# Project 2 & 3

# motion control using Servo motor and Stepper motor

## Objective

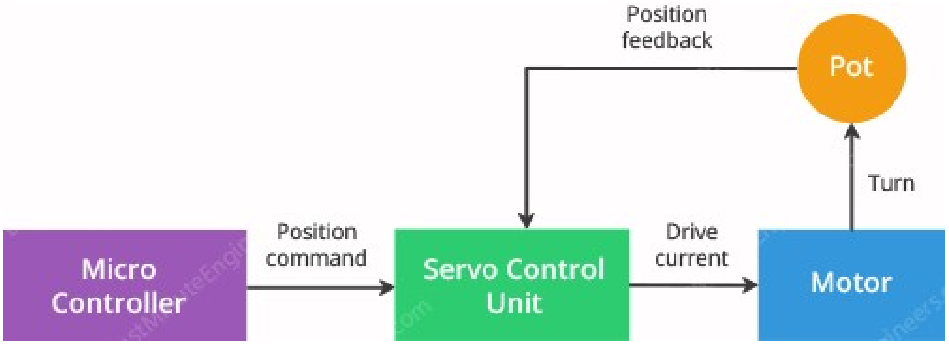
Control of Servo motor rotation using Arduino UNO.

## Required Equipment's

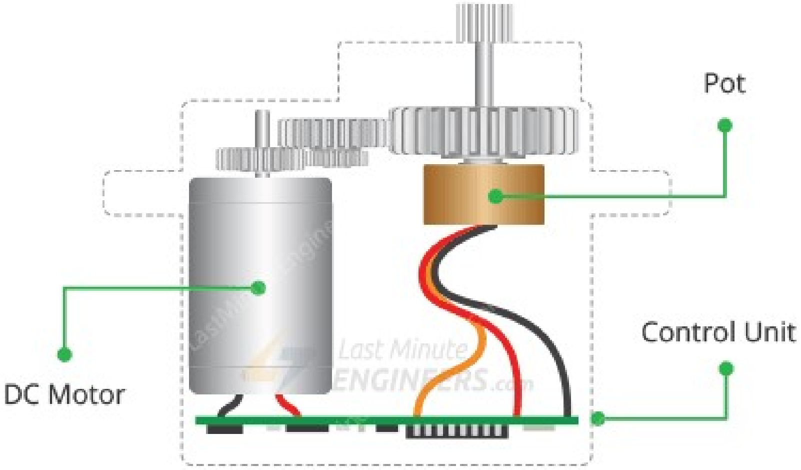
Arduino Uno, Servo Motor, 3 jumper Cables.

## What is a Servo Motor?

Servo is a general term for a closed loop control system. A closed loop system uses the feedback signal to adjust the speed and direction of the motor to achieve the desired result.

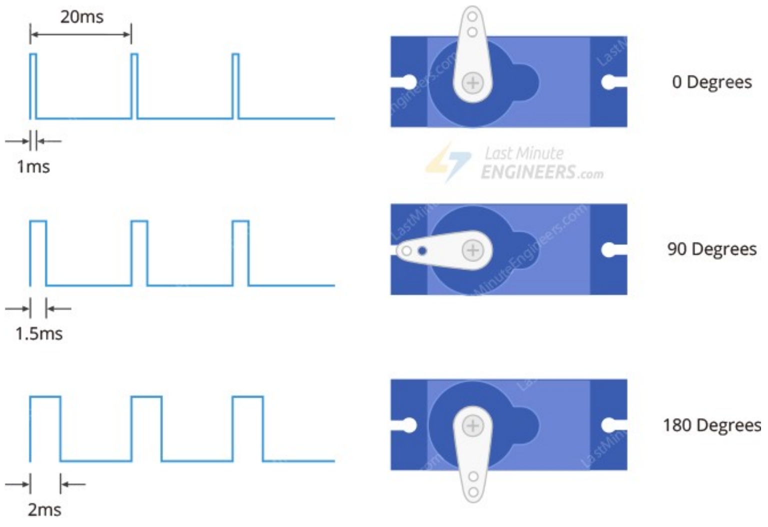


It contains a small DC motor connected to the output shaft through the gears. The output shaft drives a servo arm and is also connected to a potentiometer (pot). The potentiometer provides position feedback to the servo control unit where the current position of the motor is compared to the target position. According to the error, the control unit corrects the actual position of the motor so that it matches the target position.



