

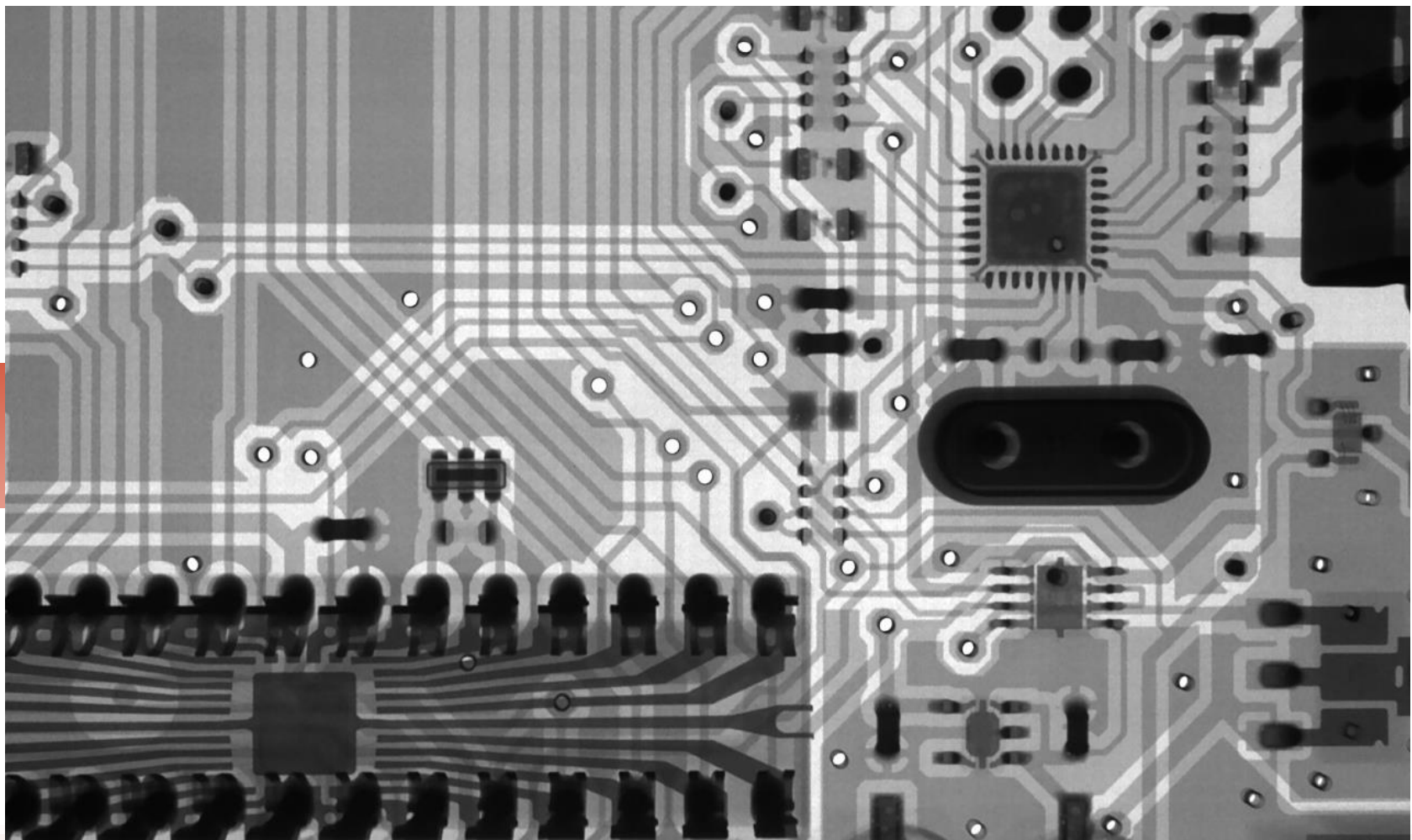


Arduino SYS-STEM for Schools

Training Methodology

MODULE 7

Servo & Continuous Servo Motors



SERVO and CONTINUOUS SERVO



EXPECTED LEARNING OUTCOMES

Knowledge

Upon completion of this unit, the student will be able to:

- Understand how servo and continuous servo works
- Know the purpose of using servo and continuous servo module and in which situations they can be applied

Competences and Skills

Upon completion of this unit, the student will be able to:

- Use a servo and continuous servo module on the Arduino platform

Description of the sensor module

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback.

