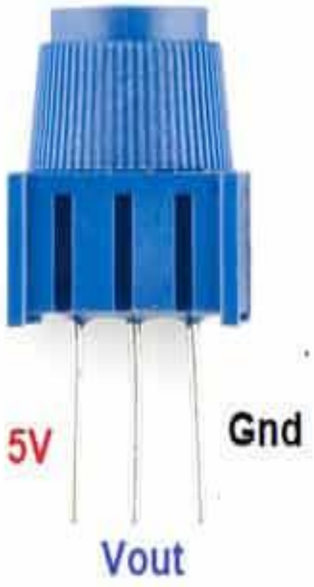
	I2C/SCL	Serial Clock	
	I2C/SDA	Serial Data	
	Analog Reference Voltage		
	Ground		
13	Digital Pin13	SPI/SCK	
12	Digital Pin12	SPI/MISO	
11	Digital Pin11	SPI/MOSI	PWM
10	Digital Pin10	SPI/SS	PWM
9	Digital Pin9		PWM
8	Digital Pin8		
7	Digital Pin7		
6	Digital Pin6	PWM	
5	Digital Pin5	PWM	
4	Digital Pin4		
3	Digital Pin3	Ext Int 1	PWM
2	Digital Pin2	Ext Int 0	
1	Digital Pin1	Serial Port TXD	
0	Digital Pin0	Serial Port RXD	