## How Servo Motors Work?

You can control the servo motor by sending a series of pulses to the signal line. A conventional analog servo motor expects to receive a pulse roughly every 20 milliseconds (i.e. signal should be 50Hz).

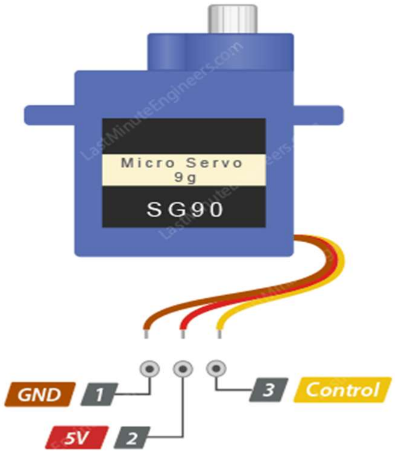


The length of the pulse determines the position of the servo motor.

- If the pulse is high for 1ms, then the servo angle will be zero.
- If the pulse is high for 1.5ms, then the servo will be at its center position.
- If the pulse is high for 2ms, then the servo will at 180 degrees.
- Pulses ranging between 1ms and 2ms will move the servo shaft through the full 180 degrees of its travel.

## Servo Motor Pinout

Servo motors typically have three connections and are as follows:

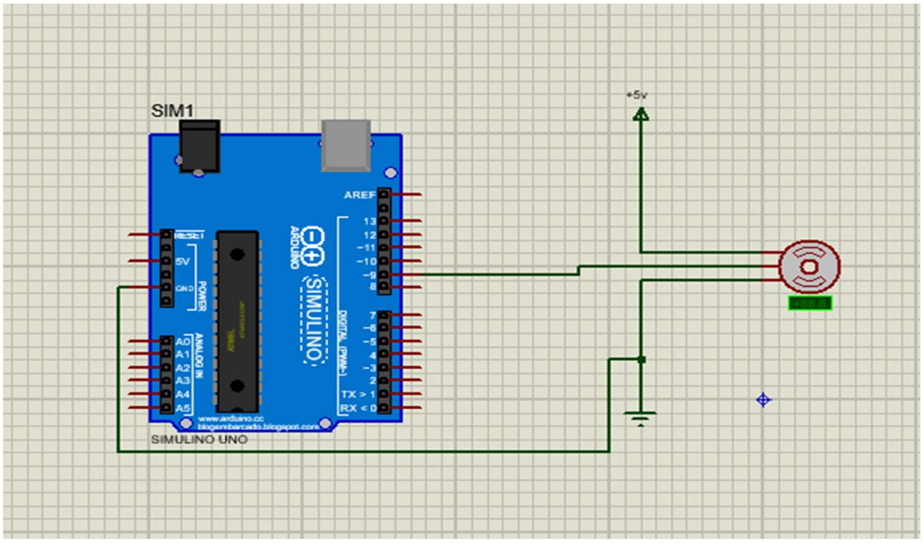


GND is a common ground for both the motor and logic.

5V is a positive voltage that powers the servo.
Control is input for the control system.

## Simulation of servo using Proteus

After opening Proteus click on open icon, then navigate to the simulation folder “servo” and open **servo.pdsprj** file.