It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor, although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

DC Motor

- Motion is continuous
- Speed controlled by applied voltage

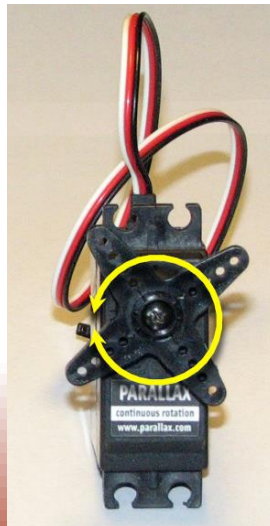
Servo

- Capable of holding a position
- Speed controlled by delay between position updates
- Hybrid of motor, gears and controller.

Two types of servos

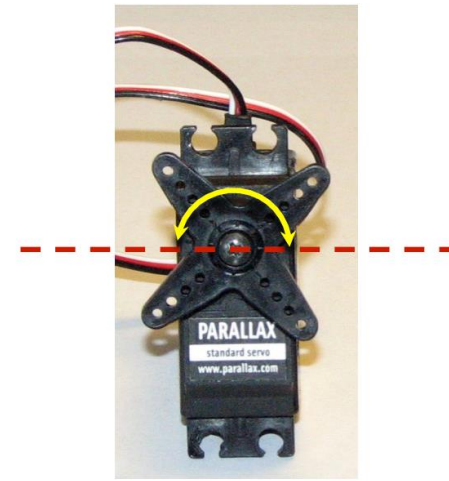
continuous rotation

- can rotate all the way around in specific direction
- pulse tells servo which way to spin & how fast to spin