Servo Pin Connection

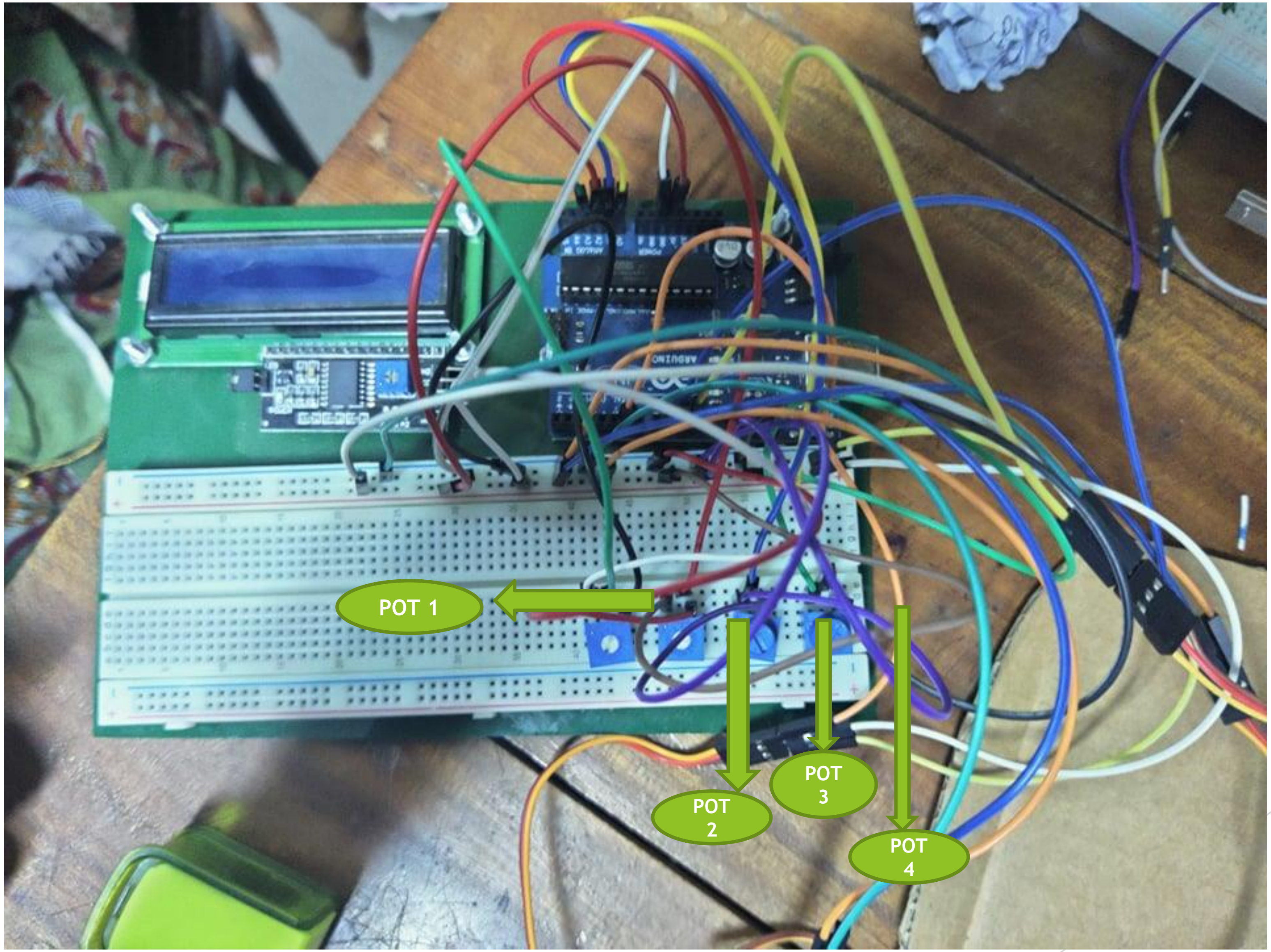


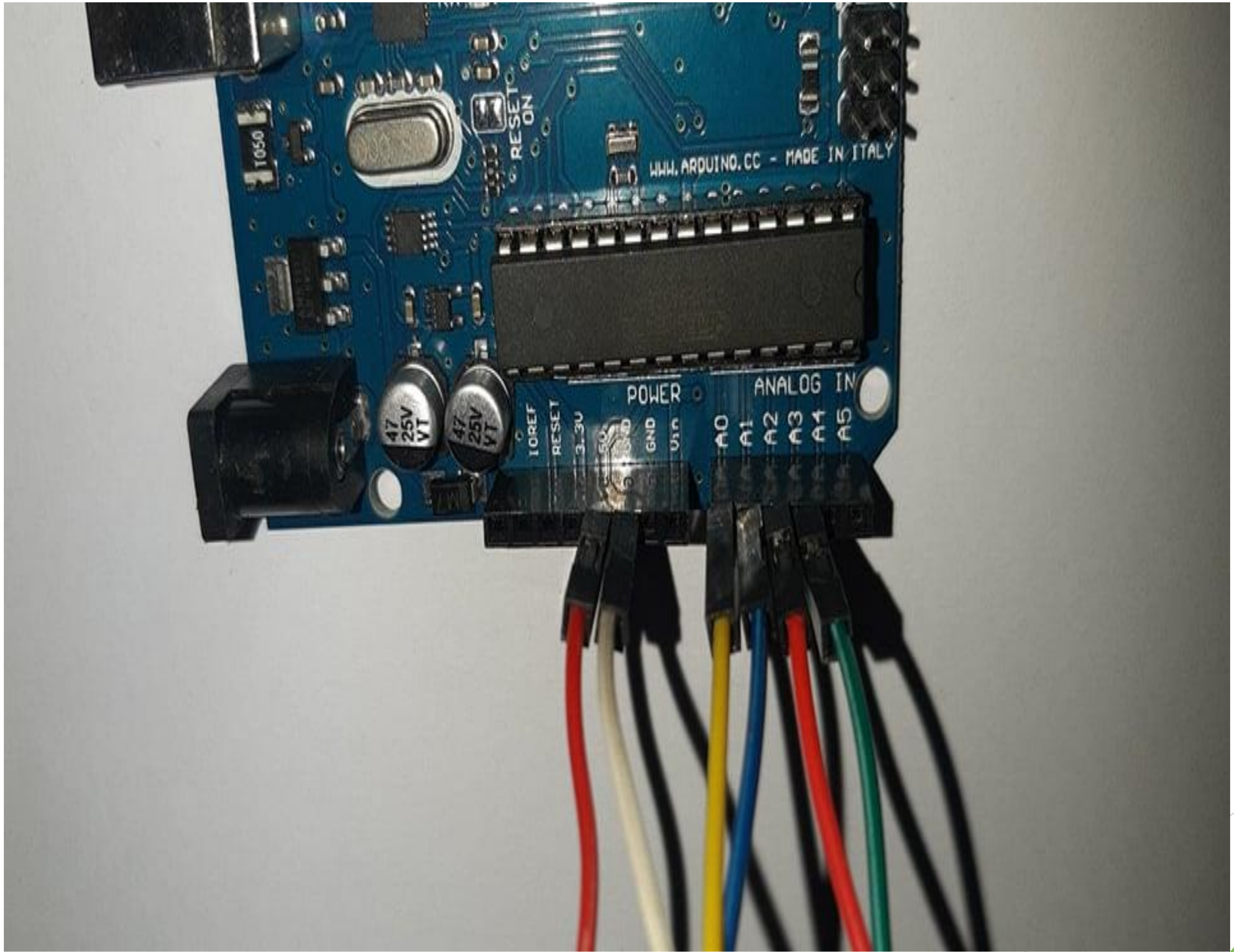
Potentiometer

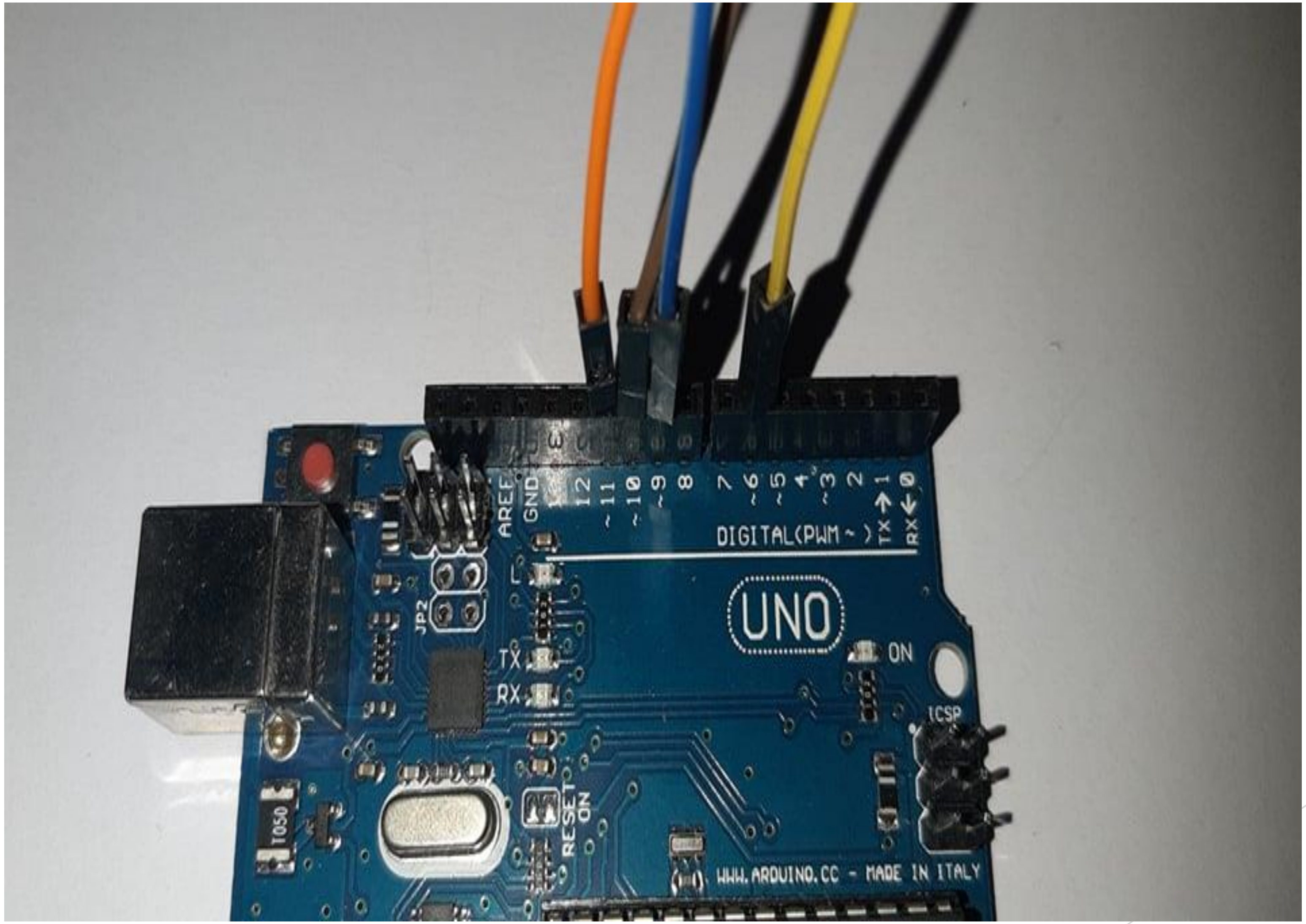