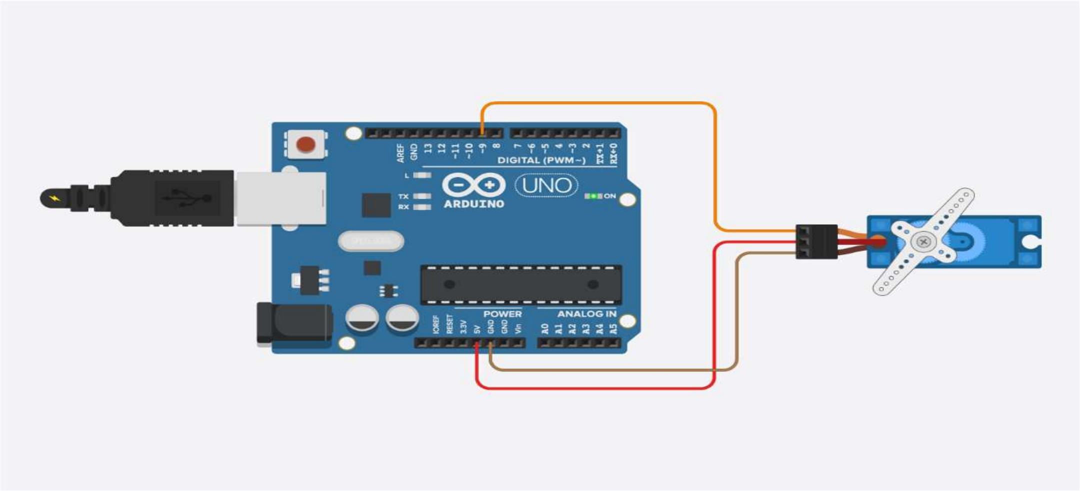


Double click on Arduino board and upload the **servo. ino.standard.hex** file, then run the simulation from animation panel.

## Implementation on Arduino

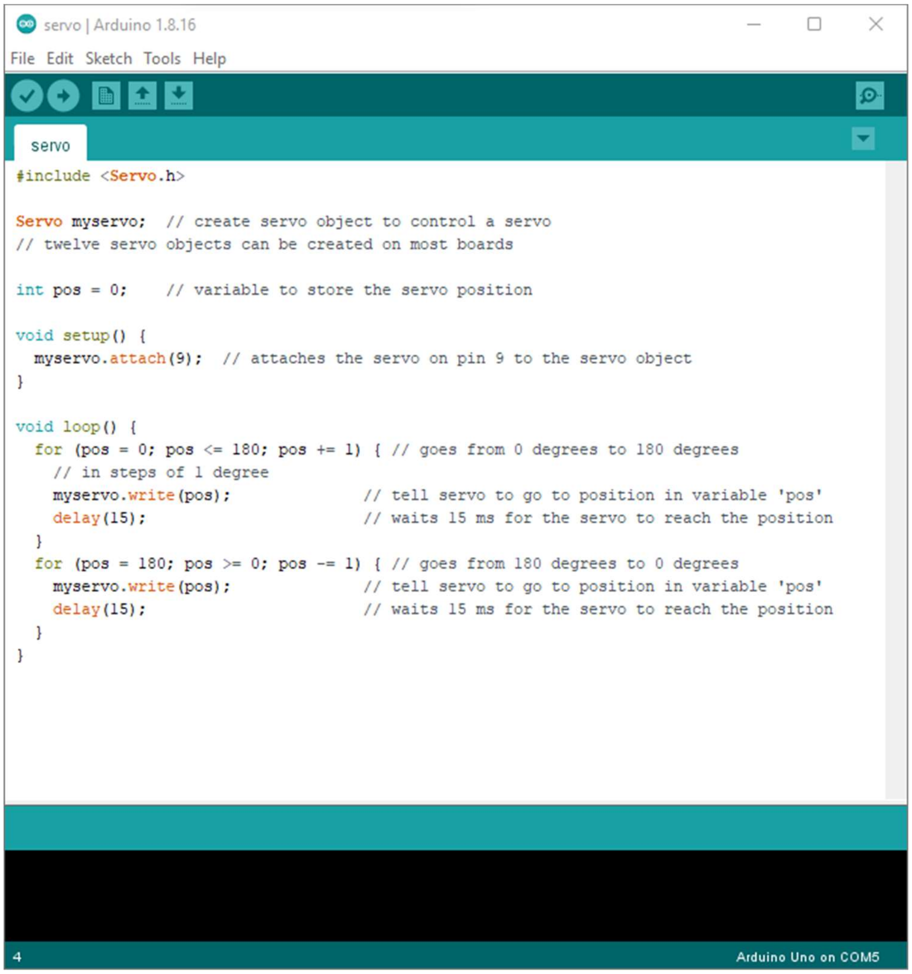
In this Project the SG90 Micro Servo Motor is used. It runs on 4.8-6VDC (5V Typical) and can rotate approximately 180 degrees (90 in each direction). It consumes around 10mA at idle and 100mA to 250mA when moving, it can be power it up through 5-volt output on the Arduino.