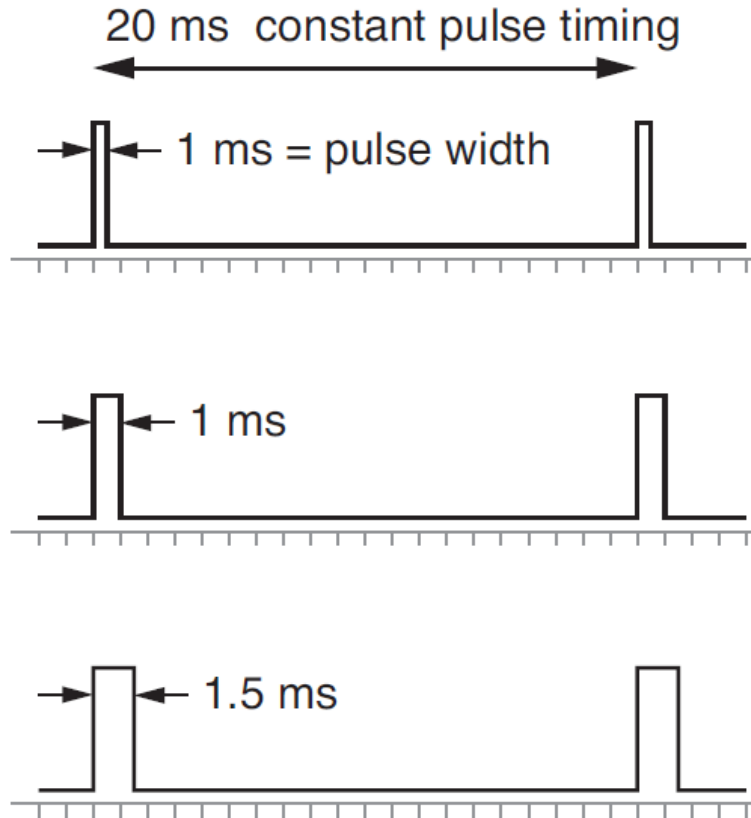


standard

- can only rotate 180 degrees
- pulse tells servo which position to hold



Control signal is a pulse train



Pulse frequency is fixed
Typical: 20 ms

Pulse width determines position
Typical: 1ms to 2 ms

Description of the sensor module

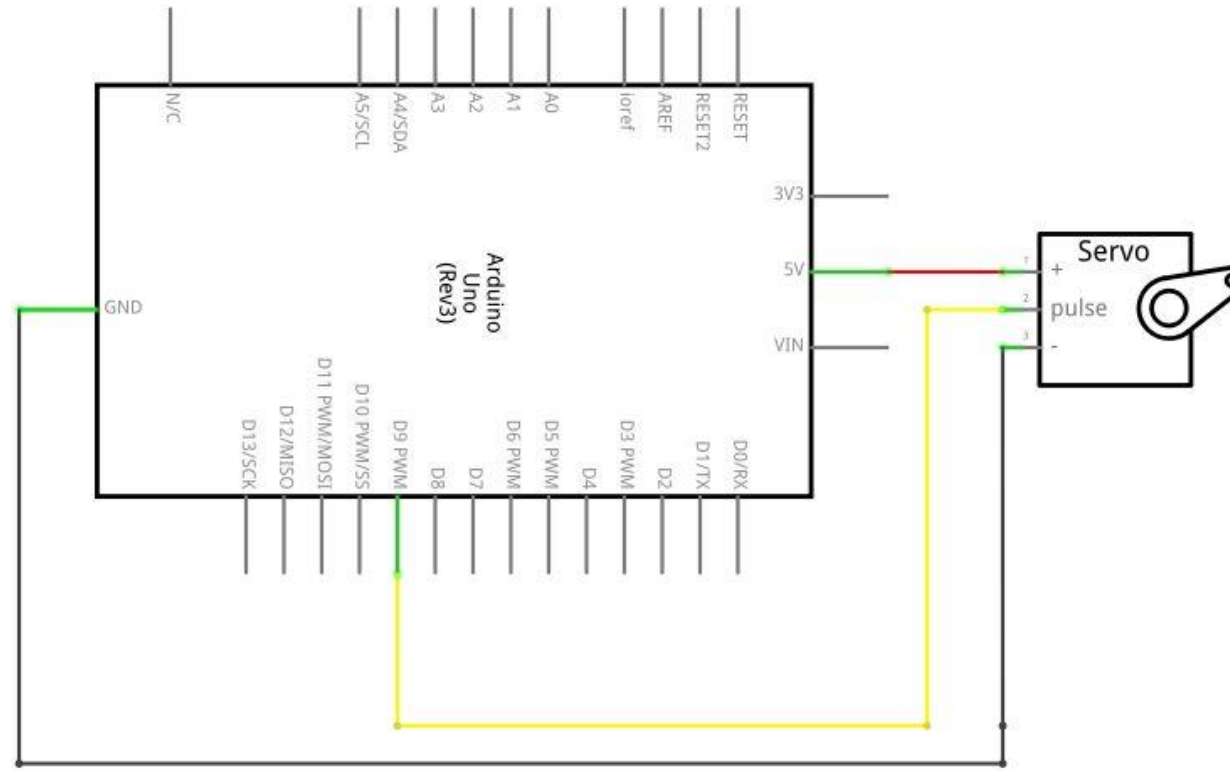
- Pin out:
 - 1. **GND**: ground
 - 2. **VCC**: 5V
 - 3. **PWM** → D9 pin Arduino



