

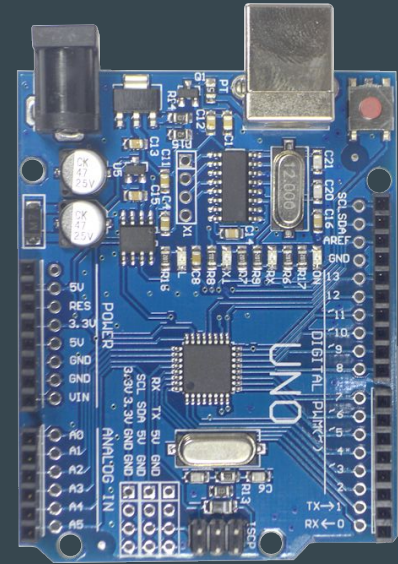
Introduction to Arduino Programming

What is Arduino ?

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

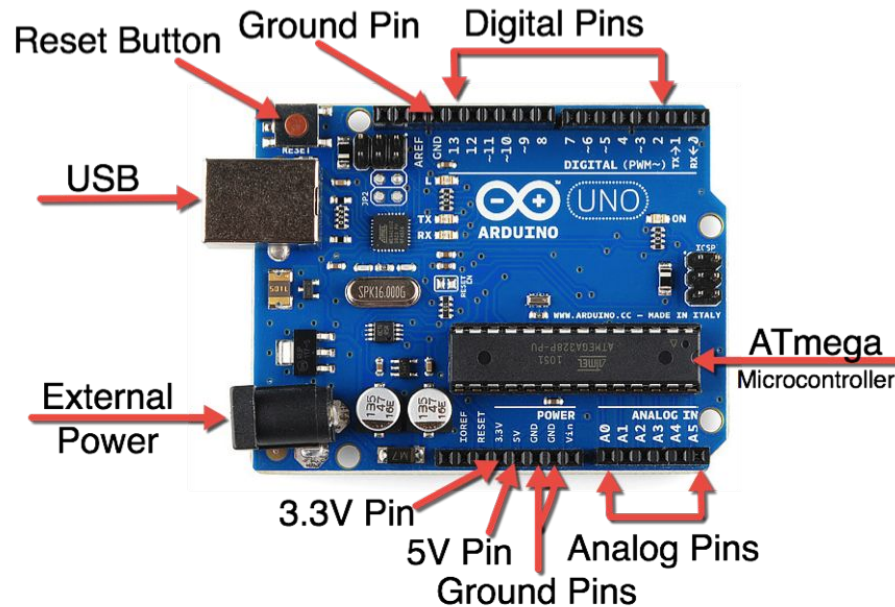
These Single-board microcontroller are intended to make the application of interactive objects or environments more accessible.

These are Designed to make the process of using electronics multi-disciplinary projects more accessible.



An Arduino UNO

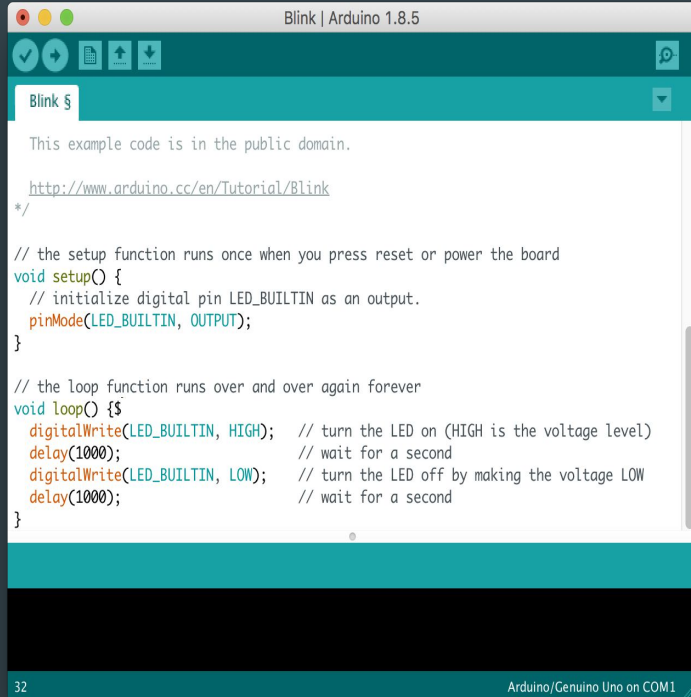
The Basic Arduino Board & Arduino IDE



Core Features of the Arduino UNO Board :
(ATmega 328)