As show in figure below connect the Red wire to the 5V on Arduino and Brown wire to ground. Finally connect the Yellow wire to the PWM (**Pulse Width Modulation**) enabled pin 9.



## Arduino Code

After that open Arduino IDE then select **file > open…**, navigate to servo folder then open **servo.ino** sketch.



```
#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 += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the position
  }
}
```

Use the verify button to compile the code then send it to the Arduino Board by clicking on upload button. You will immediately see the motor moving in one direction and then going back in another.

## Objective

Control of Stepper motor rotation using Arduino UNO.

## Required Equipment's

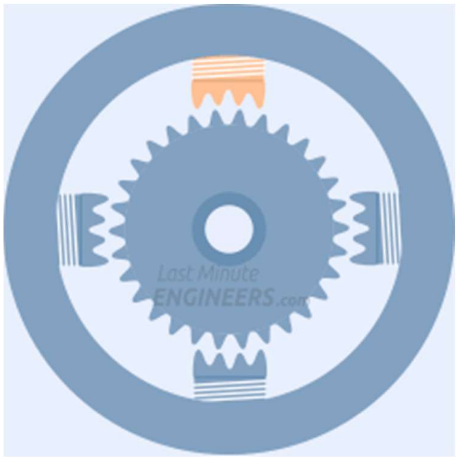
Arduino Uno, 28BYJ-48 Stepper Motor, **ULN2003** Driver, 6 jumper Cables.

## What is a Stepper Motor?

Stepper motors are a special type of brushless motors that divides a full rotation into a number of equal "steps". They are usually found in desktop printers, 3D printers, CNC milling machines, and anything else that requires precise positioning control. The 28BYJ-48 stepper motors usually come with a ULN2003 based driver board which makes them super easy to use.

## How Stepper Motors Work?

Stepper motors use a cogged wheel (having 32 teeth) and four electromagnets to rotate the wheel one 'step' at a time. Each HIGH pulse sent, energizes the coil, attracts the nearest teeth of the cogged wheel and drives the motor one step.



The way you pulse these coils greatly affects the behavior of the motor.

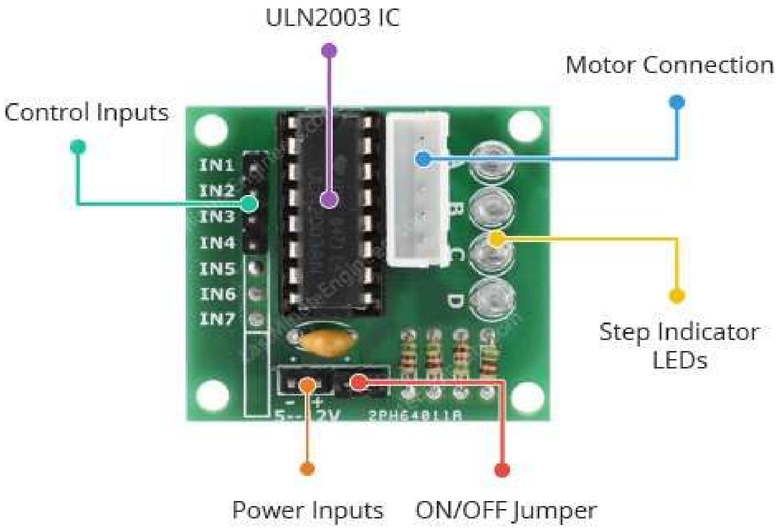
- The sequence of pulses determines the spinning direction of the motor.
- The frequency of the pulses determines the speed of the motor.
- The number of pulses determines how far the motor will turn.