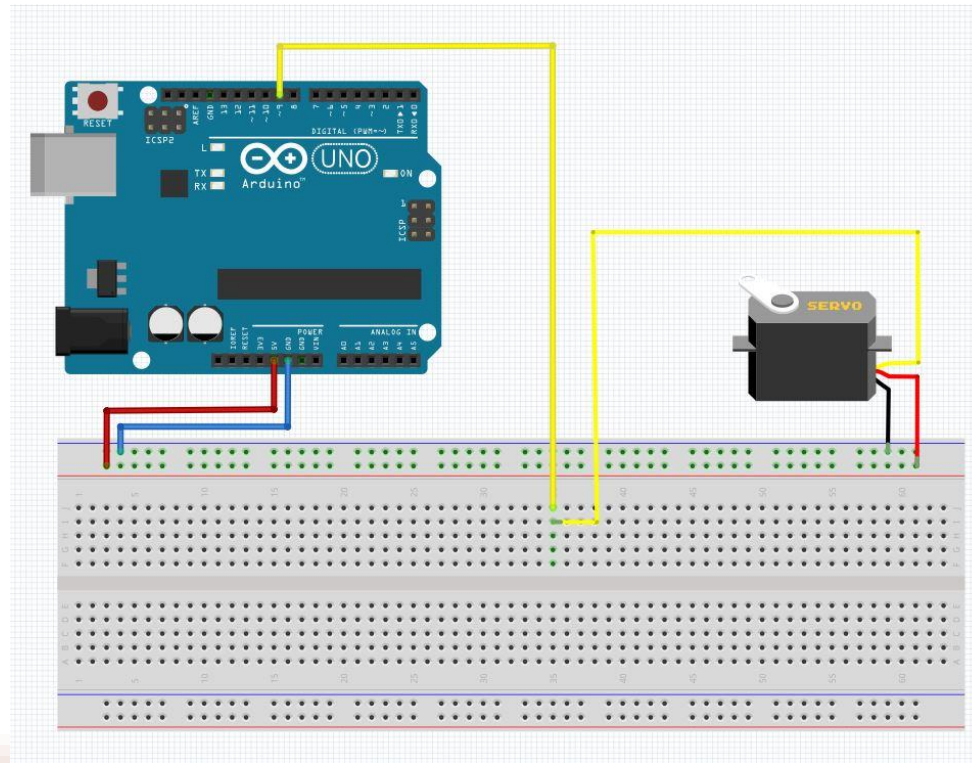
Description of examples of using the sensor module

- The example will show how to control a servo using an arduino code. In the code, we will define variables that will move the servo 180 degrees clockwise, and after a short wait 180 degrees counterclockwise.
- For continuous servo we will use a variable that describes the direction and speed of motion. A signal of 0 will start rotating the continuous servo counterclockwise at maximum speed. The 180 signal will start rotating the servo clockwise at maximum speed.

Example - Wiring diagram (SERVO and CONTINUOUS SERVO)



Example - Wiring diagram on a breadboard (SERVO and CONTINUOUS SERVO)



Example - Program code (SERVO)

