IoT-Workshop

ETCE Lab

February 2024

1 Welcoming

Welcome to our Internet of Things (IoT) Workshop

2 Introduction

Throughout this workshop,we'll embark on a journey through the fundamentals of IoT, starting with an overview of what IoT is and how it's being use in different industries especially in mining.

3 what you will learn

1. Experience with popular microcontrollers such as the Arduino Nano 33 and Arduino Uno
2. Being able to work with various sensors
3. Foundational coding skills, including essential functions and techniques, necessary for IoT projects.

4 let's start

Attention

Please remember this for all experiments before you look at the first one.

1. Treat the equipment very gently.
2. Look at the diagram first to see how the wires and sensors connect to the Arduino.
3. Always ask questions.
4. We will begin coding after double-checking that the wires and sensors are correctly connected to the Arduino.
5. Try to learn the codes and keep questioning the tutor.
6. When you completely understand the codes, start typing them in the Arduino-related software.
7. Look optimistically. If you did not have any errors.
8. Then connect the Arduino hardware to the computer with the related cable.
9. Finally, you upload the codes to Arduino.

1) Detecting fire

we will learn how to detect fire using a flame sensor.

Components and supplies:

1. Flame Sensor (model with an analog out)

   These types of sensors are used for short range fire detection(up to about 3 feet)

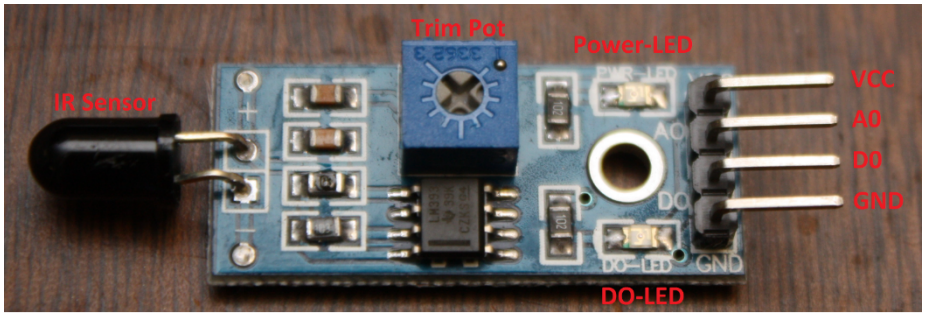
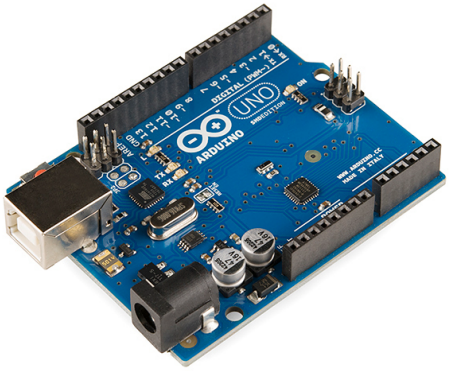


Figure 1: flame sensor

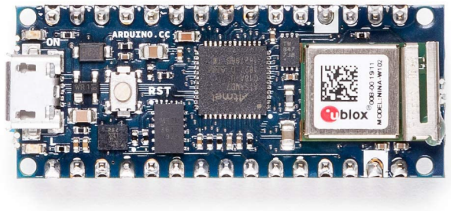
2. Male to Female jumper wires

Figure 2: Jumper wires

3. An Arduino, any flavor



Arduino UNO



Arduino Nano 33

4. Lighter or another flame source for testing

Figure 3: Lighter

NOTE : If you are using Arduino nano 33, then you need to use " Bread Board " also.

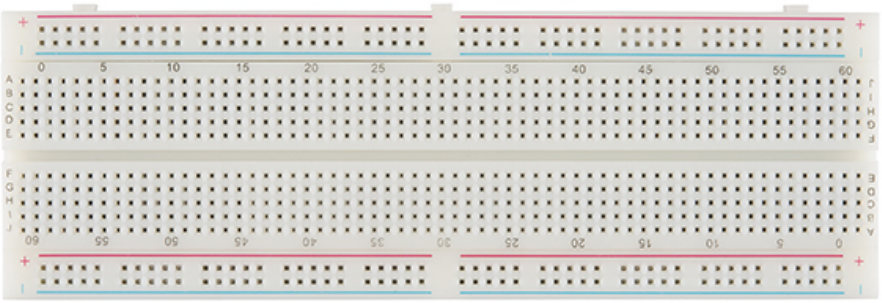


Figure 4: Bread Board

Important pins in EVERY Arduino

- VCC→Positive voltage input: 5v for analog 3.3v for Digital
- A0→ Analog output
- D0→ Digital output
- GND→Ground

Wiring to an Arduino:

To wire the Flame Sensor to the Arduino simply connect the following as shown:

- Flame sensor→ Arduino
- VCC → 5V
- GND → GND
- A0 → Analog in 0

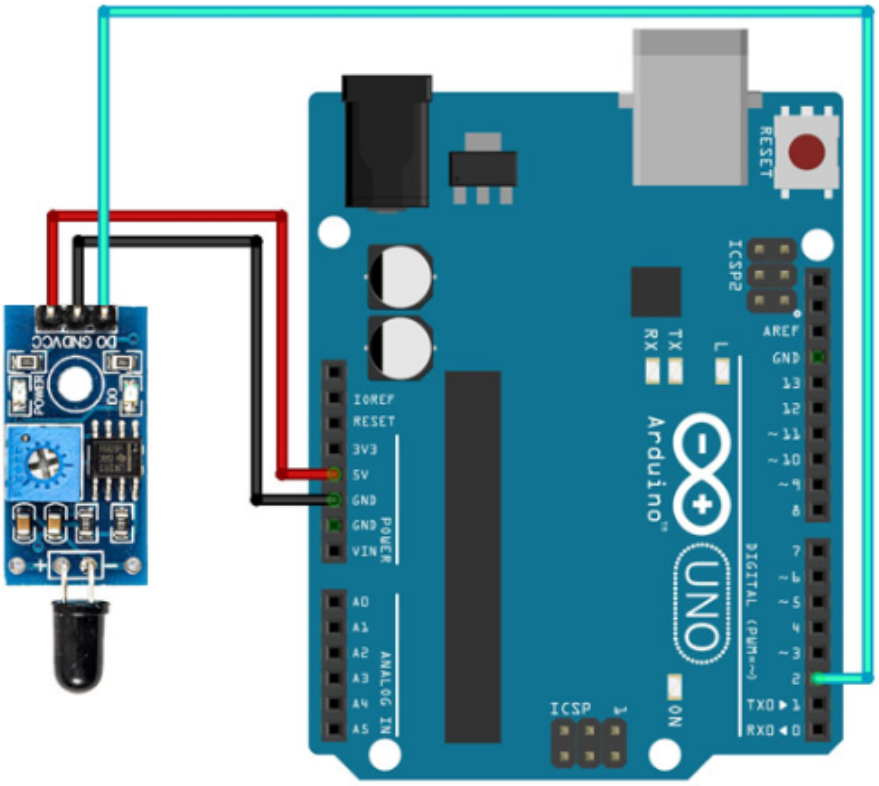


Figure 5: Schematic of flame detector

Code:

```
// lowest and highest sensor readings:
const int sensorMin = 0;     //  sensor minimum
const int sensorMax = 1024;  // sensor maximum

void setup()  {
  // initialize serial communication @ 9600 baud:
  Serial.begin(9600);
}
void loop() {
  // read the sensor on analog A0:
        int sensorReading  = analogRead(A0);
  // map the sensor range (four options):
  // ex: 'long  int map(long int, long int,
  //long int, long int, long int)'
        int range=map(sensorReading,sensorMin,sensorMax,0,3);
```

```
  // range value:
  switch (range) {
  case 0:    // A fire closer than 1.5 feet away.
    Serial.println("** Close  Fire **");
    break;
  case 1:    // A fire between 1-3 feet away.
    Serial.println("**  Distant Fire **");
    break;
  case 2:    // No fire detected.
    Serial.println("No  Fire");
    break;
  }
  delay(1);  // delay between reads
}
```

Congratulation!!

2) Smoke Detector

we will learn how to create a smoke detector using gas sensor.

Components and supplies:

1. Resistor 220 ohm

Figure 6: Resistor 220 ohm

2. Gas Sensor

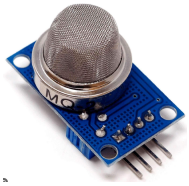


Figure 7: Gas sensor

3. An Arduino, any flavor

4. Buzzer

   A buzzer is an electronic device that emits sound when it is powered by a 5-volt electrical current. These buzzers are commonly used in alarms, timers, confirmation of user input.

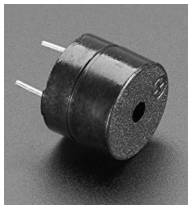


Figure 8: Buzzer

5. Bread Board

6. Jumper wires

7. LED Lights

Green LED

Red LED

Wiring to an Arduino:

To wire the smoke sensor to the Arduino simply connect the following as shown:

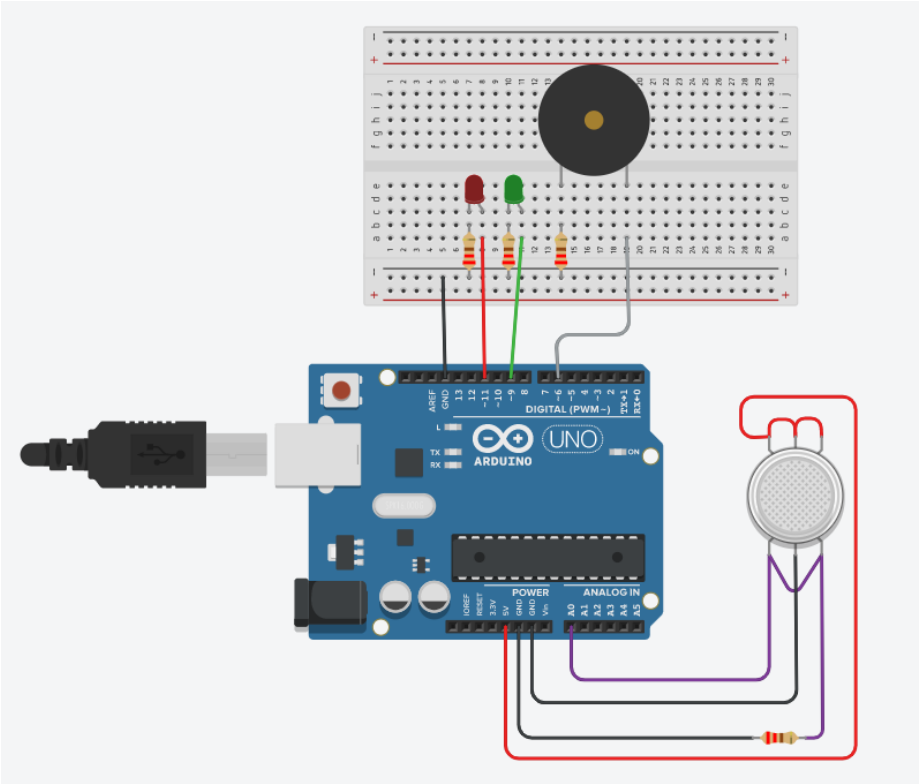


Figure 9: Schematic of Smoke Detector

Code:

```
int red_LED_PIN = 11;
int green_LED_PIN = 9;
int blue_LED_PIN =  10;
int buzzer = 6;
int smoke_detector = A0;
int safety_lim = 60; //Sets  smoke density safe limit

void setup() {
  pinMode(red_LED_PIN, OUTPUT);
  pinMode(green_LED_PIN, OUTPUT);
  pinMode(blue_LED_PIN, OUTPUT);
  pinMode(buzzer,  OUTPUT);
  pinMode(smoke_detector, INPUT);
  Serial.begin(9600); //baud rate
}

void  loop() {
```

```
  int sensor_read = analogRead(smoke_detector);
        //reads and  stores the reading from the detector in sensor_read

  Serial.print("Smoke  Density: ");
  Serial.println(sensor_read);

  if (sensor_read > safety_lim)
    // Checks if reading is beyond safety limit
  {
        analogWrite(red_LED_PIN,255);
    analogWrite(green_LED_PIN, 0);
    tone(buzzer,500, 100); //piezo rings
  }
  else
  {
    analogWrite(green_LED_PIN, 255);
    analogWrite(red_LED_PIN,0);
    noTone(buzzer); //peizo wont ring
  }
  delay(50);
}
```

Congratulation!!

3) Vibration Detection

We will learn to be alert by LED lights if there is a vibration.

Components and supplies:

1. Vibration sensor

Figure 10: Vibration sensor

2. Arduino
3. Green and Red LED lights
4. Jumper wires

Wiring to an Arduino:

To wire the vibration sensor to the Arduino simply connect the following as shown:

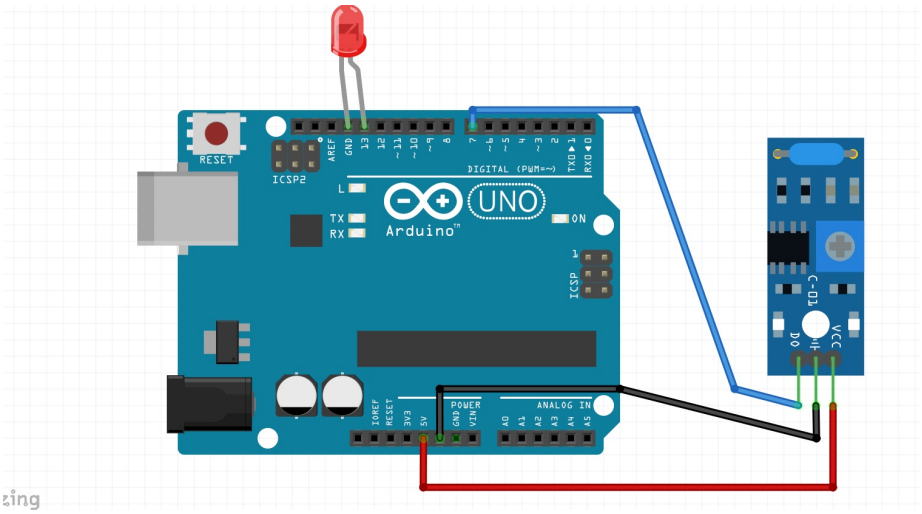


Figure 11: Schematic of vibration Detector

Code:

```
int vib_pin=7;
int led_pin=13;
void setup() {
  pinMode(vib_pin,INPUT);
  pinMode(led_pin,OUTPUT);
}

void loop() {
  int val;
  val=digitalRead(vib_pin);
  if(val==1)
  {
    digitalWrite(led_pin,HIGH);
    delay(1000);
    digitalWrite(led_pin,LOW);
    delay(1000);
   }
   else
   digitalWrite(led_pin,LOW);
}
```

Congratulation!!

★★★

4) Make a siren

We will learn how to use button controlled siren with different LED transitions.

Components and supplies:

1. Push button switch 12mm

Figure 12: Push button

2. 9V battery (generic)

Figure 13: 9V battery

3. Arduino
4. Buzzer
5. Resistor 100 ohm
6. Resistor 221 ohm
7. Resistor 10k ohm)
8. Jumper wires

9. Bread Board
10. Green and Red LED lights

Wiring to an Arduino:

To wire the LED light and resistors to the Arduino simply connect the following as shown:

- negative terminal of all LED's → 220 ohm resistor
- positive terminal of all red LED's → pin 3 to pin 7
- positive terminal of all blue LED's → pin 8 to pin 12
- negative rail of buzzer → 100 ohm resistor
- positive rail of buzzer → pin 13
- one out of four pins of pushButton → pin 2
- down of pushButton → GND rail using a pull down resistor of 10k ohm
- last button pin → 5v