## The 28BYJ-48 Stepper Motor

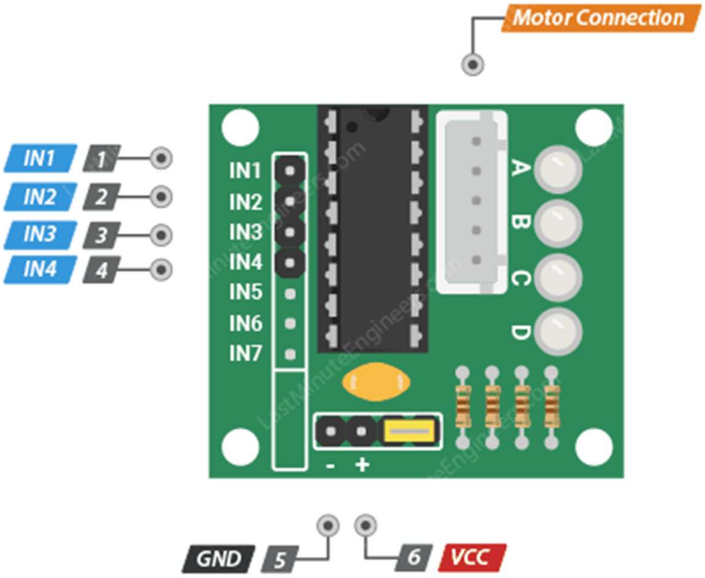
The 28BYJ-48 is a 5-wire unipolar stepper motor that runs on 5 volts. It can be positioned accurately and precisely, one 'step' at a time which make it quite reliable since the motor does not use contact brushes. The 28BYJ-48 stepper motors have 2038 steps per revolution and it comes with a ULN2003 based driver board.

## The ULN2003 Driver

The ULN2003 is one of the most common motor driver ICs, consisting of an array of 7 Darlington transistor pairs, each pair is capable of driving loads of up to 500mA and 50V. Four out of seven pairs are used on this board. It has a connector that mates the motor wires perfectly which makes it very easy to connect the motor to the board. There are also connections for four control inputs as well as power supply connections. The board has four LEDs that show activity on the four control input lines (to indicate stepping state). Which also comes with an ON/OFF jumper to isolate power to the stepper Motor.



## The ULN2003 Driver Pinout



IN1 – IN4 pins are used to drive the motor. Should be connected to a digital output pins on the Arduino.