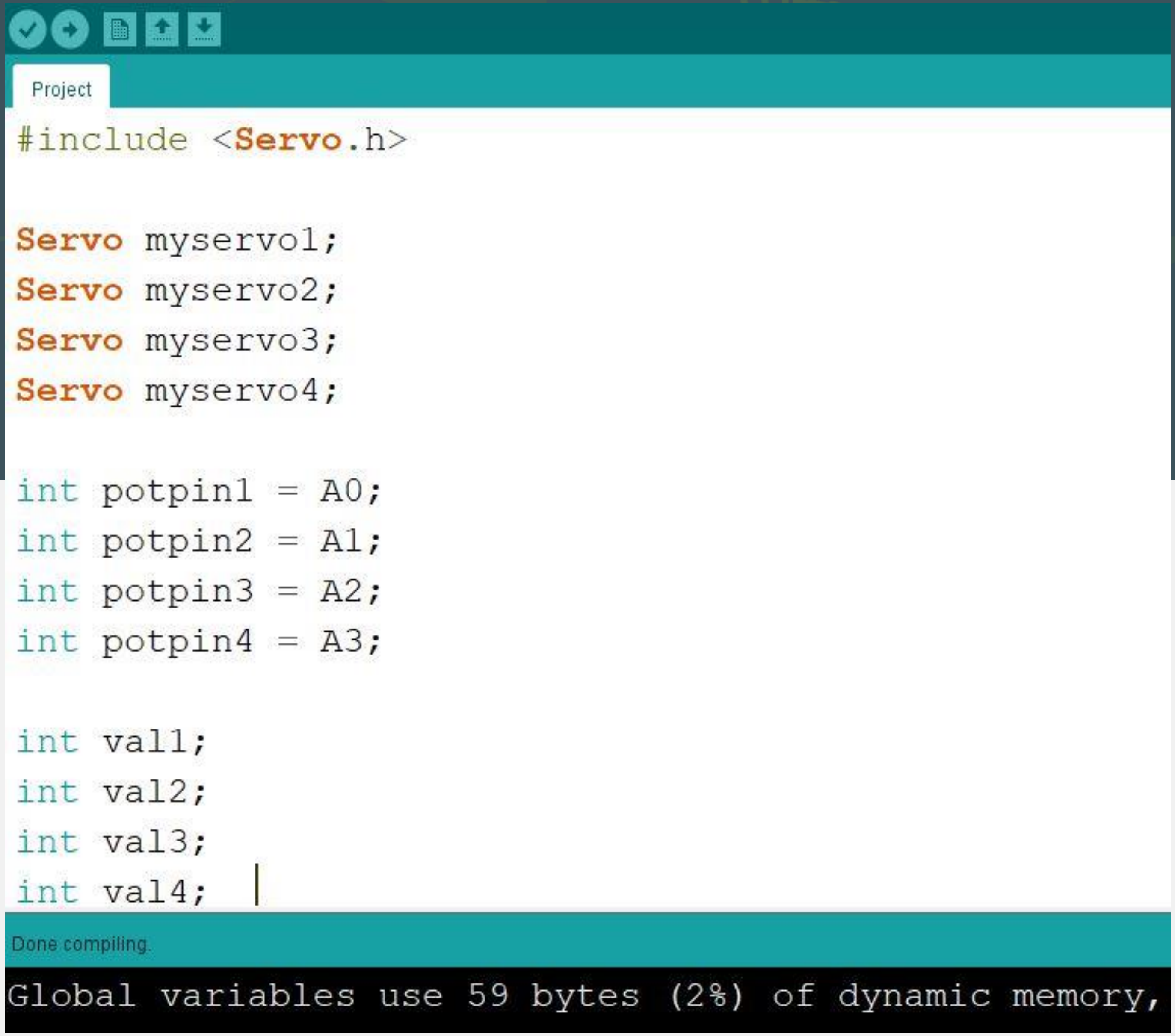
Experimental Breadboard Circuit




RESET
ON
WWW.ARDUINO.CC - MADE IN ITALY
POWER
IOREF
RESET
3.3V
5V
GND
GND
Vin
ANALOG IN
A0
A1
A2
A3
A4
A5
47
25V
VT


AREF
GND
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX 1
RX 0
DIGITAL(PWM ~)
UNO
ON
TX
RX
L
JP2
ICSP
RESET
ON
WWW.ARDUINO.CC - MADE IN ITALY
T050

