

| Sl No. | Name of the Experiment | Page No. | Date of Experiment | Date of Submission | Remarks |
|--------|--|----------|--------------------|--------------------|---------|
| 1. | Learn the basics of Arduino boards & features | 8 | | | |
| 2. | Arduino program to blink an LED & implement traffic signal. | 2. | | | |
| 3. | Program to vary the intensity of LED based on LDR | 4 | | | |
| 4. | Toggle LED by pressing a button & to implement switch debounce circuit | 7 | | | |
| 5. | Implement Serial communication event. | 3 | | | |
| 6. | Implement temperature & humidity sensor & switch on an LED if temperature is too hot | 5 | | | |
| 7. | Drive a DC motor & Stepper motor | 14 | | | |
| 8. | Implement Ultrasonic Sensor to measure distance to an obstacle and "buzz" when too close to object | 6 | | | |
| 9. | Implement 16x2 LCD display & display temp & current date & time. | 10-11 | | | |
| 10. | Implement GSM Module & send SMS using some carrier to cellphone. | 12-13 | | | |
| 11. | Basics of Raspberry Pi | 9 | | | |
| 12. | Implement MQTT Protocol & publish data | 1 | | | |

OUTPUT :

| | | | |
|----------------|---|----------|-------------|
| Just published | 3 | to topic | Temperature |
| Just published | 5 | to topic | Temperature |
| Just published | 7 | to topic | Temperature |
| Just published | 1 | to topic | Temperature |
| Just published | 2 | to topic | Temperature |

Name of Experiment.....MQTT.....
Experiment No.....12..... Date.....
Experiment Result.....

Page No. 1

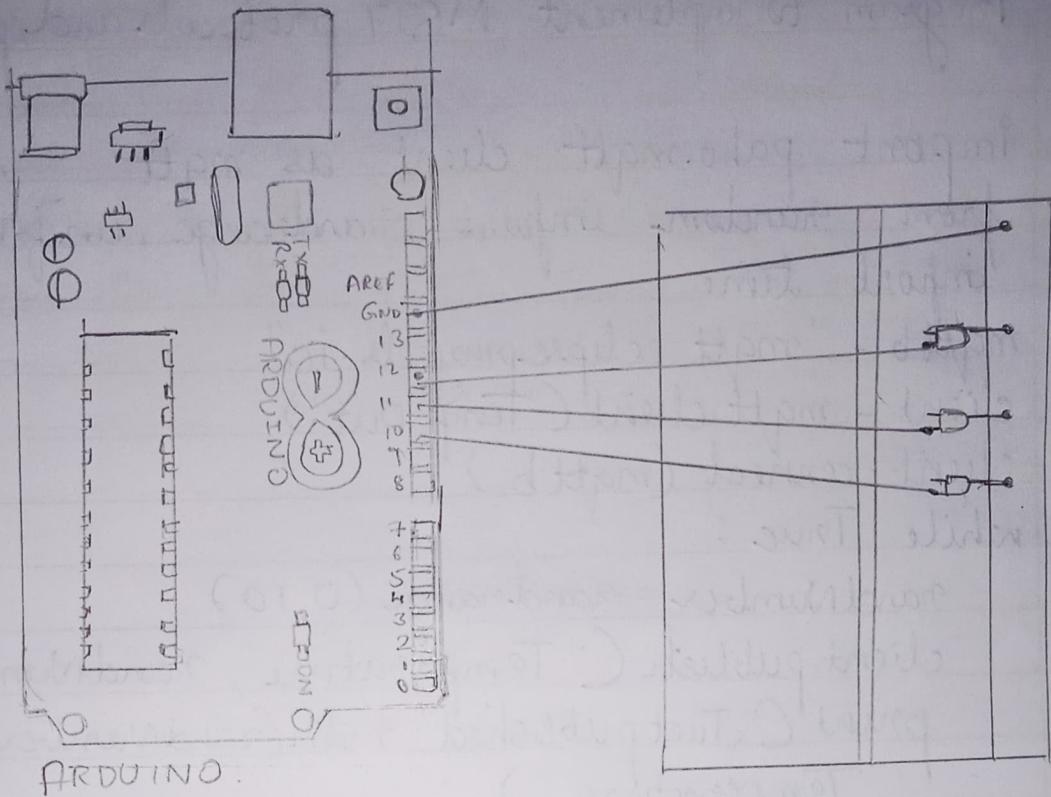
```
12. Program to implement MQTT protocols and publish some data.  
import paho.mqtt.client as mqtt  
from random import randint, uniform  
import time  
mattb = "matt.eclipseprojects.io"  
client = mqtt.Client("Temp-out")  
client.connect(mattb)  
  
while True:  
    randNumber = randint(0,10)  
    client.publish("Temperature", randNumber)  
    print("Just published "+str(randNumber)+" to topic  
        Temperature")  
  
    time.sleep(1)
```

OUTPUT

COMPONENTS

REQUIRED

- 1) Arduino
- 2) LED lights
(Green, Yellow,
Red)
- 3) Jumper Wires.
- 4) Bread Board



2. Arduino programs to implement a traffic signal.

```
#define green 11
```

```
#define yellow 10
```

```
#define red 12
```

```
void setup() {
```

```
  pinMode(green, output);
```

```
  pinMode(yellow, output);
```

```
  pinMode(red, output);
```

```
}
```

```
void lights (int value1, int value2, int value3)
```

```
  digitalWrite (value1, high);
```

```
  digitalWrite (value2, low);
```

```
  digitalWrite (value3, low);
```

```
}
```

```
void loop () {
```

```
  lights (green, yellow, red);
```

```
  delay (1000);
```

```
  lights (yellow, green, red);
```