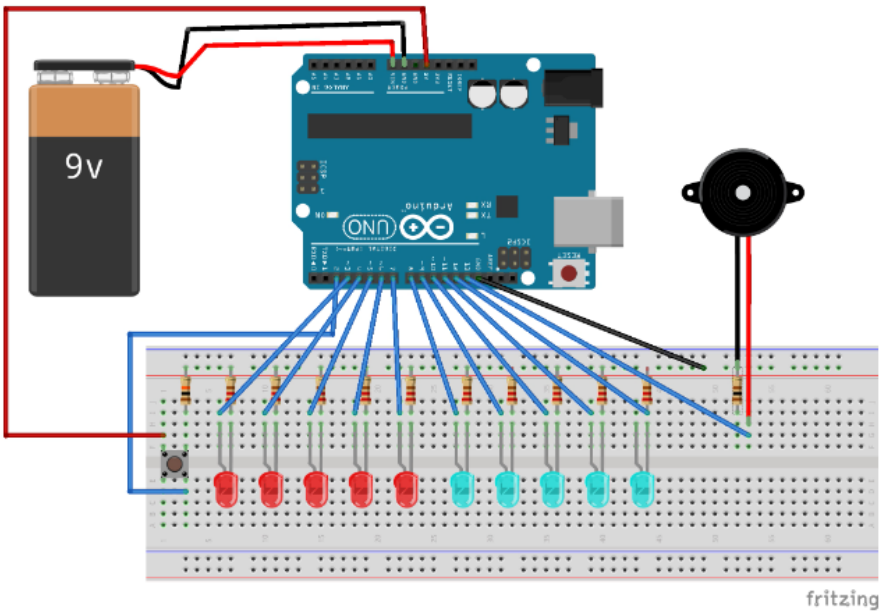


Figure 14: Schematic of making siren

Code:

```
boolean lastbutton=LOW;
boolean currentbutton=LOW;
int input=0;
int buzz=13; // Buzzer Pin
int j=3;
int k=12;
void setup() {
for(int i=3;i<=12;i++)
pinMode(i,OUTPUT);
pinMode(2,INPUT);
}
boolean debounce(boolean last){
//Function to solve the problem of button debouncing
  boolean current=digitalRead(2);
  if(last!=current)
  {
    delay(5);
    current=digitalRead(2);
    }
  return current;
  }
void loop() {
  for(int i=3;i<=12;i++)
  digitalWrite(i,LOW);
  currentbutton=debounce(lastbutton);
  if(lastbutton==LOW && currentbutton==HIGH)
  {
  input++;
  }
  lastbutton=currentbutton;
  settone(input);
}

void settone(int input)
{

if(input==1)
one();
else if(input==2)
oneA();
else if(input==3)
two();
else if(input==4)
twoA();
else if(input==5)
three();
else if(input==6)
```

```
threeA();
else if(input==7)
four();
}

void one() {
   //This function produces the 1st siren sound with ON/OFF led.
  // Whoop up
  for(int hz = 440; hz < 1000; hz+=25){
    tone(buzz, hz, 50);
    delay(5);
    for(int i=3;i<=7;i++)
    digitalWrite(i,HIGH);
  }
  // Whoop down
  for(int hz = 1000; hz > 440; hz-=25){
    tone(buzz, hz, 50);
    delay(5);
    for(int i=3;i<=7;i++)
    {
      digitalWrite(i,LOW);
      digitalWrite(i+5,HIGH);
      }
  }
  }

void oneA() {
   //This function produces differnt transition on 1st siren.

  // Whoop up
  for(int hz = 440; hz < 1000; hz+=25){
    tone(buzz, hz, 50);
    delay(5);
  }
  if(j>=3){
    digitalWrite(j,HIGH);
    j=j+1;
    digitalWrite(k,HIGH);
    k=k-1;
    if(j==8)
    j=3;
    if(k==7)
    k=12;
    }
  // Whoop down
  for(int hz = 1000; hz > 440; hz-=25){
    tone(buzz, hz, 50);
    delay(5);
  }
  }
```

```
void two() {
  //This function produces the 2nd siren sound with progressive led.
  // Whoop up
  for(int hz = 440; hz < 1000; hz+=25){
    tone(buzz, hz, 50);
    delay(5);
  }
     loopF(3,12,20);
     loopR(12,3,20);
  // Whoop down
  for(int hz = 1000; hz > 440; hz-=25){
    tone(buzz, hz, 50);
    delay(5);
  }
  }
void twoA() {
   //This function produces differnt transition on 2nd siren.
  // Whoop up
  for(int hz = 440; hz < 1000; hz+=25){
    tone(buzz, hz, 50);
    delay(5);
  }
     loopF(3,k,20);
     loopR(12,j,20);
     k--;
     if(k==3)
     k=12;
     j++;
     if(j==12)
     j=3;
  // Whoop down
  for(int hz = 1000; hz > 440; hz-=25){
    tone(buzz, hz, 50);
    delay(5);
  }
  }
  void three() {
  //This function produces the 3rd siren(AMBULANCE) sound with led.
  tone(buzz,440,200);
    delay(300);
  for(int i=3;i<=6;i++)
  digitalWrite(i,HIGH);
  noTone(buzz);
  tone(buzz,494,500);
  delay(300);
  for(int i=3;i<=6;i++){
  digitalWrite(i,LOW);
  digitalWrite(i+6,HIGH);
  }
  noTone(buzz);
```

```
  tone(buzz,523,300);
   delay(200);
  digitalWrite(7,HIGH);
  delay(50);
  digitalWrite(8,HIGH);
  delay(50);
  noTone(buzz);
}
  void threeA() {
  //This function produces differnt transition on 3rd siren.
  tone(buzz,440,200);
    delay(100);
    loopF(5,10,20);
    loopR(10,5,20);
  noTone(buzz);
  for(int i=3;i<=4;i++){
  digitalWrite(i,HIGH);
  digitalWrite(i+8,HIGH);}
  tone(buzz,494,500);
  delay(300);
  noTone(buzz);
  for(int i=3;i<=4;i++){
  digitalWrite(i,LOW);
  digitalWrite(i+8,LOW);
  }
  tone(buzz,523,300);
   delay(300);
  noTone(buzz);
}
void four() {
//This function produces the 4th siren(POLICE) sound with led.
for(int i=3;i<=11;i+=2)
digitalWrite(i,HIGH);
  for(int hz = 440; hz < 1000; hz++){
    tone(buzz, hz, 50);
    delay(5);
  }
for(int i=3;i<=11;i+=2)
digitalWrite(i,LOW);
 for(int i=4;i<=12;i+=2)
digitalWrite(i,HIGH);
  for(int hz = 1000; hz > 440; hz--){
    tone(buzz, hz, 50);
    delay(5);
    }
  }
  // SOME EXTRA FUNCTIONS OTHER THAN THE SIREN TONES
    void loopF(int spin,int epin,int dela){
 //loopF can blink the led in forward direction so spin must
 //be lower than epin.
```

```
  for(int i=spin;i<=epin;i++){
  digitalWrite(i,HIGH);
  delay(dela);
  low();
  if(spin==epin){
  spin=3;
  epin=12;}
  }}
void loopR(int epin,int spin,int dela){
//loopR can blink the led in reverse/backward direction
//so epin must be lower than spin.
  for(int i=epin;i>=spin;i--){
  digitalWrite(i,HIGH);
  delay(dela);
  low();
  if(spin==epin){
  spin=3;
  epin=12;}
  }}
void low(){             //Used by loopF and loopR
 for(int i=3;i<=12;i++)
digitalWrite(i,LOW);
  }
```