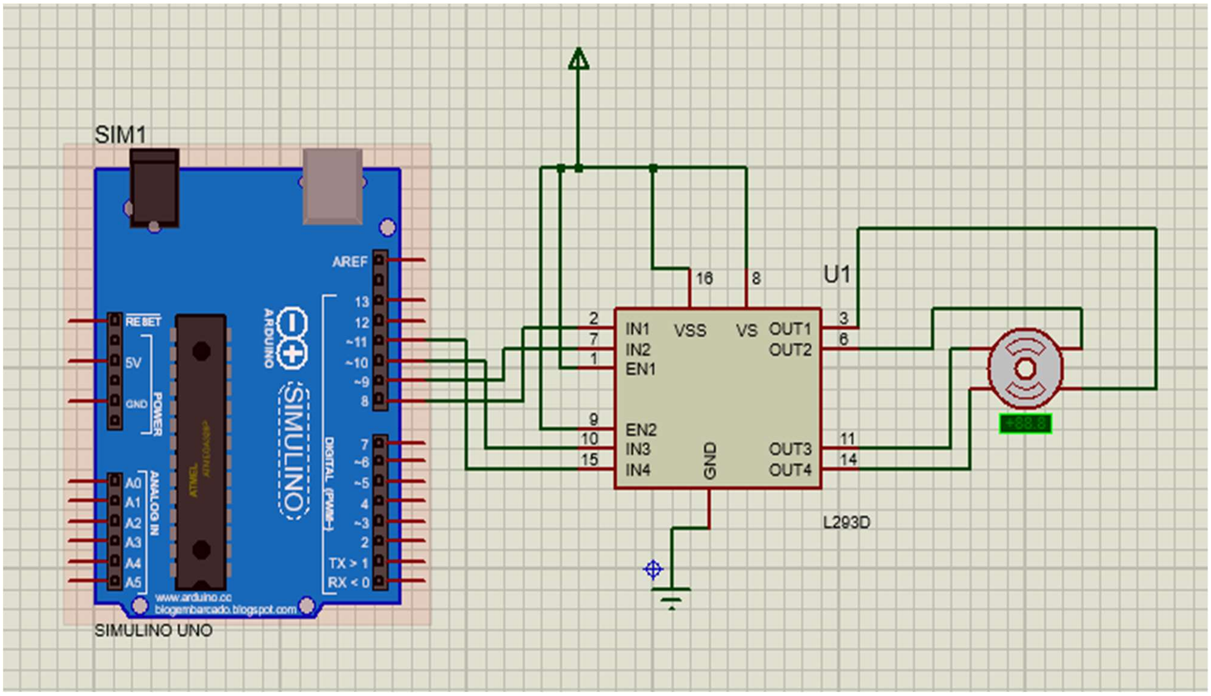
GND is a common ground pin.

VCC pin supplies power for the motor. Connect it to an external 5V power supply. Because the motor draws too much power, it's better to never use the 5V power from your Arduino to power this stepper motor.

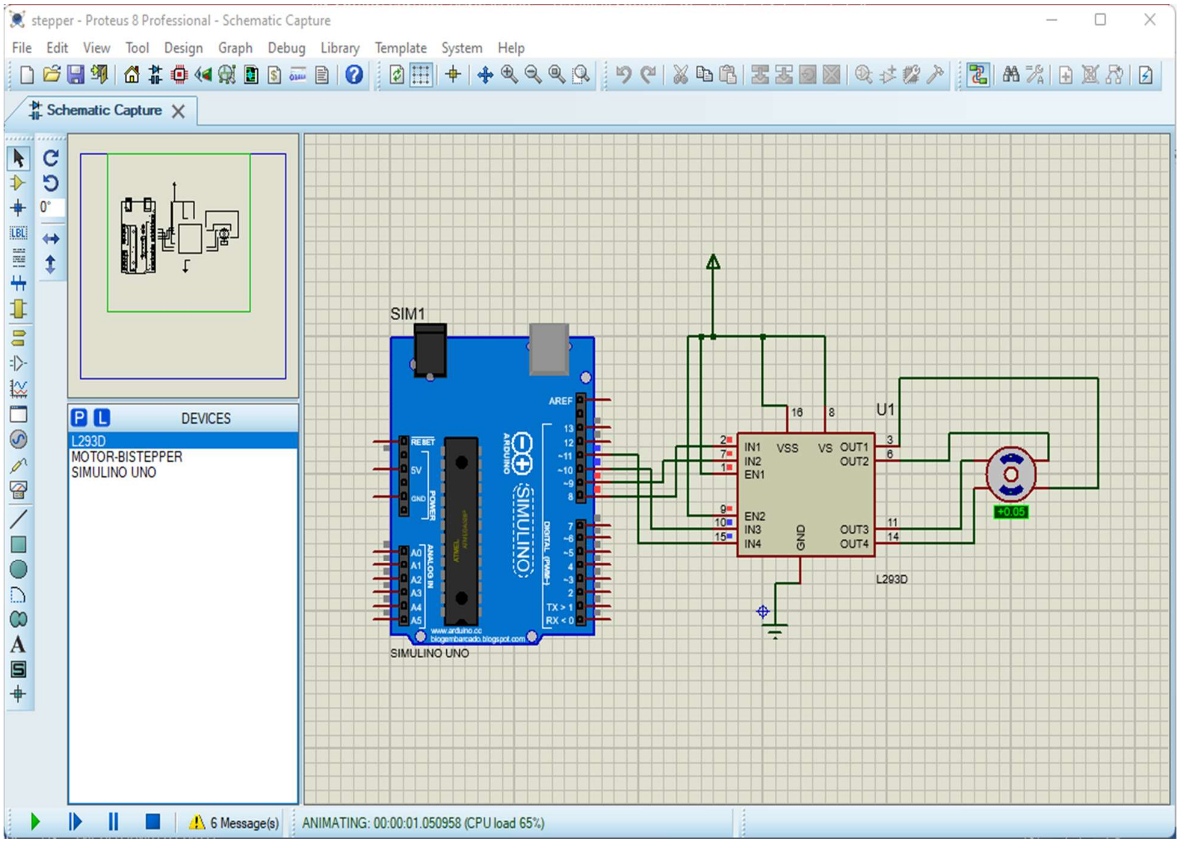
Motor Connector This is where the motor plugs into. The connector is keyed, so it only goes in one way.