- 14 Digital I/O Pins - 6 PWM
- 32 Kb Flash Memory
- 2 Kb SRAM - 1KB EEPROM
- 16 Mhz Clock Speed
- Input voltage (Recommended) (7 - 12V)
- Input Voltage (Limits) (6 - 20 V)
- USB Power : 5V
- DC Current per I/O : 40mA
- DC Current output for 3.3V Pin : 50mA

Arduino IDE

A screenshot of the Arduino IDE interface. The title bar at the top reads "Blink | Arduino 1.8.5". Below the title bar is a toolbar with icons for opening, saving, compiling, and uploading. The main text area contains the following code:

```
This example code is in the public domain.  
  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {$  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

The status bar at the bottom shows "32" on the left and "Arduino/Genuino Uno on COM1" on the right.

- Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
- It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
- It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment..
- The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
- This environment supports both C and C++ languages.

And Yes , it does have Dark Theme in it :)

Basic Arduino Programming

Parts of the environment :

Save - ctrl/cmd+S

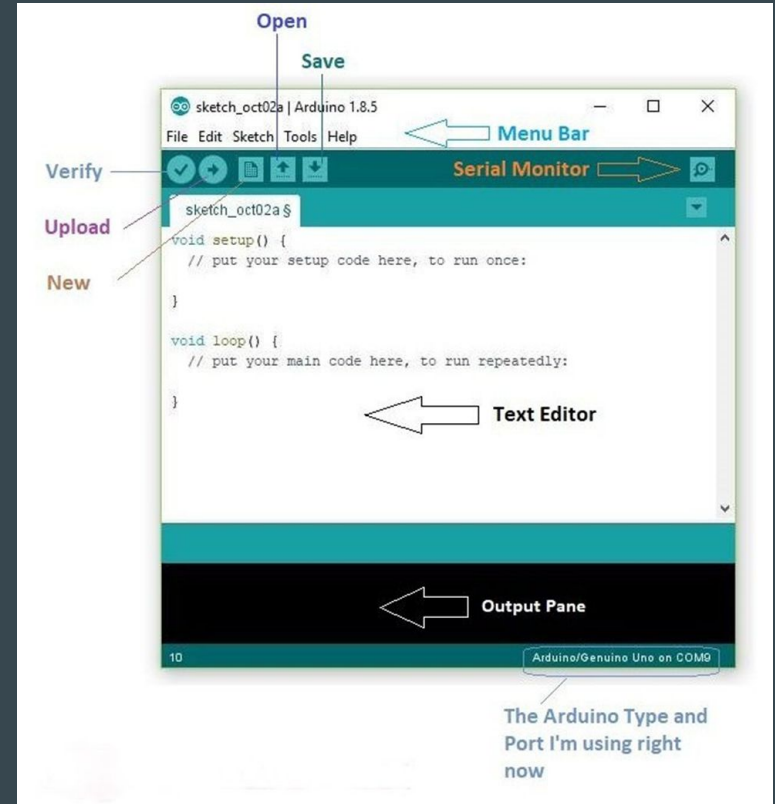
Verify - ctrl/cmd+R

Upload - ctrl/cmd+U

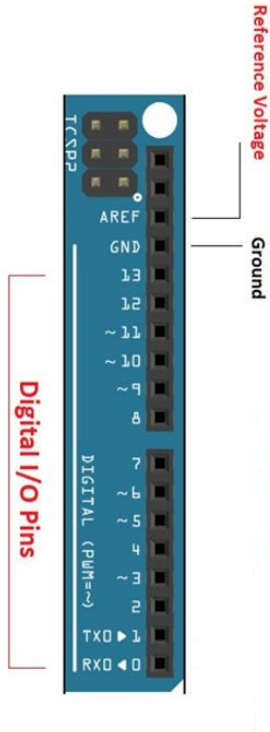
Text editor

Output pane

Board and port used



WORKING WITH PINS