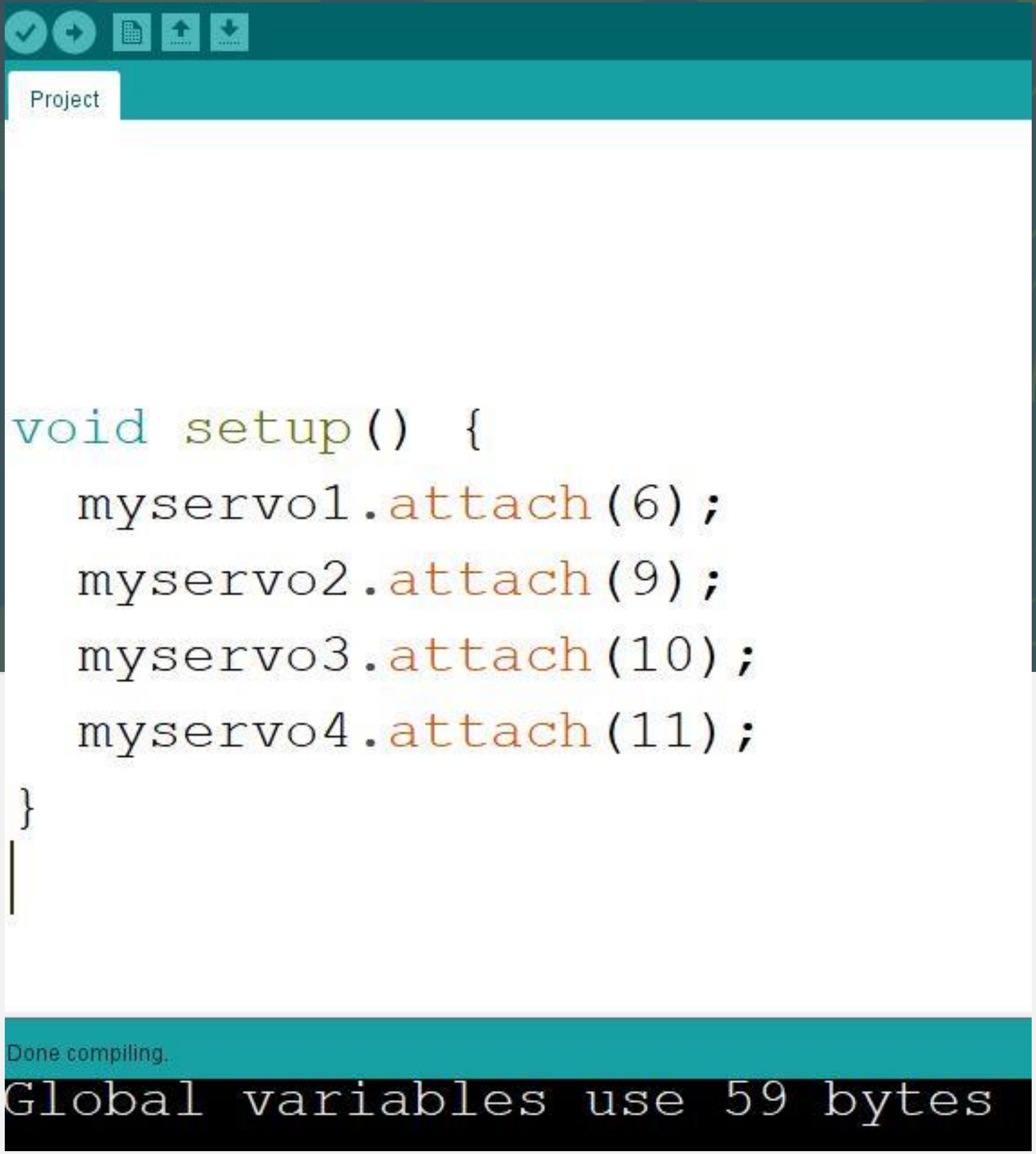
Arduino Code



```
#include <Servo.h>

Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;

int potpin1 = A0;
int potpin2 = A1;
int potpin3 = A2;
int potpin4 = A3;

int val1;
int val2;
int val3;
int val4;
```