## Simulation of stepper motor using Proteus

Open Proteus 8 then click on open icon 📂, navigate to the simulation folder "stepper" and open **stepper.pdsprj** project file. The following window will show up:



In the Editing window area Double click on Arduino board and upload the “**stepper.ino.standard.hex”** file, then run the simulation from animation panel.



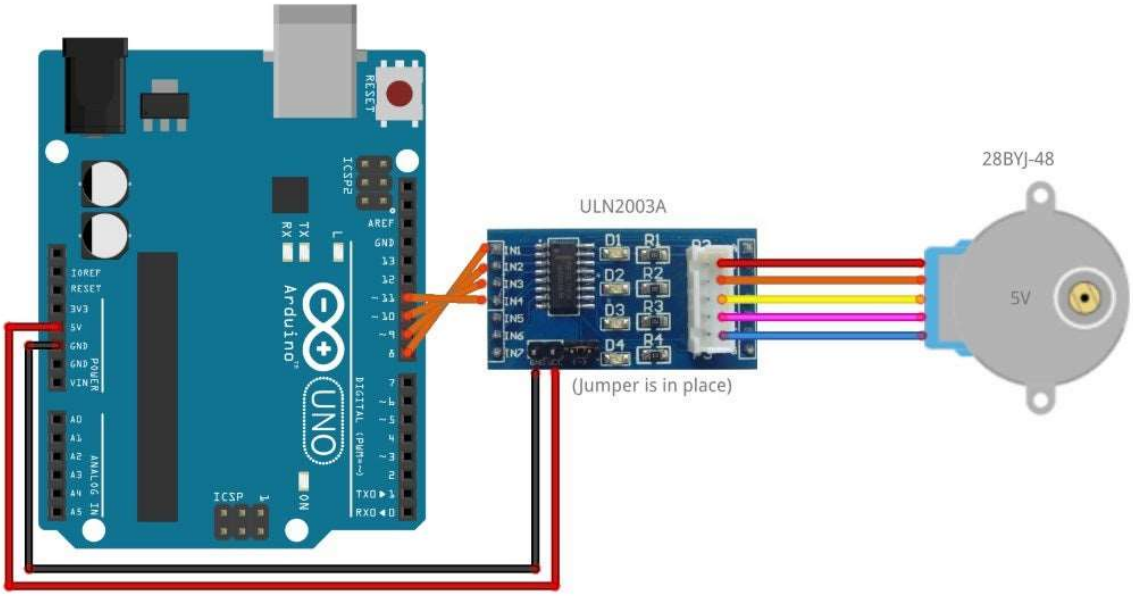
L293D
MOTOR-BISTEPPER
SIMULINO UNO

6 Message(s)

ANIMATING: 00:00:01.050958 (CPU load 65%)