```
int led = 12;
int button = 1;

void setup() {
  pinMode(led, OUTPUT); // e.g. speakers, leds, motors
  pinMode(button, INPUT); // e.g. buttons, IR sensors, microphones, accelerometers
}

void loop() {
  digitalWrite(led, LOW);
  delay(1000);
  digitalWrite(led, HIGH);
  delay(1000);
}
```

pins in arduino

example code

Code Explanation :

```
void setup(){  
  
  // One Time Setup and Pin Initialisations  
  
}  
  
void loop(){  
  
  // Loops forever  
  
}
```

Code Explanation : Continued

```
int led = 2; // Define the Pin number
```

```
void setup(){
```

```
pinMode(led, OUTPUT ) ; // Set the Mode the Pin is going to function  
}
```

```
void loop(){
```

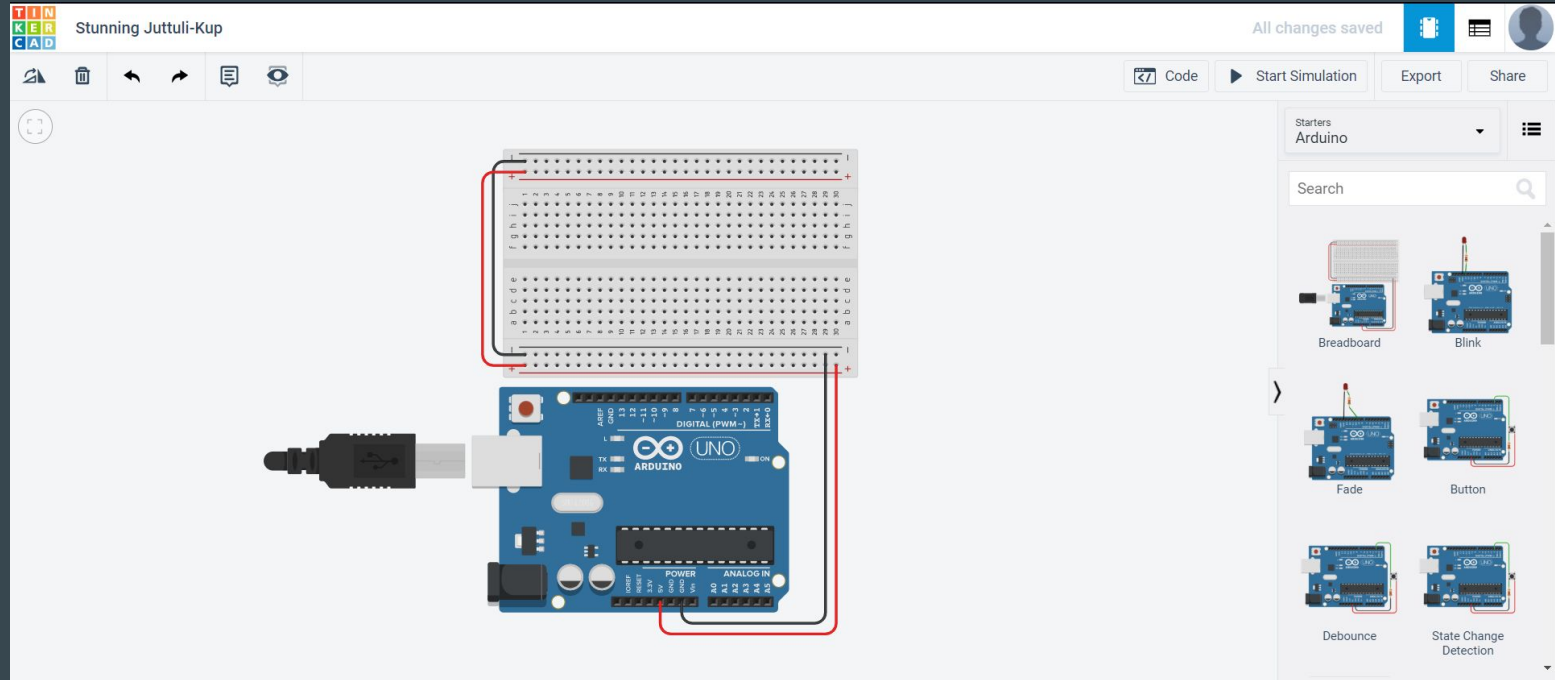
```
digitalWrite(2 , HIGH); // Set the Value of the Pin to HIGH ( 3.3V)
```