Done compiling.

```
Global variables use 59 bytes (2%) of dynamic memory,
```

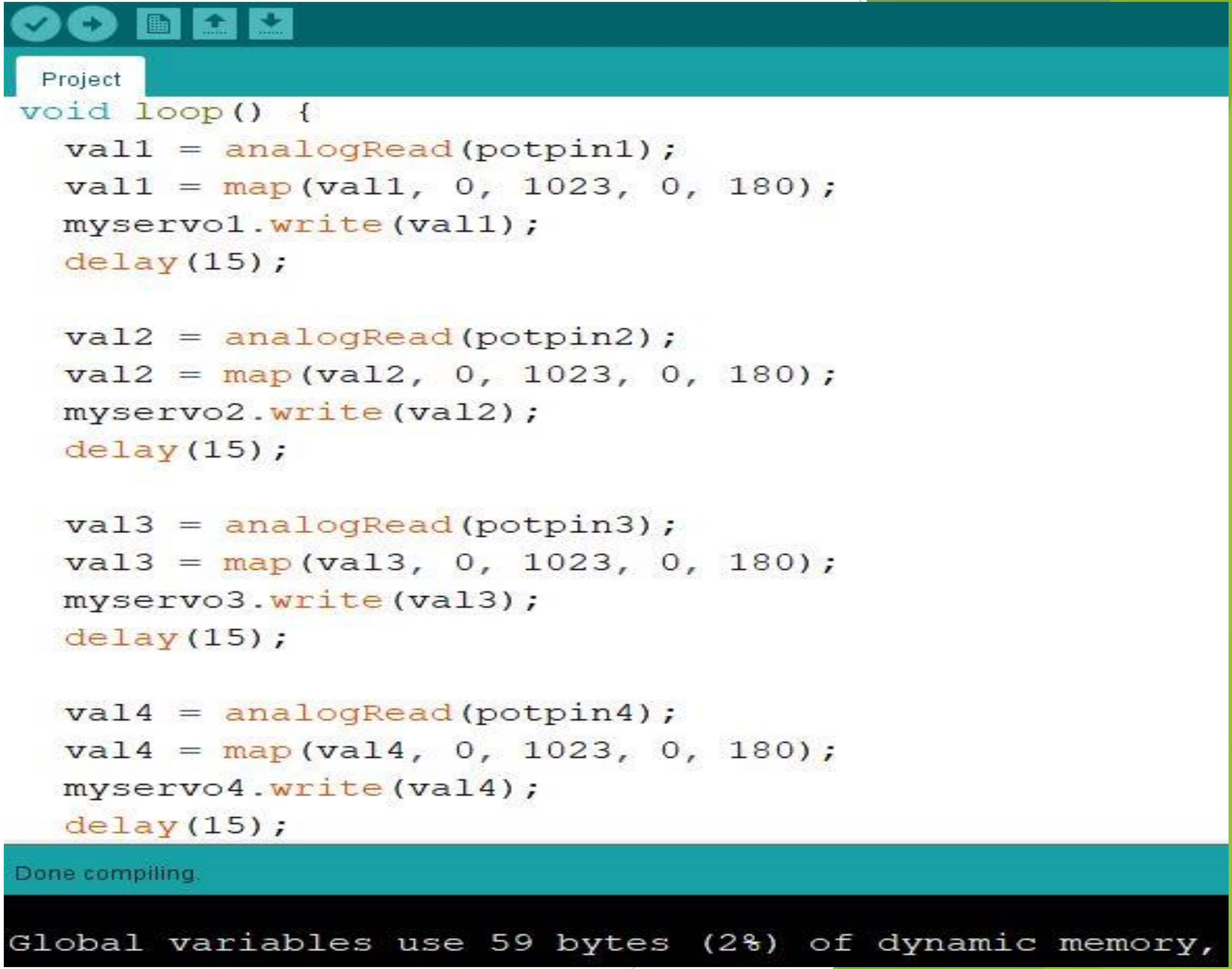


```
void setup() {
  myservo1.attach(6);
  myservo2.attach(9);
  myservo3.attach(10);
  myservo4.attach(11);
}
```

```
Done compiling.
Global variables use 59 bytes
```