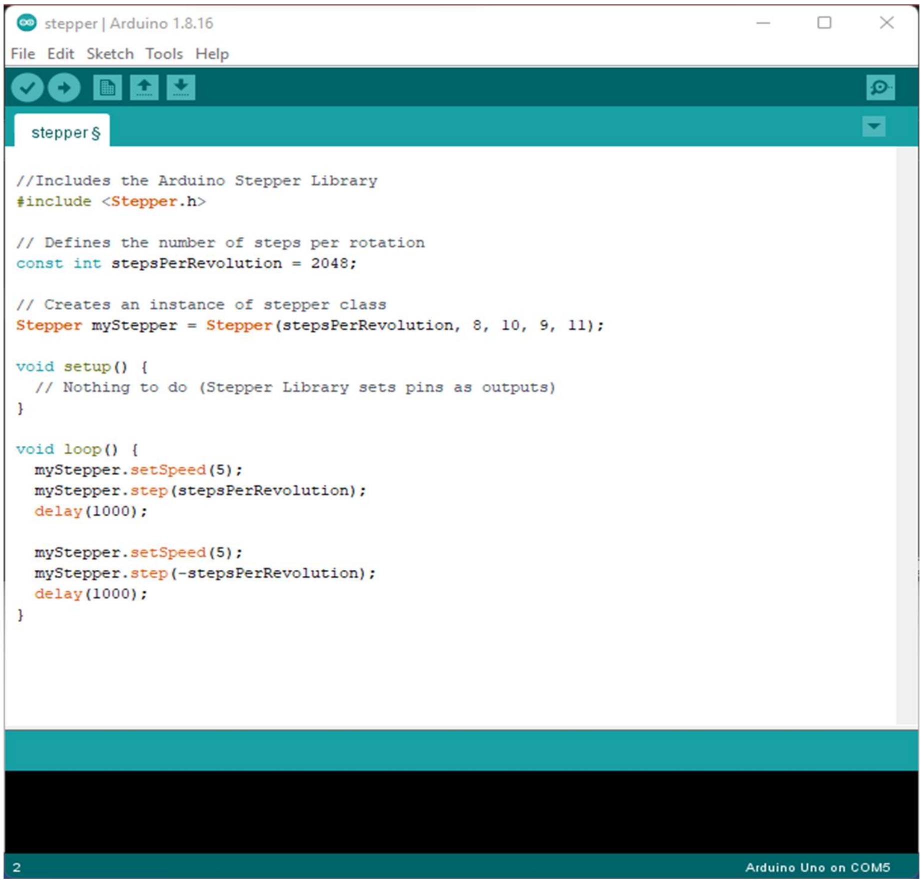
## Implementation on Arduino

Connect the available equipment's similar to the illustration shown below:



## Arduino Code

Navigate to the "stepper" simulation folder, and open the "**stepper.ino**" sketch, compiled using the verify button then upload it to the Arduino board. You will immediately see the motor sweep from clock wise direction to the counterclockwise direction.



```
//Includes the Arduino Stepper Library
#include <Stepper.h>

// Defines the number of steps per rotation
const int stepsPerRevolution = 2048;

// Creates an instance of stepper class
Stepper myStepper = Stepper(stepsPerRevolution, 8, 10, 9, 11);

void setup() {
  // Nothing to do (Stepper Library sets pins as outputs)
}

void loop() {
  myStepper.setSpeed(5);
  myStepper.step(stepsPerRevolution);
  delay(1000);

  myStepper.setSpeed(5);
  myStepper.step(-stepsPerRevolution);
  delay(1000);
}
```