Servo

```
#include <Servo.h>

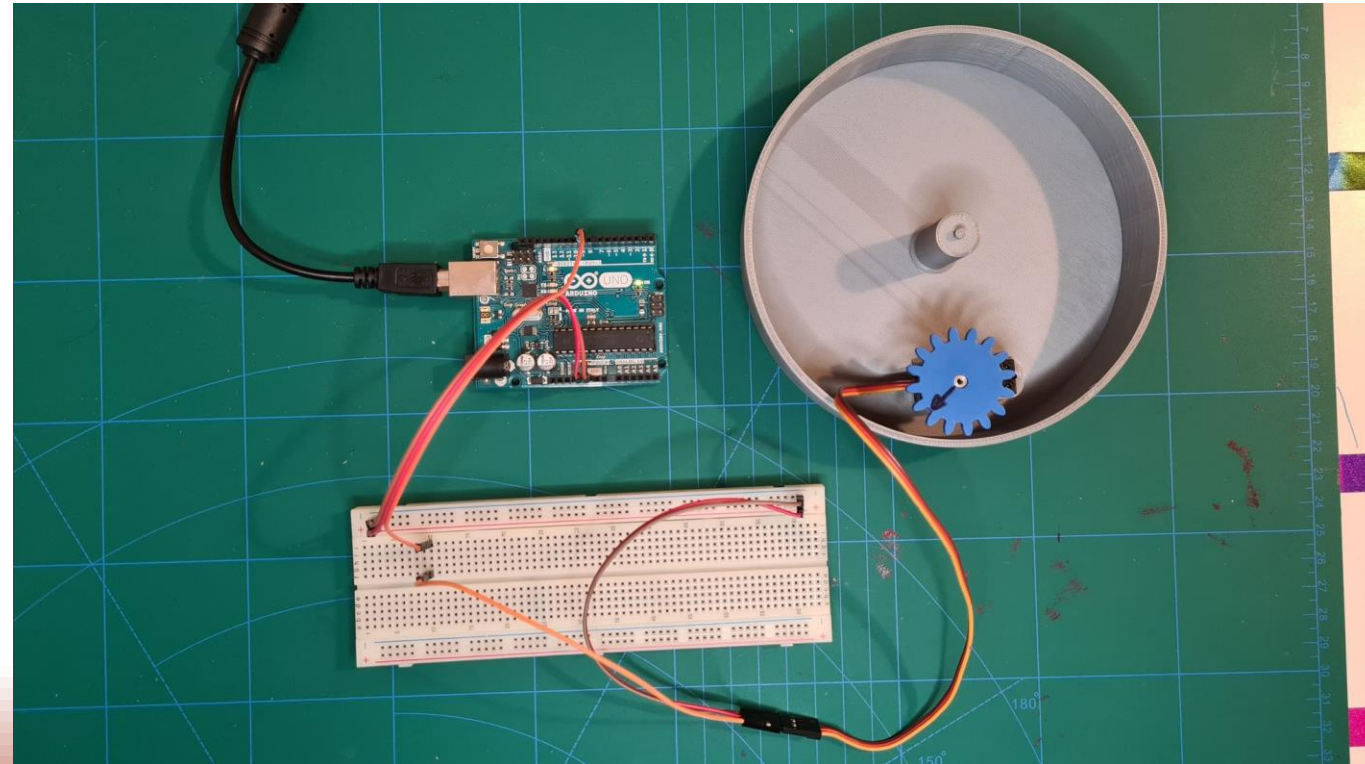
Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

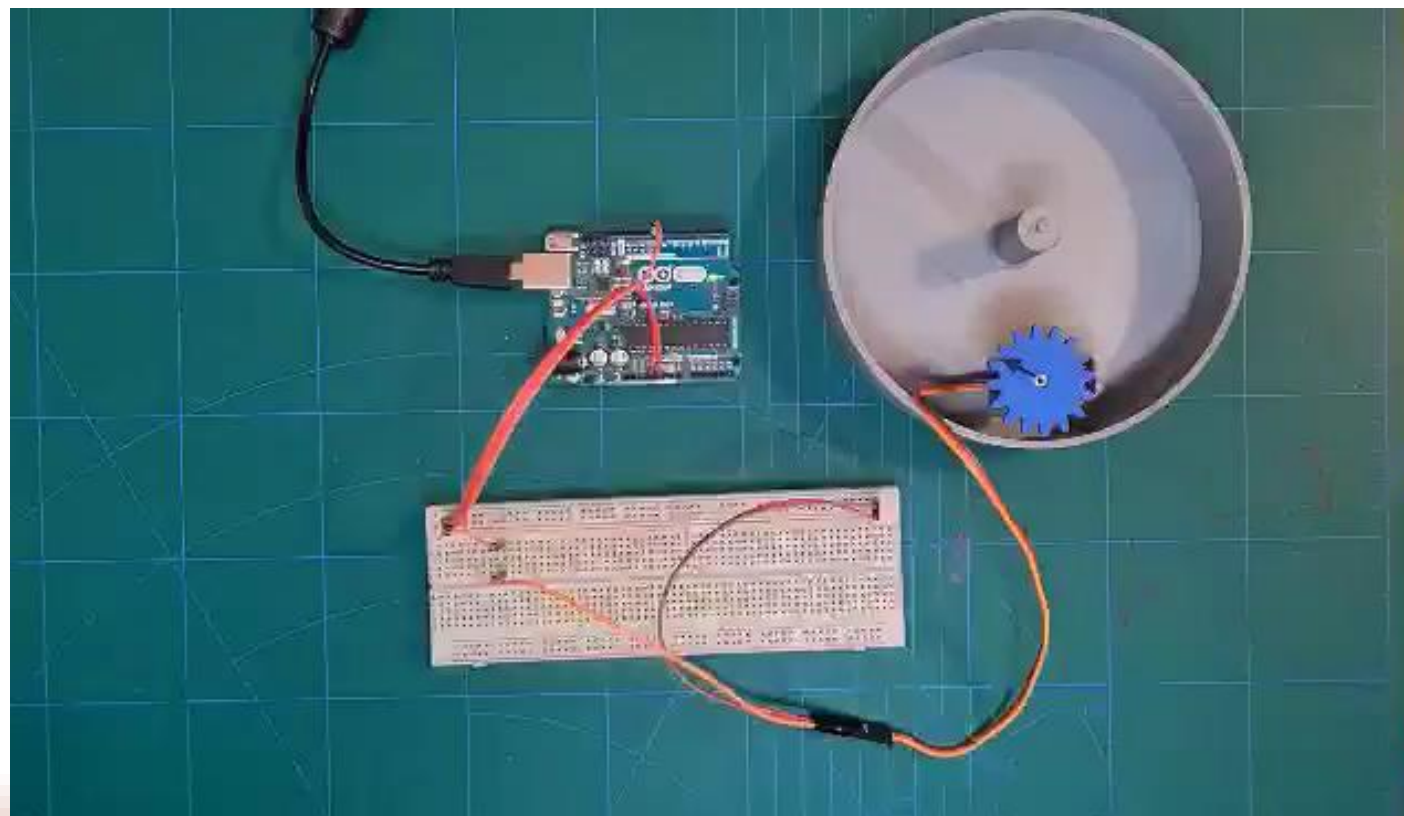
void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 2) { // goes from 0 degrees to 180 degrees
    // in steps of 2 degree
    myservo.write(pos);                // tell servo to go to position in variable 'pos'
    delay(20);                          // waits 20ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 2) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);                // tell servo to go to position in variable 'pos'
    delay(20);                          // waits 15ms for the servo to reach the position
  }
}
```