Setup part of the code

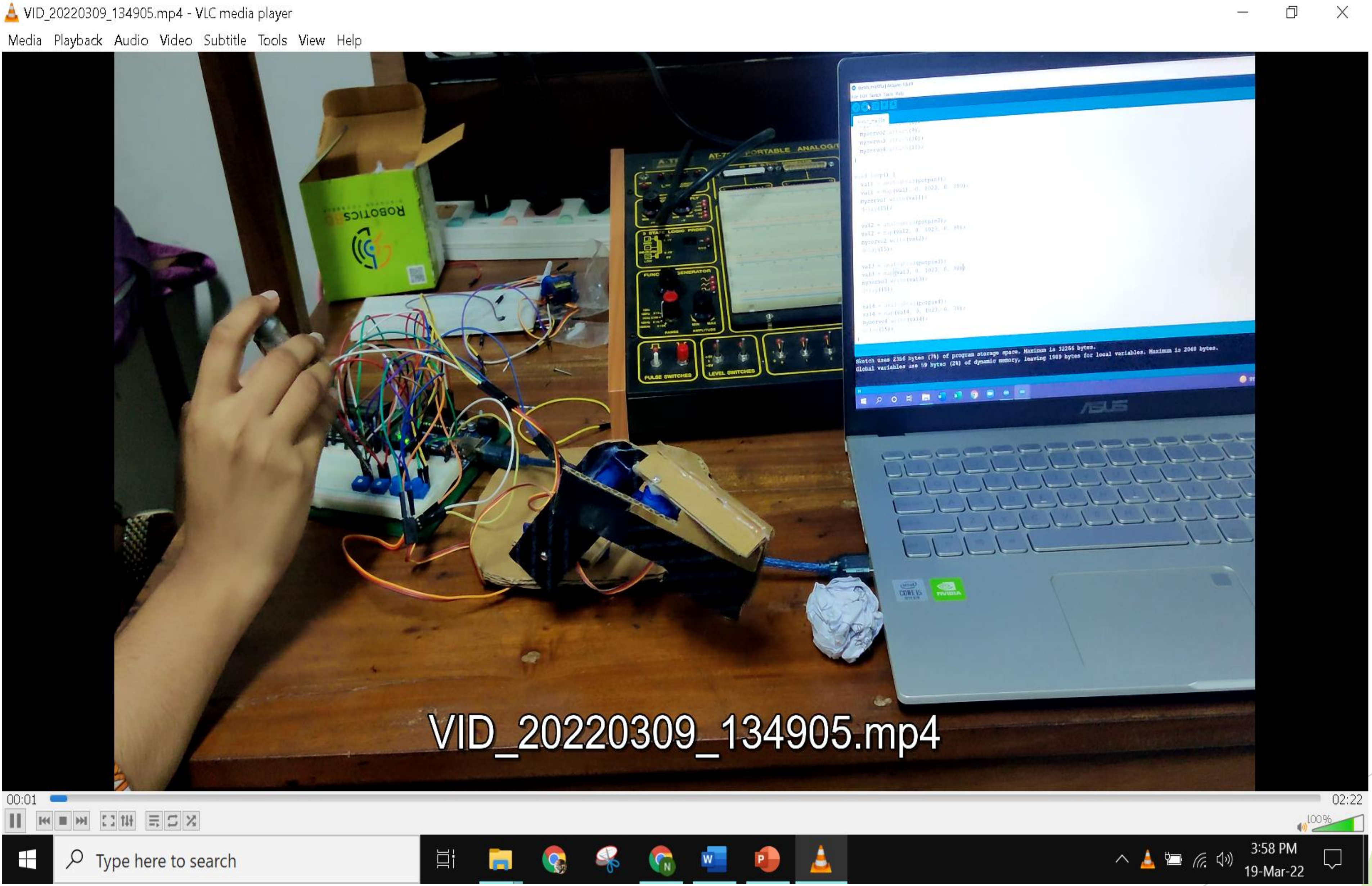
Loop part of the code



```
void loop() {
  val1 = analogRead(potpin1);
  val1 = map(val1, 0, 1023, 0, 180);
  myservo1.write(val1);
  delay(15);

  val2 = analogRead(potpin2);
  val2 = map(val2, 0, 1023, 0, 180);
  myservo2.write(val2);
  delay(15);

  val3 = analogRead(potpin3);
  val3 = map(val3, 0, 1023, 0, 180);
  myservo3.write(val3);
  delay(15);

  val4 = analogRead(potpin4);
  val4 = map(val4, 0, 1023, 0, 180);
  myservo4.write(val4);
  delay(15);
```

Done compiling.

```
Global variables use 59 bytes (2%) of dynamic memory,
```



```
Sketch uses 2356 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 59 bytes (2%) of dynamic memory, leaving 1989 bytes for local variables. Maximum is 2048 bytes.
```