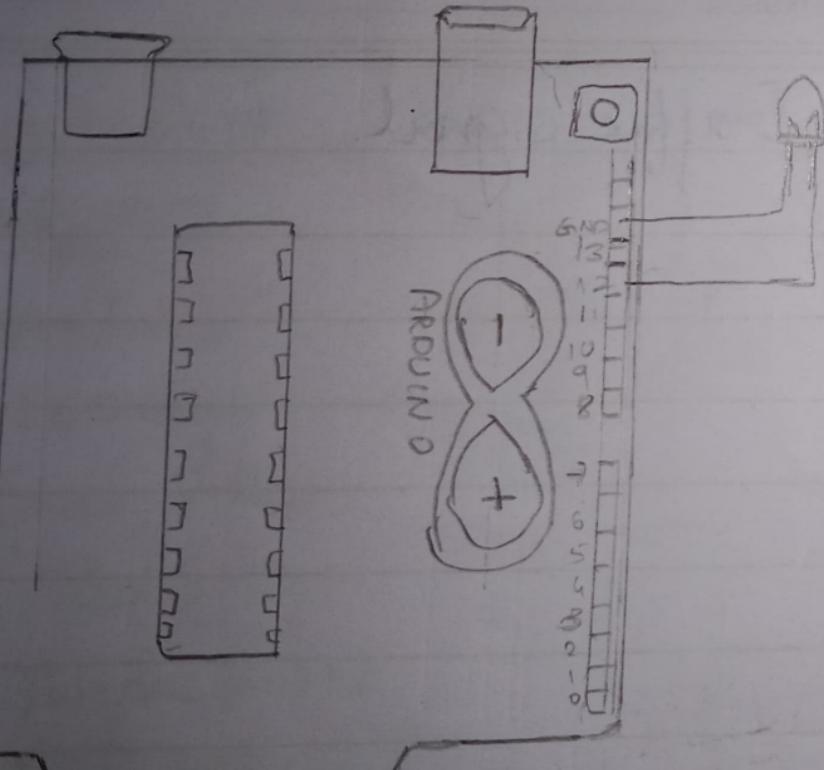
```
  delay (1000);
```

```
  lights (red, yellow, green);
```

```
  delay (1000);
```

```
}
```

OUTPUT



COMPONENTS REQUIRED

- 1) Arduino
- 2) LED
- 3) Jumpers

5. Arduino program to implement a serial communication

```
void setup() {
```

```
pinMode(12, OUTPUT);
```

```
digitalWrite(12, LOW);
```

```
Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
if (Serial.available() > 0) {
```

```
char letter = Serial.read();
```

```
if (letter == '1') {
```

```
digitalWrite(12, HIGH);
```

```
Serial.println("The LED is ON");
```

```
}
```

```
else if (letter == '0') {
```

```
digitalWrite(12, LOW);
```

```
Serial.println("The LED is OFF");
```

```
}
```

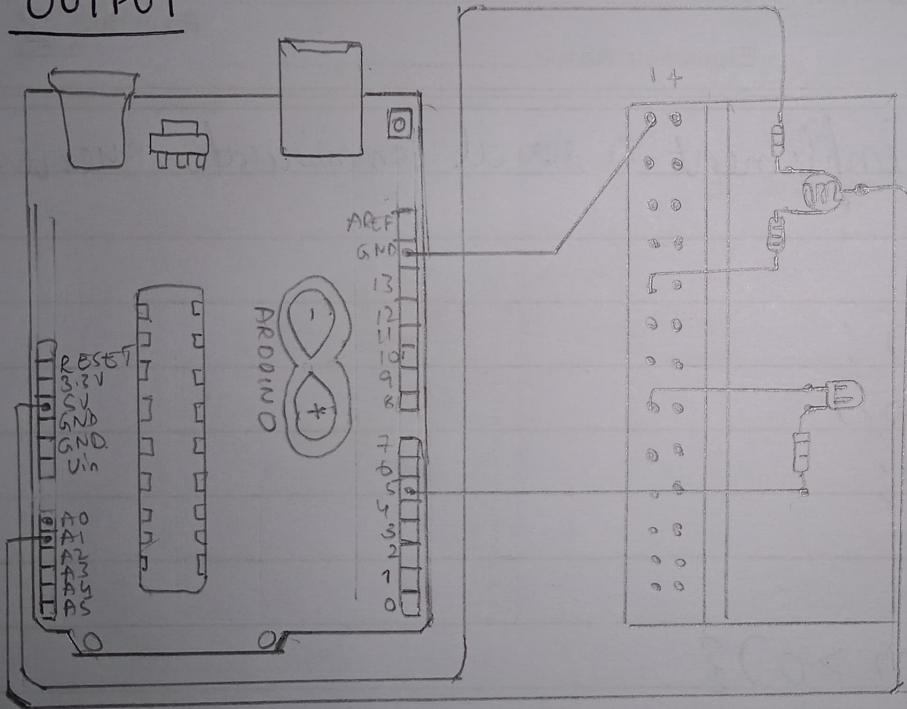
```
else { Serial.println("Invalid Input");
```

```
}
```

```
delay(200);
```

```
}
```

OUTPUT



COMPONENTS REQUIRED

- 1) Arduino
- 2) LDR
- 3) LED
- 4) Jumper
- 5) Bread board
- 6) Resistors

Name of Experiment..... Vary intensity of light

Experiment No..... 3.

Date.....