```
delay(200) ; // wait for 200 milliseconds
```

```
digitalWrite(2 , LOW); // Set the Value of Pin to LOW ( 0 V )
```

```
delay(200); // wait for 200 milliseconds ( Question : what happens if i delete this line ? )  
}
```


TinkerCAD Environment :



Next Steps :

- Go to tinkercad.com/circuits
- Launch Circuits
- Create your account
- Create a New Circuit

Problem - 1

Let's understand the Tinkercad Environment and solve the following problems :

1. Blink an In-build LED (Pin 13)
2. Blink an External LED
3. Attach an push button and Turn off and on LED based on the inputs
4. Attach a Potentiometer and get Input Values from it.
5. Rotate a Servo by giving some custom input values.
6. Rotate servo using inputs from potentiometer.
7. Attach a Photo resistor and use them to control the brightness of a led.

Blink In-build LED :

- Drag and Drop an Arduino from the Components List
- Click on Code and select Text
- Write the following code
- Click on Start Simulation to see the output

CODE :

```
void setup(){  
    pinMode(13, OUTPUT); // 13 is also the LED_BUILTIN  
}  
  
void loop(){  
    digitalWrite(13, HIGH);  
    delay(1000); // Wait for 1000 millisecond(s)  
    digitalWrite(13, LOW);  
    delay(1000); // Wait for 1000 millisecond(s)  
}
```

Blink an External LED :

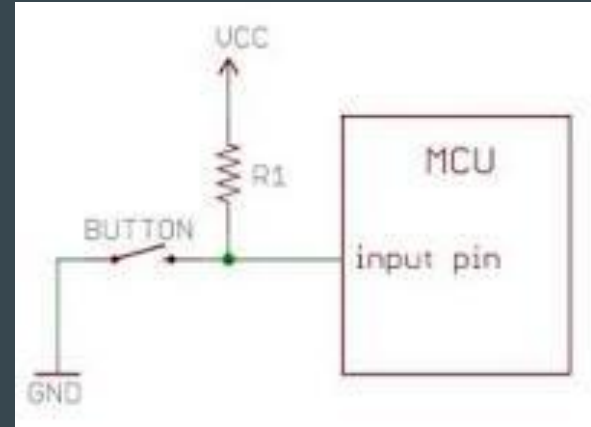
- Drag and Drop an Arduino from the Components List
- Drag and Drop a LED and Resistor and connect the LED to 13th Pin in the Arduino.
- Click on Code and select Text
- Write the following code
- Click on Start Simulation to see the output

CODE :