Congratulation!!

5) I2C Liquid Crystal Displays using Arduino

we will learn how to work with LCD and printing simple texts using Arduino.

Components and supplies:

1. Arduino UNO
2. Male/Female Jumper Wires
3. I2C 16x2 Arduino LCD Display Module

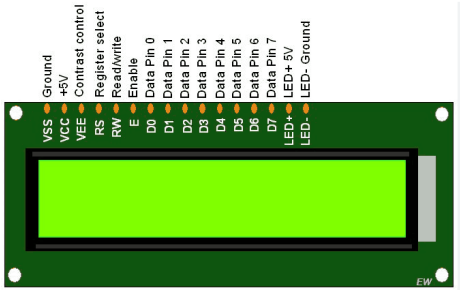


Figure 15: LCD Display Modul

Wiring to an Arduino:
Please follow the diagram

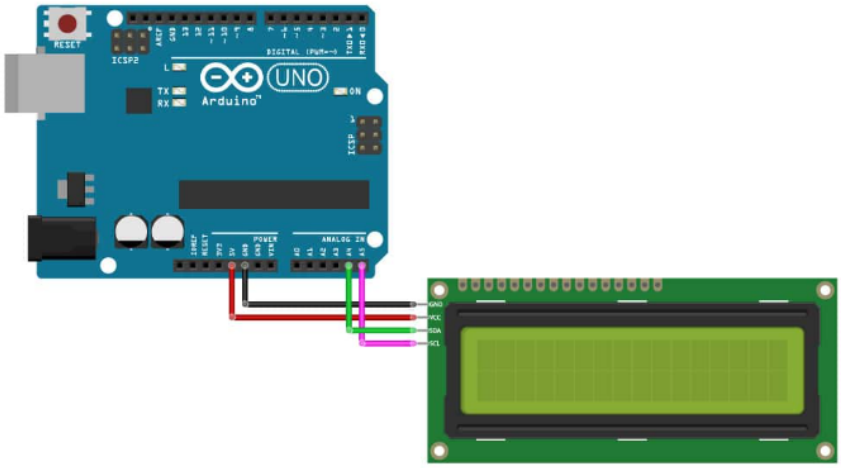


Figure 16: Diagram

Code:

```
#include LiquidCrystal_I2C.h

#include  Wire.h

//initialize the liquid crystal library
//the first parameter is  the I2C address
//the second parameter is how many rows are on your screen
//the  third parameter is how many columns are on your screen
LiquidCrystal_I2C lcd(0x27,  16, 2);

void setup() {

  //initialize lcd screen
  lcd.init();
  // turn on the backlight
  lcd.backlight();
}
void loop() {
  //wait  for a second
  delay(1000)
  // tell the screen to write on the top row
  lcd.setCursor(0,0);
  // tell the screen to write   hello  , f r o m  on the top  row
  lcd.print(  Hello  , F r o m );
  // tell the screen to write on the bottom  row
  lcd.setCursor(0,1);
  // tell the screen to write     Arduino_uno_guy      on the bottom row
  // you can change whats in the quotes to be what you want
it to be!
  lcd.print(    Arduino_uno_guy    );

}
```

Congratulation!!

★★★★★

6) Blinking LED using Arduino and Push button switch

we will learn how to turn on and turn off the LED light using Push button switch.

Components and supplies:

1. Arduino UNO
2. Pushbutton switch

Figure 17: Pushbutton switch

3. Resistor 10k ohm
4. Breadboard
5. Jumper wires
6. LED Lights

Wiring to an Arduino:
Please follow the diagram

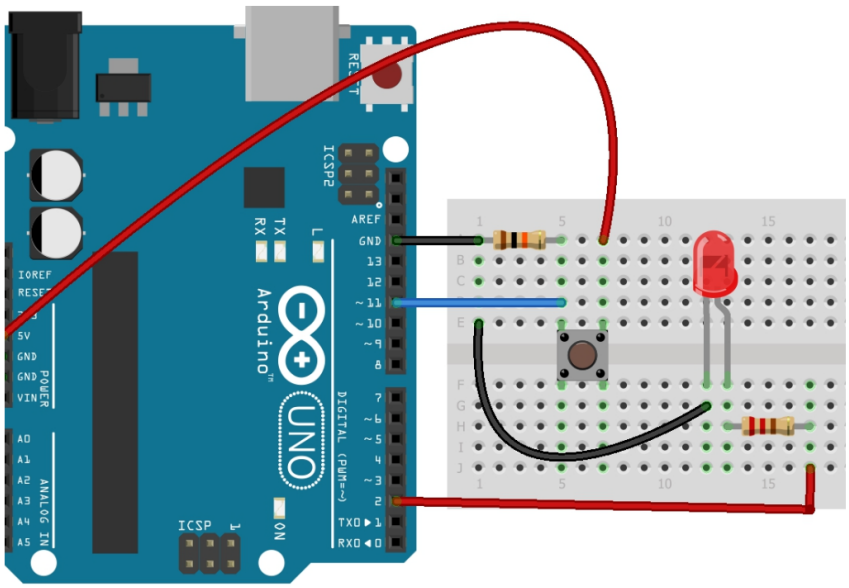


Figure 18: Diagram

Code:

```
int led1 = 2;
int led2 = 3;
int led3 = 5;
int button = 6;

void  setup() {
  pinMode(button,INPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
    }

void loop() {
int ButtonStatus = digitalRead(button);
if  (ButtonStatus == HIGH)
{
  digitalWrite(led1,HIGH);
  digitalWrite(led2,HIGH);
  digitalWrite(led3,HIGH);
} else {
  digitalWrite(led1,LOW);
  digitalWrite(led2,LOW);
  digitalWrite(led3,LOW);
}
    }
```

Congratulation!! you got the fifth star :)

7)Car Parking Simulator

we will experience simulation of a real life car parking using Arduino, Ultrasonic sensors, LCD Display and RGB LEDs.

Components and supplies:

1. Arduino UNO
2. LCD Display
3. Resistor 221 ohm
4. Breadboard
5. Jumper wires (generic)
6. Male/Female Jumper Wires
7. USB-A to B Cable

Figure 19: USB-A to B Cable

8. Ultrasonic Sensor - HC-SR04 (Generic)

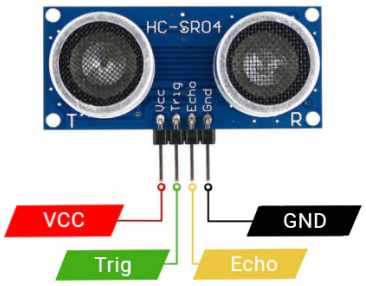


Figure 20: Ultrasonic Sensor - HC-SR04 (Generic)