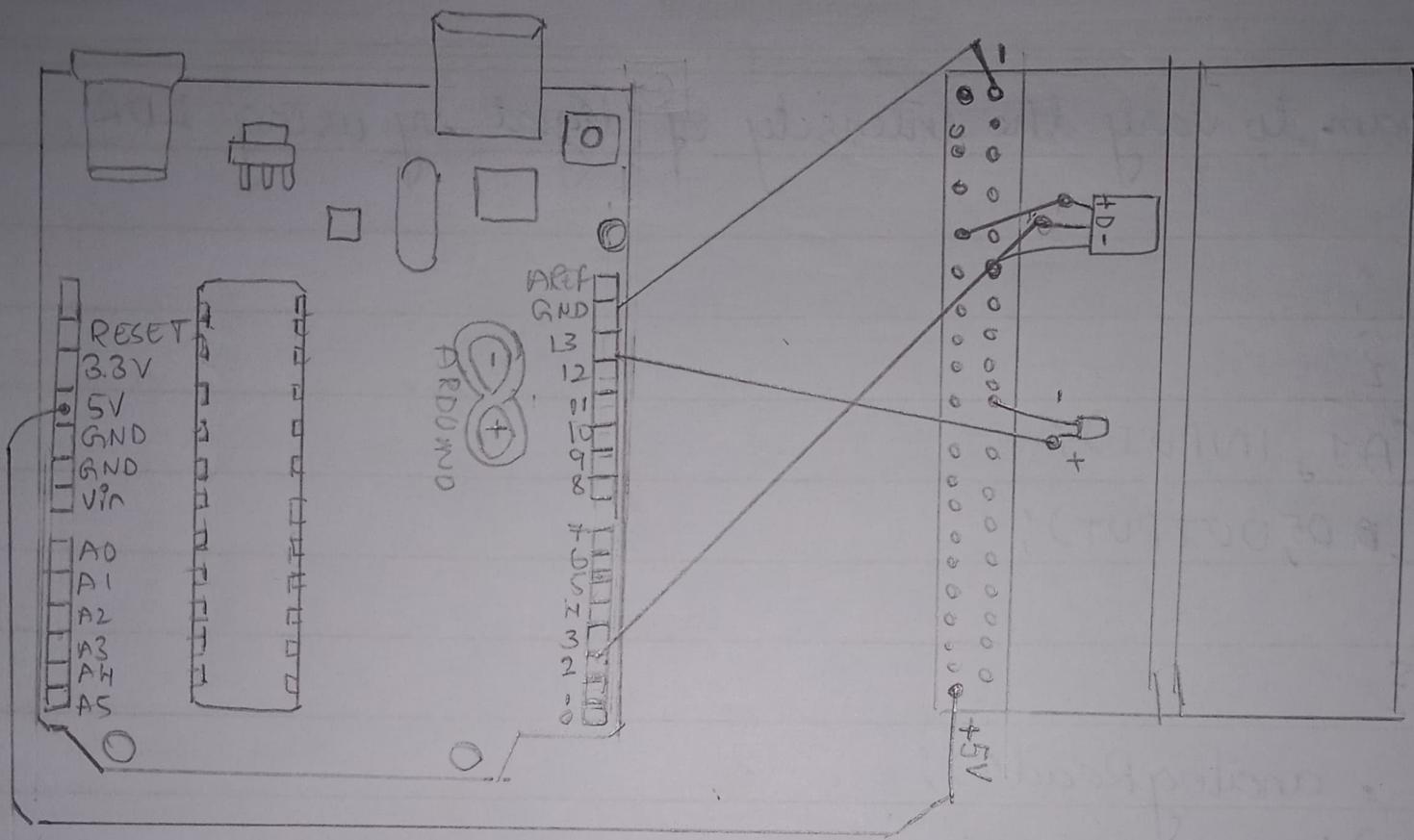
Experiment Result.....

Page No. 4.

3. Arduino program to vary the intensity of light by using LDR

```
int intensity;  
void setup() {  
    pinMode (A1, INPUT);  
    pinMode (D05, OUTPUT);  
}  
void loop() {  
    intensity = analogRead();  
    delay(1000);  
    analogWrite(D05, intensity);  
}
```