Example - Display of the connection (SERVO)



Example - Demonstration of work (SERVO)





Example - Program code (CONTINUOUS SERVO)

SERVO_CON

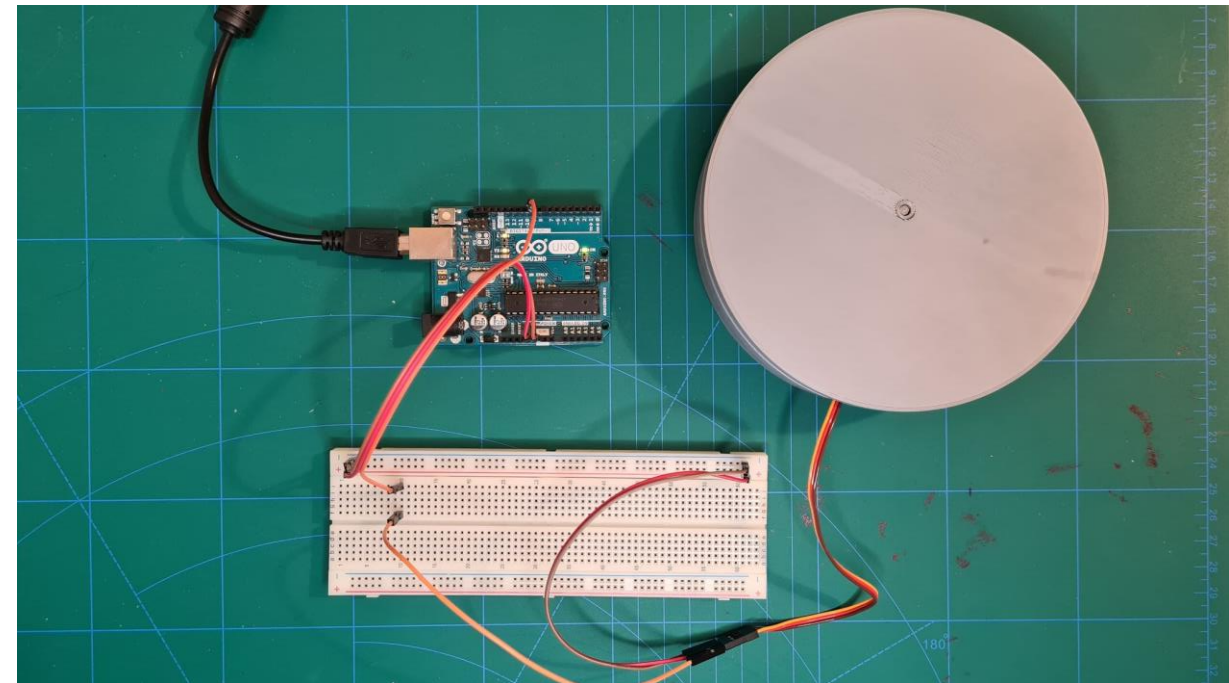
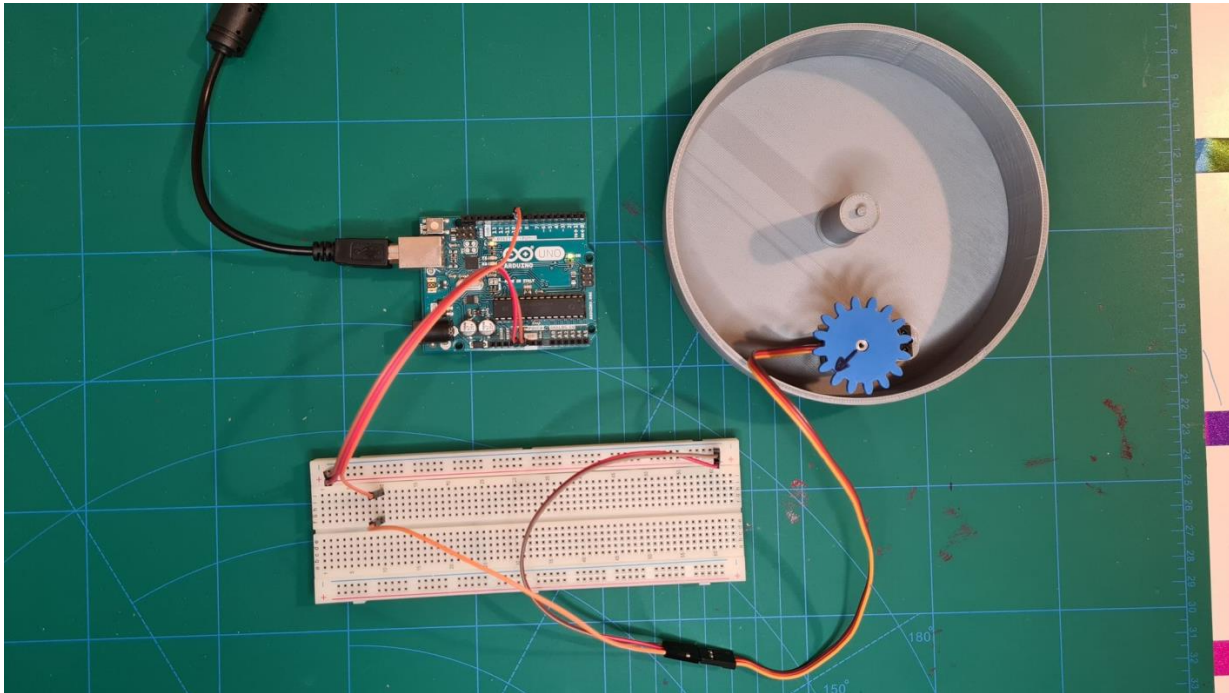
```
#include "Servo.h"
int servoPin = 9; // connect servo to pin 10
Servo myServo;

void setup(void) {
  myServo.attach(servoPin);
  Serial.begin(9600); // begin serial monitor
}