9. RGB Diffused Common Cathode

Figure 21: RGB Diffused Common Cathode

Wiring to an Arduino:
Please follow the diagram

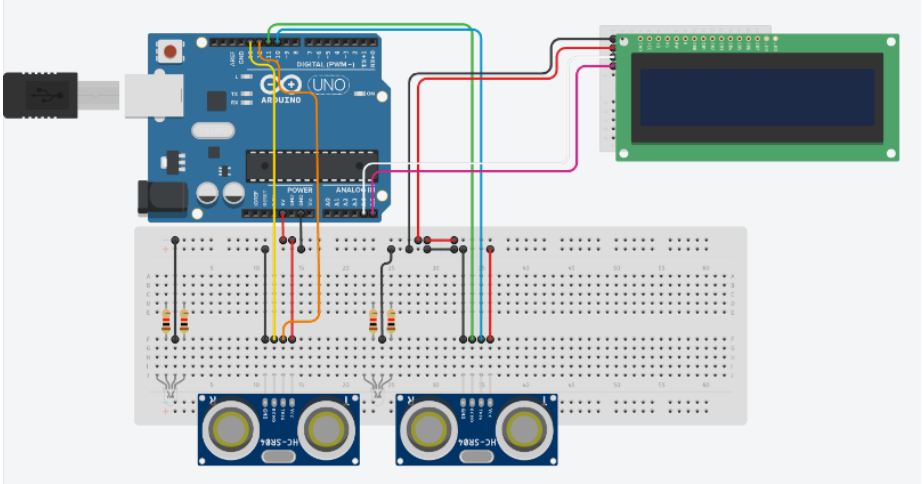


Figure 22: Car Parking Simulator Diagram

Code:

```
#include <Ultrasonic.h> //Library for the ultrasonic sensor
#include<LiquidCrystal_I2C.h>  //Library for the display
#include<Wire.h>
LiquidCrystal_I2C lcd (0x27, 16,  2); //Creates an object lcd
Ultrasonic ultrasonic1 (13, 12); //Creates an object  ultrasonic1
Ultrasonic ultrasonic2 (11, 10); //Creates an object ultrasonic1
int  ledOccupied1 = 9; //LED to indicate if the spot is occupied
int ledFree1 = 8;  //LED to indicate if the spot is free
int ledOccupied2 = 7; //LED to indicate  if the spot is occupied
int ledFree2 = 6; //LED to indicate if the spot is free
int  val1; //val1 as a value variable for ultrasonic sensor
int val2; //val1 as a  value variable for ultrasonic sensor
byte SpaceA[] = //Custom character to indicate  a free spot
{
  0b11111,
  0b10001,
  0b10011,
  0b10111,
  0b11111,
  0b10011,
  0b10011,
  0b10011
};
byte SpaceB[] = //Custom character  to indicate a free spot
{
  0b11111,
  0b10001,
  0b11001,
  0b11101,
  0b11111,
  0b11001,
  0b11001,
  0b11001
};
byte SpaceC[] = //Custom  character to indicate a free spot
{
  0b10011,
  0b10011,
  0b10011,
  0b10011,
  0b10000,
  0b10000,
  0b10000,
  0b11111
};
byte  SpaceD[] = //Custom character to indicate a free spot
{
  0b11001,
```

```
  0b11001,
  0b11001,
  0b11001,
  0b00001,
  0b00001,
  0b00001,
  0b11111
};
byte  NoSpaceA[] = //Custom character to indicate a occupied spot
{
  0b11111,
  0b10000,
  0b10000,
  0b11000,
  0b11100,
  0b11110,
  0b10111,
  0b10011
};
byte NoSpaceB[] = //Custom character to indicate a occupied spot
{
  0b11111,
  0b00001,
  0b00001,
  0b00011,
  0b00111,
  0b01111,
  0b11101,
  0b11001
};
byte NoSpaceC[] = { //Custom character to indicate  a occupied spot
  0b10011,
  0b10111,
  0b11110,
  0b11100,
  0b11000,
  0b10000,
  0b10000,
  0b11111
};
byte NoSpaceD[] = { //Custom character  to indicate a occupied spot
  0b11001,
  0b11101,
  0b01111,
  0b00111,
  0b00011,
  0b00001,
  0b00001,
  0b11111
};
```

```
void setup() {
  lcd.init(); //Starts the display
  lcd.backlight(); //Turns on the display  backlight
  //Create the custom characters
  lcd.createChar(1, SpaceA);
  lcd.createChar(2, SpaceB);
  lcd.createChar(3, SpaceC);
  lcd.createChar(4,  SpaceD);
  lcd.createChar(5, NoSpaceA);
  lcd.createChar(6, NoSpaceB);
  lcd.createChar(7, NoSpaceC);
  lcd.createChar(8, NoSpaceD);
  //Define all  LEDs as Output
  pinMode(ledOccupied1, OUTPUT);
  pinMode(ledFree1, OUTPUT);
  pinMode(ledOccupied2, OUTPUT);
  pinMode(ledFree2, OUTPUT);
}

void  loop() {
  //Value variables will store all informations from the sensors
  val1 = ultrasonic1.Ranging(CM);
  val2 = ultrasonic2.Ranging(CM);
  //If  the first spot is occupied
  if ((val1 < 10) && (val2 > 10)) {
    digitalWrite(ledOccupied1,  HIGH);
    digitalWrite(ledFree1, LOW);
    digitalWrite(ledOccupied2, LOW);
    digitalWrite(ledFree2, HIGH);
    lcd.clear();
    lcd.write(5);
    lcd.setCursor(1,  0);
    lcd.write(6);
    lcd.setCursor(0, 1);
    lcd.write(7);
    lcd.setCursor(1,  1);
    lcd.write(8);
    lcd.setCursor(7, 0);
    lcd.write(1);
    lcd.setCursor(8,  0);
    lcd.write(2);
    lcd.setCursor(7, 1);
    lcd.write(3);
    lcd.setCursor(8,  1);
    lcd.write(4);
    delay(500);
  }
  //If the second spot is  occupied
  else if ((val1 > 10) && (val2 < 10)) {
    digitalWrite(ledOccupied1,  LOW);
```

```
  digitalWrite(ledFree1, HIGH);
  digitalWrite(ledOccupied2, HIGH);
  digitalWrite(ledFree2, LOW);
  lcd.clear();
  lcd.write(1);
  lcd.setCursor(1,  0);
  lcd.write(2);
  lcd.setCursor(0, 1);
  lcd.write(3);
  lcd.setCursor(1,  1);
  lcd.write(4);
  lcd.setCursor(7, 0);
  lcd.write(5);
  lcd.setCursor(8,  0);
  lcd.write(6);
  lcd.setCursor(7, 1);
  lcd.write(7);
  lcd.setCursor(8,  1);
  lcd.write(8);
  delay(500);
}
//If the first and second  spots are free
else if ((val1 > 10) && (val2 > 10)) {
  digitalWrite(ledOccupied1,  LOW);
  digitalWrite(ledFree1, HIGH);
  digitalWrite(ledOccupied2, LOW);
  digitalWrite(ledFree2, HIGH);
  lcd.clear();
  lcd.write(1);
  lcd.setCursor(1,  0);
  lcd.write(2);
  lcd.setCursor(0, 1);
  lcd.write(3);
  lcd.setCursor(1,  1);
  lcd.write(4);
  lcd.setCursor(7, 0);
  lcd.write(1);
  lcd.setCursor(8,  0);
  lcd.write(2);
  lcd.setCursor(7, 1);
  lcd.write(3);
  lcd.setCursor(8,  1);
  lcd.write(4);
  delay(500);
}
//If the first and second  spots are occupied
else if ((val1 < 10) && (val2 < 10)) {
  digitalWrite(ledOccupied1,  HIGH);
  digitalWrite(ledFree1, LOW);
  digitalWrite(ledOccupied2, HIGH);
```

```
    digitalWrite(ledFree2, LOW);
    lcd.clear();
    lcd.write(5);
    lcd.setCursor(1,  0);
    lcd.write(6);
    lcd.setCursor(0, 1);
    lcd.write(7);
    lcd.setCursor(1,  1);
    lcd.write(8);
    lcd.setCursor(7, 0);
    lcd.write(5);
    lcd.setCursor(8,  0);
    lcd.write(6);
    lcd.setCursor(7, 1);
    lcd.write(7);
    lcd.setCursor(8,  1);
    lcd.write(8);
    delay(500);
  }
  else {
    lcd.clear();
    digitalWrite(ledOccupied1, LOW);
    digitalWrite(ledFree1, LOW);
    digitalWrite(ledOccupied2,  LOW);
    digitalWrite(ledFree2, LOW);
  }
}
```

Congratulation!!