OUTPUT



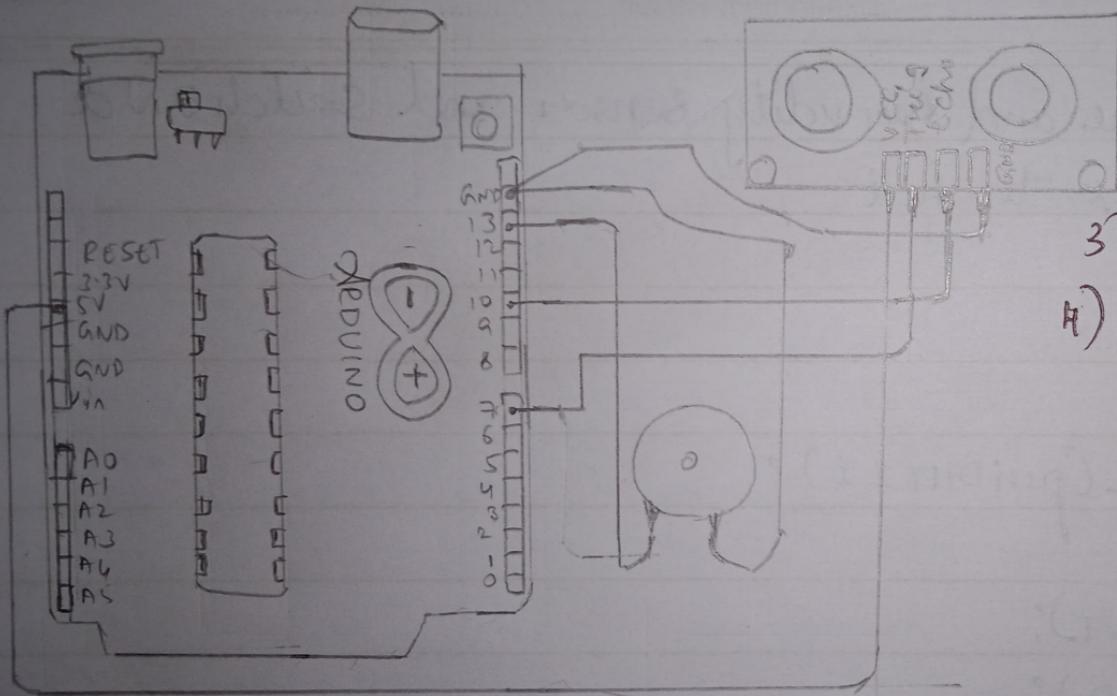
COMPONENTS REQUIRED

- 1) Arduino
- 2) DHT Sensor
- 3) LED
- 4) Jumpers
- 5) BreadBoard

6 Implement temperature and humidity sensor and switch ON a LED if temperature is too hot

```
#include <SimpleDHT.h>
int pinDHT11 = 2;
SimpleDHT11 dht11(pinDHT11);
void setup() {
    pinMode(2, OUTPUT);
    Serial.begin(9600);
    Serial.println("Temperature & Humidity");
}
void loop() {
    byte temp = 0, humid = 0;
    int err = SimpleDHTerrSuccess;
    if ((err = dht11.read(&temp, &humid, NULL)) != SimpleDHTerrSuccess)
    {
        Serial.println("Read DHT11 failed, err = ");
        Serial.println(SimpleDHTerrCode(err));
        Serial.print("DHT11 duration ");
        Serial.println(*SimpleDHTerrDuration(err));
        return;
    }
    Serial.print((int)temp);
    Serial.print(" °C ");
    Serial.print((int)humid);
    Serial.println(" H ");
    if ((int)temp > 32)
        digitalWrite(13, HIGH);
    else
        digitalWrite(13, LOW);
}
```

OUTPUT



COMPONENTS REQUIRED

- 1) Arduino
- 2) Ultrasonic Sensor
- 3) Buzzers
- 4) Jumpers

8 Implement Ultrasonic Sensor to measure distance to an obstacle and buzz when too close to object

```
int trigpin = 7;
```

```
int echopin = 10;
```

```
int led = 13;
```

```
void setup () {
```

```
Serial.begin (9600);
```

```
pinMode (led, OUTPUT);
```

```
pinMode (trigpin, OUTPUT);
```

```
pinMode (echopin, INPUT);
```

```
}
```

```
void loop () {
```

```
long duration, distance;
```

```
digitalWrite (trigpin, LOW);
```

```
delayMicroseconds (2);
```

```
digitalWrite (trigpin, HIGH);
```

```
delayMicroseconds (10);
```

```
duration = pulseIn (echopin, HIGH);
```

```
distance = (duration / 2) / 29.1;
```

```
Serial.print ("distance");
```

```
delay (100);
```

```
if (distance < 80)
```

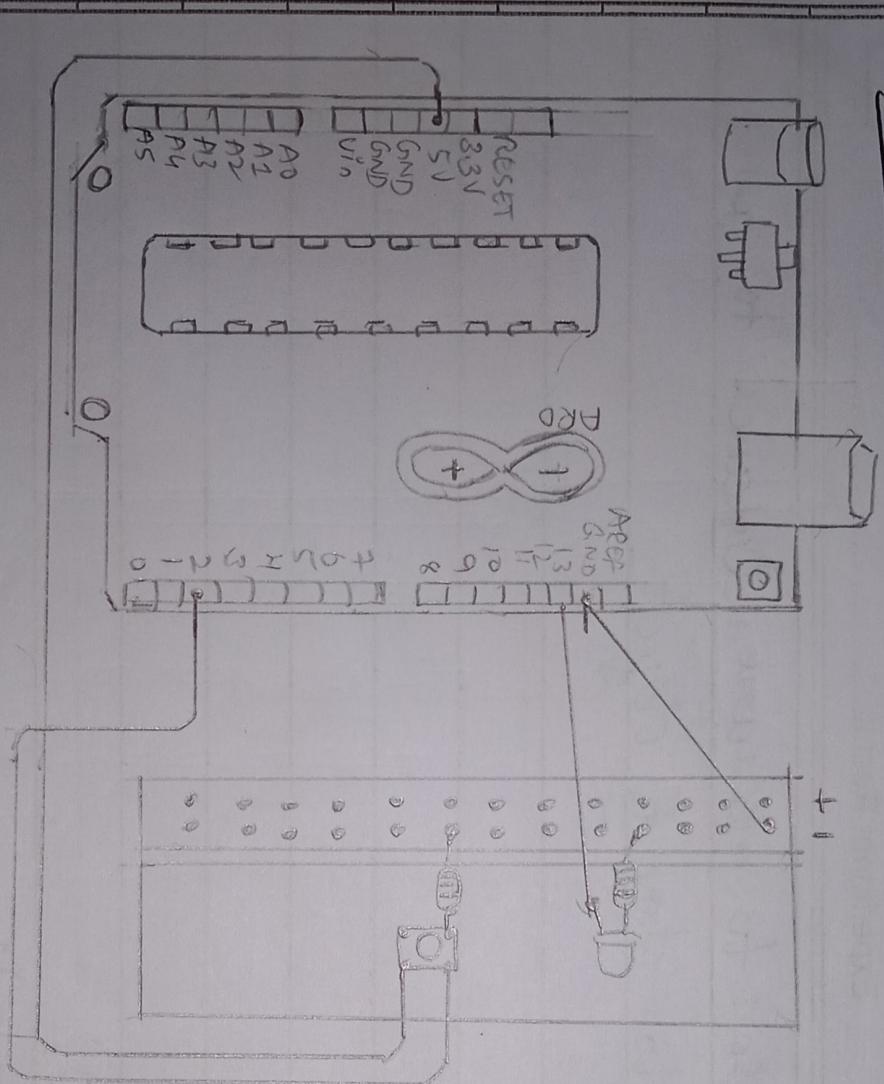
```
digitalWrite (led, HIGH);
```

```
else if (distance >= 5)
```

```
digitalWrite (led, LOW);
```

```
delay (100);
```