```
void setup(){  
    pinMode(13, OUTPUT); // 13 is also the LED_BUILTIN  
}  
  
void loop(){  
    digitalWrite(13, HIGH);  
    delay(1000); // Wait for 1000 millisecond(s)  
    digitalWrite(13, LOW);  
    delay(1000); // Wait for 1000 millisecond(s)  
}
```

Attach a Button :

- INPUT_PULLUP
 - Instead of Attaching an External Pull-up resistor , we will use the Pull-up resistor in the Arduino by Specifying it in the pinMode
 - `pinMode(button , INPUT_PULLUP);`
- digitalRead()
 - Reads the value from a specified digital pin, either HIGH or LOW.
 - Input parameters : Pin number
 - `val = digitalRead(button);`

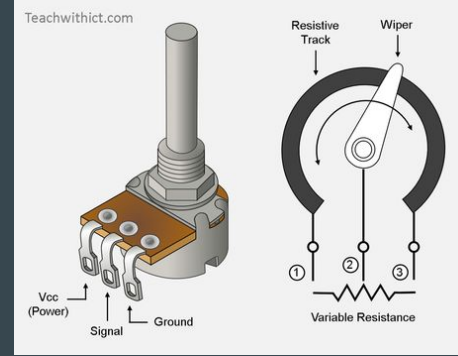


Attach a Potentiometer and get Input Values from it.

Serial Monitor : The Serial Monitor is part of the Arduino IDE software. Its job is to allow you to both send messages from your computer to an Arduino board (over USB) and also to receive messages from the Arduino.

`Serial.begin(baud_rate);` //Starts the Serial comms

`Serial.println(value);` // Prints Data



`analogRead()` : Reads the value from the specified analog pin. Arduino boards contain a multichannel, 10-bit analog to digital converter. This means that it will map input voltages between 0 and the operating voltage(5V or 3.3V) into integer values between 0 and 1023.

Rotate a Servo by giving some custom input values.

Servo motors are great devices that can turn to a specified position. Usually, they have a servo arm that can turn 180 degrees.

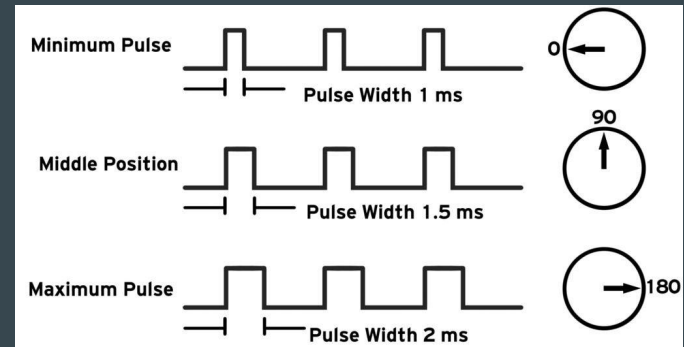
But Instead of sending Signals , we will using an Library for the same :

```
#include<Servo.h>
```

```
Servo servo_1; // Create a Servo Object
```

```
servo_1.attach(servo_pin); //Define Servo pin
```

```
servo_1.write(90); // Rotate to 90 degrees
```



Get Input from Serial Monitor

```
int pos = 0;

if (Serial.available()){           // Check if data from serial monitor is available

    pos = Serial.parseInt();       // Convert String to Integer

}
```

Rotate servo using inputs from potentiometer.

Try this Problem on your own.

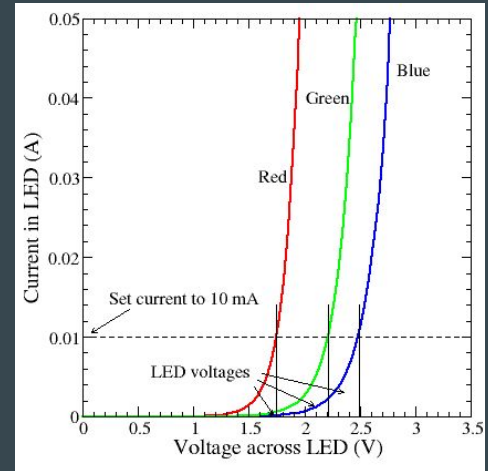
https://github.com/aswinkumar1999/Intro_to_Arduino

PWM - Brightness of the LED

We saw about Pulse width Modulation in Servo , let us see how that helps us in controlling brightness of an LED.

LED's are diodes , which implies they don't follow Ohm's law ($V = IR$).

We can control the brightness but by controlling the current , that implies having a variable resistor. Yes , Go ahead try to simulate if that works

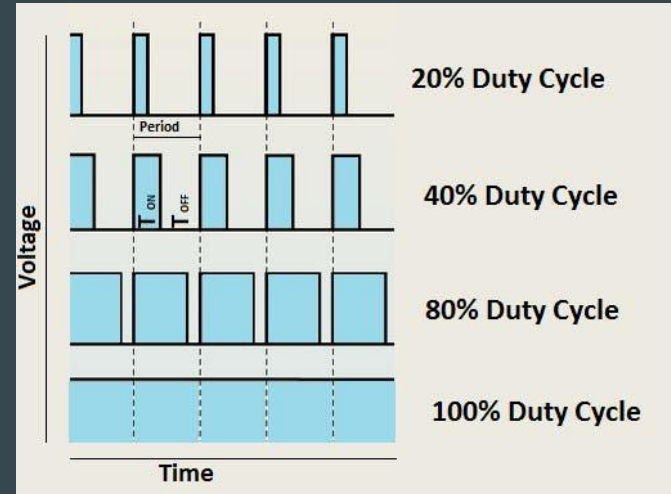


PWM - Continued

Hmm ... So you must have found out that you need to set the values of the Potentiometer appropriately, and not just that, it also has a drawback of power loss, so there's an alternate way of controlling the brightness, it is done by controlling the time which the pin is turned on.

PWM - Pulse Width Modulation.

Let's now see how to set the Duty Cycle and how it affects the brightness of the LED.



PWM - Continued

Remember first project (LED blinking) ?

CODE :