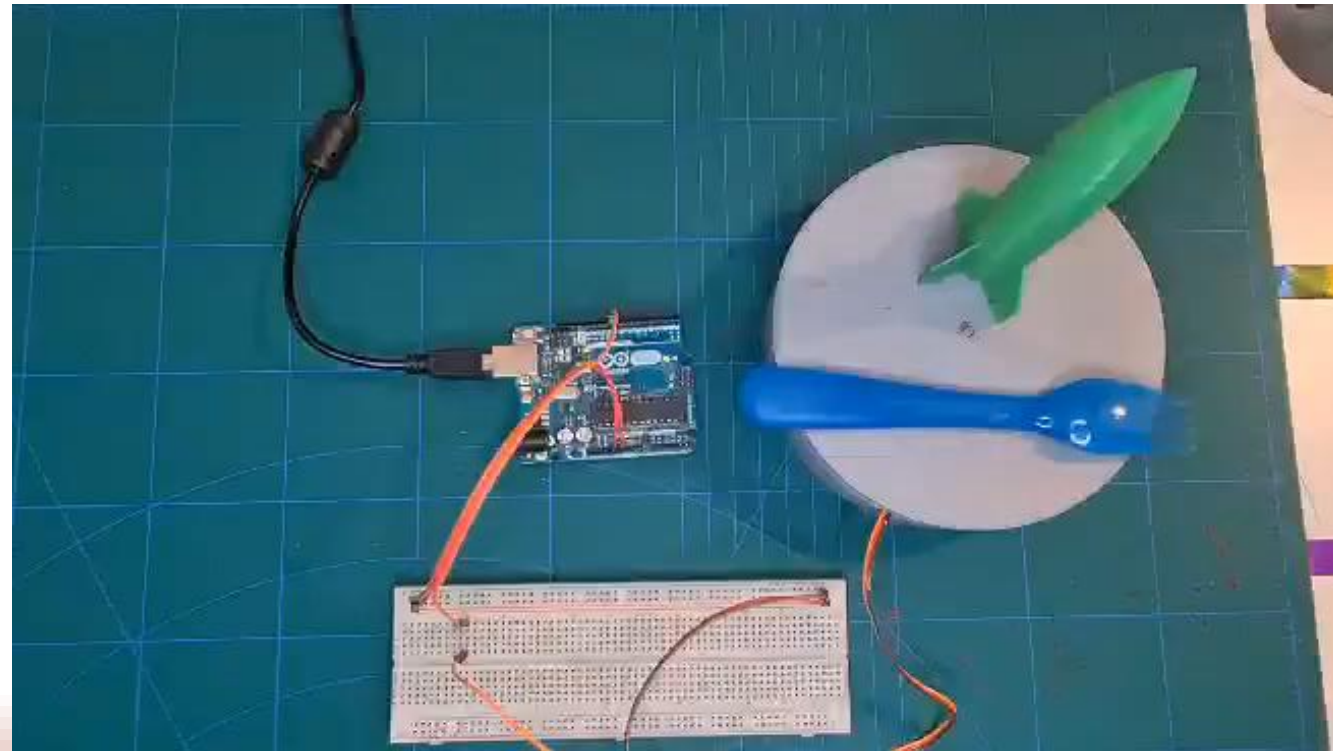
void loop(void) {
  myServo.write(0);
  Serial.println("going to the left!");
  delay(5000);

  myServo.write(180);
  Serial.println("going to the right!");
  delay(5000);
}
```

Example - Display of the connection (CONTINUOUS SERVO)



Example - Demonstration of work (CONTINUOUS SERVO)





CONGRATULATIONS

You have completed SYS-STEM Module 7 Servo and continuous servo