Output



Components Required

- 1) Arduino
- 2) Button
- 3) LED
- 4) Jumper
- 5) Breadboard
- 6) Resistor

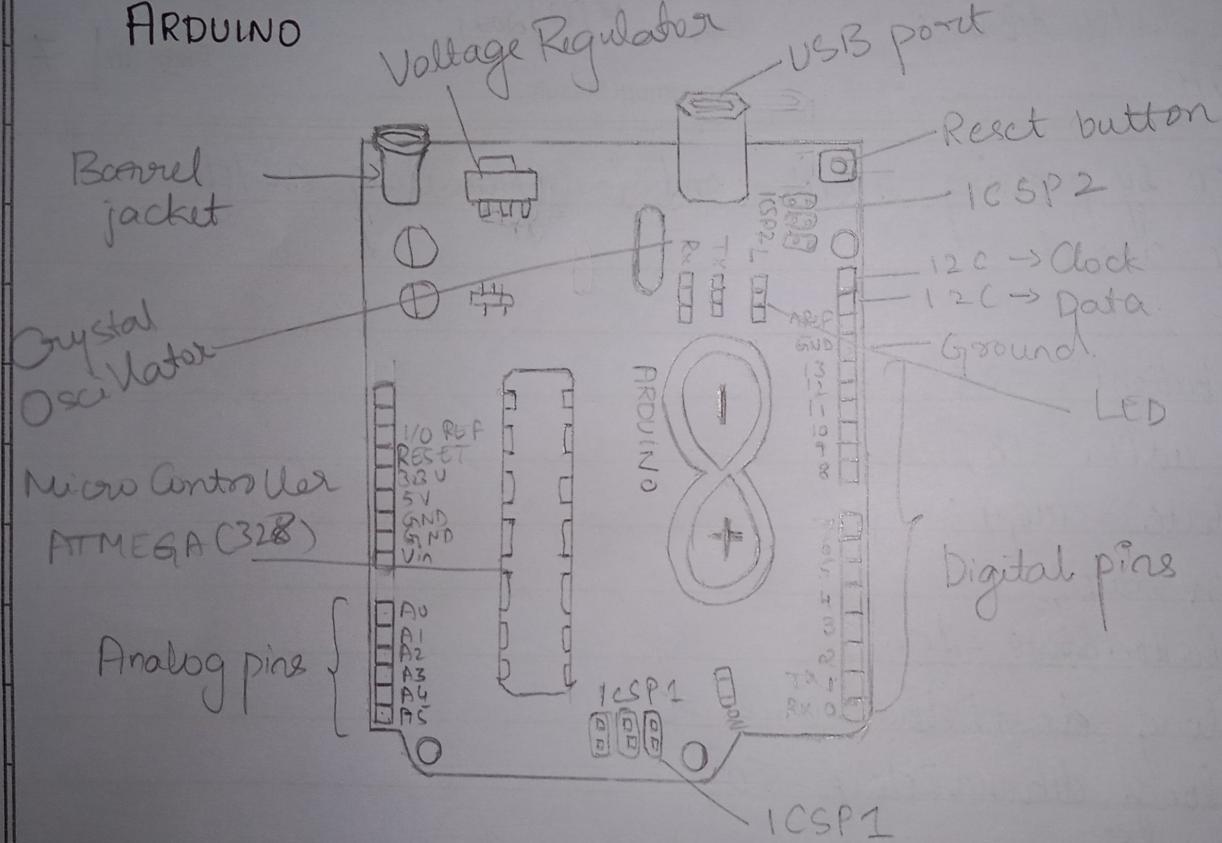
4 Toggle LED by pressing a button and to implementing switch debounce circuit.

```

const int buttonPin = 2;
const int ledPin = 13;
int ledState = HIGH;
int buttonState;
int lastButtonState = LOW;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;
void setup() {
    pinMode(buttonPin, INPUT);
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, ledState);
}
void loop() {
    int reading = digitalRead(buttonPin);
    if (reading != lastButtonState) {
        lastDebounceTime = millis();
    }
    if ((millis() - lastDebounceTime) > debounceDelay) {
        if (reading != buttonState) {
            buttonState = reading;
            if (buttonState == HIGH)
                ledState = !ledState;
        }
    }
    digitalWrite(ledPin, ledState);
    lastButtonState = reading;
}

```

ARDUINO



1. Learn the basics of Arduino boards, features and pinouts of Arduino UNO,

Arduino is an open source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and their kits for building digital devices.