```
void setup(){
  pinMode(13, OUTPUT); // 13 is also the LED_BUILTIN
}

void loop(){
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

This is a PWM signal with 50 % Duty Cycle at a Frequency of 0.5Hz , if we increase the frequency to 500 Hz or above , you could clearly see how the brightness changes from the blink code.

Go to the Blink code and Have two LEDs side by side and compare how their brightness varies with different frequencies.

PWM - Alternate Method

```
analogWrite();
```

Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds.

```
analogWrite(ledPin, duty_cycle); // Duty Cycle between 0 - 255
```

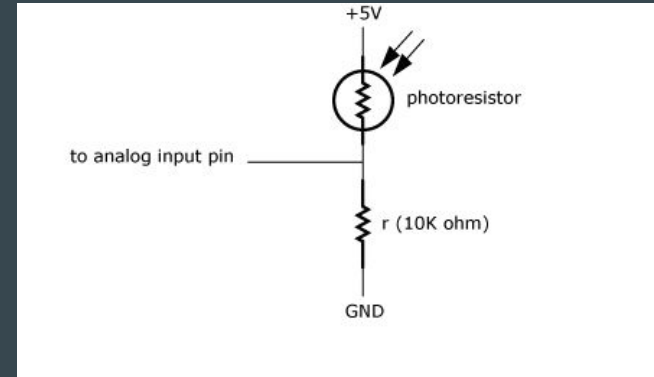
Control Brightness of LED using Photoresistor

Photoresistor is an active component that decreases resistance with respect to receiving luminosity (light) on the component's sensitive surface.

Using the Circuit Provided , Hookup the Potentiometer and Control the Brightness of the LED using the Pulse width signal estimating the Duty cycle from the Analog Input

Hint :) : You might need to google for the `map()` function.

`map(value, fromLow, fromHigh, toLow, toHigh)`



Problem - 2

The Air conditioner in your house is not functioning properly , you take a closer look at what happens and find out the PID controller of your Air Conditioner has been damaged and once you turn on the AC , it cools down your room to 16 degree celsius , but being a person from Chennai , you find it appropriate to sleep at 24 ± 1 degree celsius , so with your passion for electronics , you start designing a circuit which turns off the AC when the temperature falls below 23 degree celsius and turn it on when the temperature rises above 25 degree celsius.

Design a circuit for the same.

Problem Statement - 3

Assuming you are an Electronics Engineer in a reputed automobile company , now you are part of a team working on the automated parking system of a newly developing self-driving car , you are asked to design an system which would measure the clearance distance in the rear left and rear right of the car and report to the automated system to take further decisions, and upon the distance falling behind a certain threshold , you are supposed to blink an led which is connected to the Dashboard of the car to alert the passengers that the automated system is failing and manual control needs to be taken.

Design an Arduino circuit for the same.