Specifications

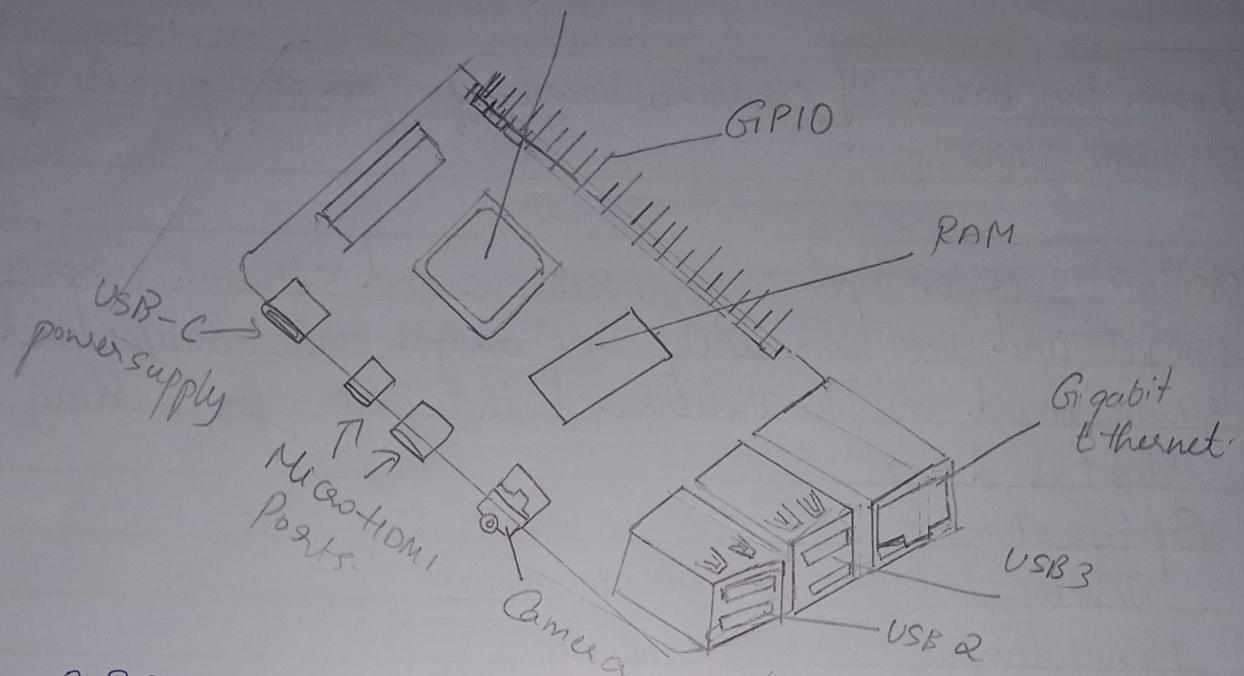
| | |
|------------------|---|
| Developer | arduino.cc |
| Manufacturer | Arduino |
| Type | Single-board microcontroller |
| Operating system | None (default) / XNU |
| CPU | Atmel AVR (8-bit), ARM Cortex-M0+ (32-bit), ARM Cortex-M3 (32-bit), Intel Quark (x86) (32-bit) |
| Memory | SRAM. |
| Storage | Flash, EEPROM. |

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or breadboards and other circuits.

The boards feature several communication interfaces, including Universal Serial Bus (USB) on some models, which is used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages.

RASPBERRY PI

Processor



| | | |
|------------------|-------|-------------------------------------|
| 3v3 power | 1 2 | - 5V power |
| GPIO 2 (SDA) | 3 4 | - 5V power |
| GPIO 3 (SCL) | 5 6 | Ground |
| 10 M (GPCLK0) | 7 8 | GPIO 14 (TXD) |
| Ground | 9 10 | GPIO 15 (RXD) |
| GPIO 17 | 11 12 | GPIO 18 (PCM-CLK) |
| GPIO 27 | 13 14 | Ground |
| GPIO 22 | 15 16 | GPIO 23 |
| 3v3 power | 17 18 | GPIO 24 |
| GPIO 10 (MOSI) | 19 20 | Ground |
| GPIO 9 (MISO) | 21 22 | GPIO 25 |
| GPIO 11 (CLK) | 23 24 | GPIO 8 (CEO) |
| Ground | 25 26 | GPIO 7 (CE1) |
| GPIO 0 (ID-SD) | 27 28 | GPIO 1 (ID-SC) |
| GPIO 5 | 29 30 | Ground |
| GPIO 6 | 31 32 | GPIO 16, 2 (PWM0) |
| GPIO 13 (PWM1) | 33 34 | GPIO 20 (PCM-DIN) Ground |
| GPIO 19 (PCM-FS) | 35 36 | GPIO 16 |
| GPIO 26 | 37 38 | GPIO 20 (PCM-DIN) |
| Ground | 39 40 | GPIO 21 (PCM-DOUT) |

11. Learn the basics of Raspberry Pi, features, pinout and configuration.

Raspberry Pi is a series of small single-board computers (SBC) developed in the UK by Raspberry Pi Foundation, in association with Broadcom.

Specifications

Also known as

Raspi, RPi

Release date

24 February 2012

Operating System

Raspberry Pi OS, FreeBSD, Linux, Windows IoT, NetBSD, RISCOS

CPU

Pi 3 A+: 1.4GHz quad-core A53 64-bit

Pi 4B: 1.5GHz quad-core A72 64-bit

Memory

Pi 3 A+: 512MB, Pi 4B: 2, 4, or 8GB

Storage

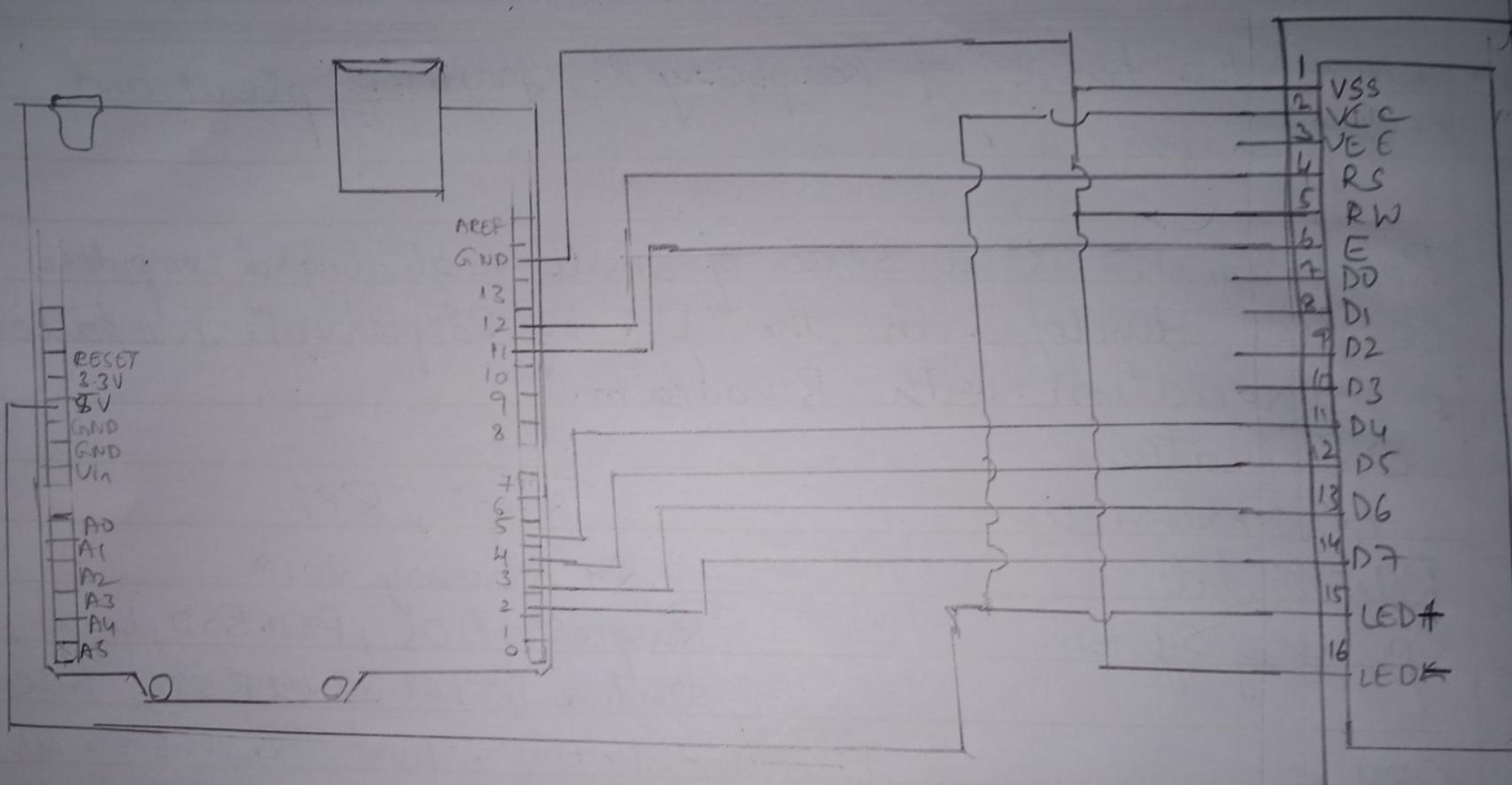
Micro SDHC slot, Universal Serial Bus Mass Storage

Graphics

Pi 3 A+, Pi 4B

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV & uses a standard keyboard and mouse. Raspberry Pi has the ability to interact with the outside world, & has been used in a wide array of digital maker projects. It can be used as replacement for digital PC or Server.

OUTPUT



9. Arduino program to implement 16x2 LCD alphanumeric display and display temperature, date & time

```
#include <DS3231 LiquidCrystal.h>
#include <DS3231.h>
DS3231 rtc(SDA, SCL);
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
    rtc.begin();
    lcd.begin(16, 2);
    rtc.setDOW(SUNDAY);
    rtc.setTime(18, 50, 35);
    rtc.setDate(16, 12, 2018);
}
void loop() {
    lcd.setCursor(0, 0);
    lcd.print("Real Time Clock");
    lcd.setCursor(0, 1);
    lcd.print("Time:");
    lcd.print(rtc.getTimeStr());
    delay(3000);
    lcd.setCursor(0, 1);
    lcd.print("Date:");
    lcd.print(rtc.getDateStr());
    delay(3000);
    lcd.setCursor(0, 1);
    lcd.print("Day:");
}
```

Name of Experiment.....

Date.....

Experiment No..... 9

Experiment Result.....

Page No.

lcd.print(CRTC.get^{DOW}~~DATE~~Str());
delay(3000);
lcd.setCursor(0,1);
lcd.print("Temp: ");
lcd.print(CRTC.getTemp());
lcd.print(" C");
lcd.print(" °");
delay(3000);

3.

10 Arduino program to implement a GSM module and send SMS using some carrier to a cellphone number.

```
#include <SoftwareSerial.h>
#include "Adafruit-FONA.h"
#define FONA_RX 2
#define FONA_TX 3
#define FONA_RST 4
#define FONA_RL_INTERRUPT 0
char sendto[21] = "70xxxxxx xx"; // Number
char message[14] = "Welcome";
String t;
SoftwareSerial fonass = SoftwareSerial(FONA_TX, FONA_RX);
Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
void setup() {
    Serial.begin(115200);
    Serial.println("FONA incoming call");
    Serial.println(F("Initialising ---"));
    delay(5000);
    fonass.begin(14800);
    if (!fona.begin(fonass)) {
        Serial.println(F("Couldn't find FONA"));
    }
    Serial.println(F("FONA is OK"));
    fona.print("AT+CSMP=17,16#1D,0\r");
    fona.sendSMS(sendto, message);
    delay(1000);
}
```

Name of Experiment.....

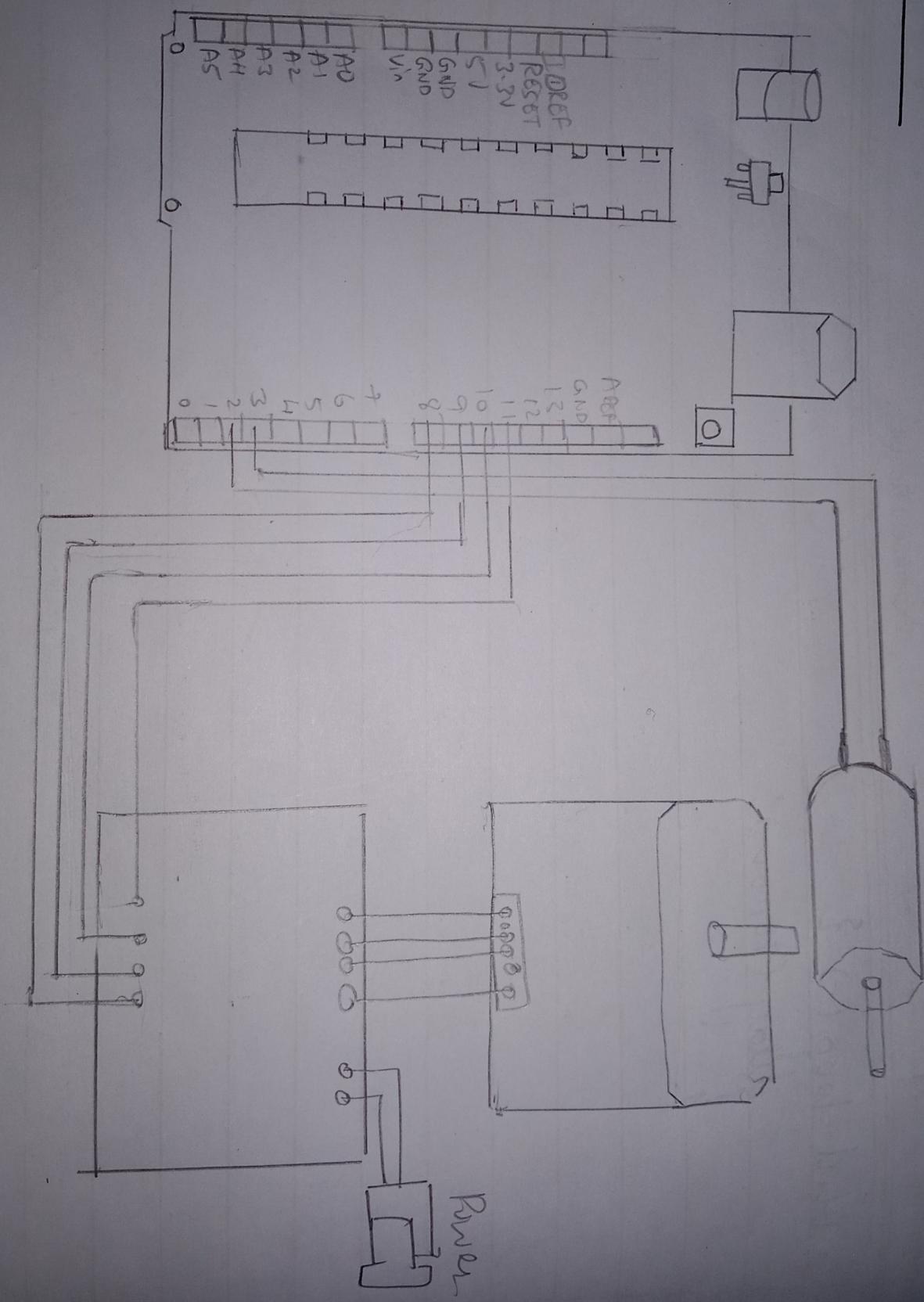
Date.....

Experiment No..... 10

Experiment Result.....

```
void loop() {  
    delay(1000);  
}
```

Output



7 Arduino program to drive a DC and Stepper Motor.

```
#include <Stepper.h>
```

```
int motor1pin1 = 2;
```

```
int motor1pin2 = 3;
```

```
#const int stepsPerRevolution = 200;
```

```
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
```

```
int stepCount = 0;
```

```
void setup() {
```

```
  pinMode(motor1pin1, OUTPUT);
```

```
  pinMode(motor1pin2, OUTPUT);
```

```
}
```

```
void loop() {
```

```
  digitalWrite(motor1pin1, HIGH);
```

```
  digitalWrite(motor1pin2, LOW);
```

```
  delay(1000);
```

```
  digitalWrite(motor1pin2, HIGH);
```

```
  digitalWrite(motor1pin1, LOW);
```

```
  delay(1000);
```

```
  int sensorValue = analogRead(A0);
```

```
  int motorSpeed = map(sensorValue, 0, 1023, 0, 100);
```

```
  if (motorSpeed > 0) {
```

```
    myStepper.setSpeed(motorSpeed);
```

```
    myStepper.step(stepsPerRevolution / 100);
```

```
}
```

```
7.
```