★★★★★★★

8) Traffic Light using Arduino

we will learn how to make the traffic light using Arduino.

Components and supplies:

1. Arduino UNO
2. Jumper wires (generic)
3. Breadboard
4. 5 mm LED Lights (Green, Red, Yellow)

Wiring to an Arduino:
Please follow the diagram

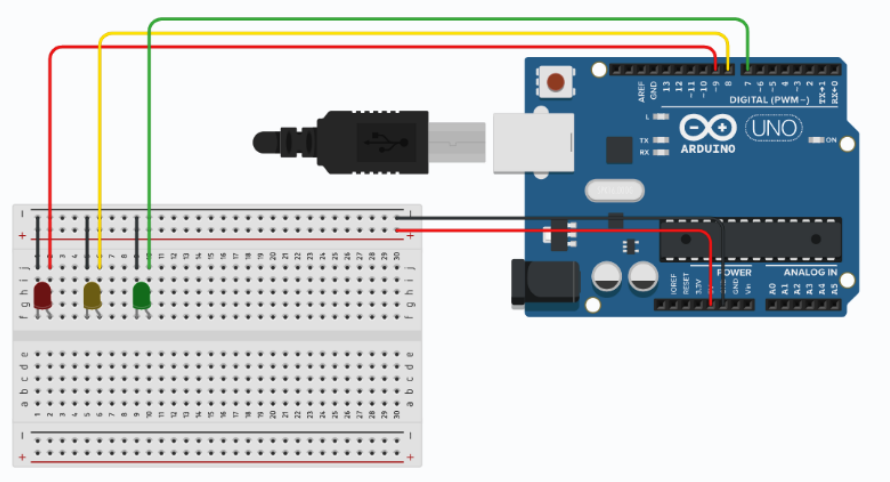


Figure 23: Traffic Light Diagram

Code:

```
int red = 9;
int yellow = 8;
int green = 7;

void setup(){

  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green,  OUTPUT);

}
void loop(){
digitalWrite(red, HIGH);
 delay(15000);
digitalWrite(red,  LOW);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);

digitalWrite(green, HIGH);
delay(20000);
digitalWrite(green,  LOW);
//
digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
```

```
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow,  LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);

  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);


}
```

Congratulation!!

★★★★★★★★

9) Making Ultrasonic Radar with Arduino and Micro-servo motor

we will learn how to make a small Ultrasonic Radar.

Components and supplies:

1. Arduino UNO
2. Jumper wires (generic)
3. Breadboard
4. 5Micro-servo motor

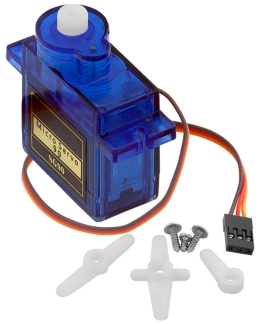


Figure 24: 5Micro-servo motor

5. Ultrasonic Sensor - HC-SR04 (Generic)

Wiring to an Arduino:
Please follow the diagram

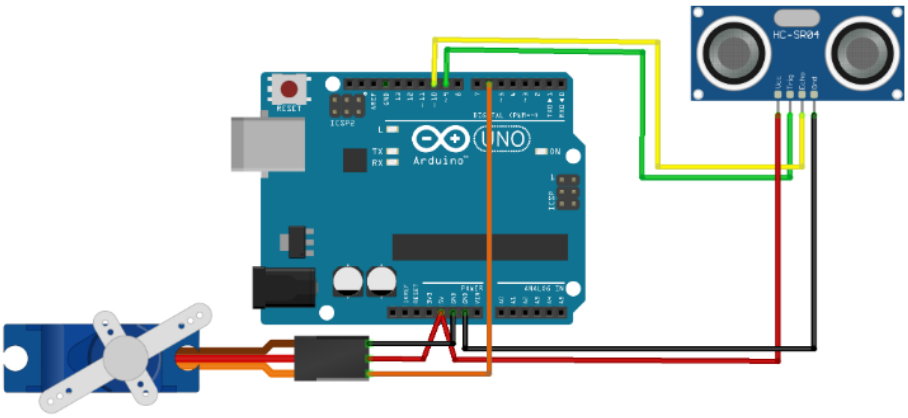


Figure 25: Ultrasonic Radar Diagram

Code:

```
#include<Servo.h>

#define trigPin 8
#define echoPin 9

long duration;
int distance ;

Servo myservo;

int calculateDistance()
{
  digitalWrite(trigPin,LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin,LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration*0.034/2;
  return distance;
}

void setup()
{
  pinMode(trigPin , OUTPUT);
  pinMode(echoPin, INPUT);
  myservo.attach(11);
  Serial.begin(9600);
}

void loop()
{
 int i ;
 for (i=15; i<=165; i++)
 {
  myservo.write(i);
  delay(15);
  calculateDistance();
  Serial.print(i);
  Serial.print(",");
  Serial.print(distance);
  Serial.print(".");
 }
 for(i=165; i>=15; i--)
 {
  myservo.write(i);
  delay(15);
  calculateDistance();
```

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

 }
}
```

Congratulation!!

10) Color Detection Using TCS3200/230 sensor

We will learn how the color detection sensor works with Arduino in terms of detecting different colors.

Components and supplies:

1. Arduino
2. TCS3200/TCS230

Figure 26: TCS3200/TCS230 sensor

3. RGB Diffused Common Cathode
4. Jumper wires

Wiring to an Arduino:
Please follow the diagram

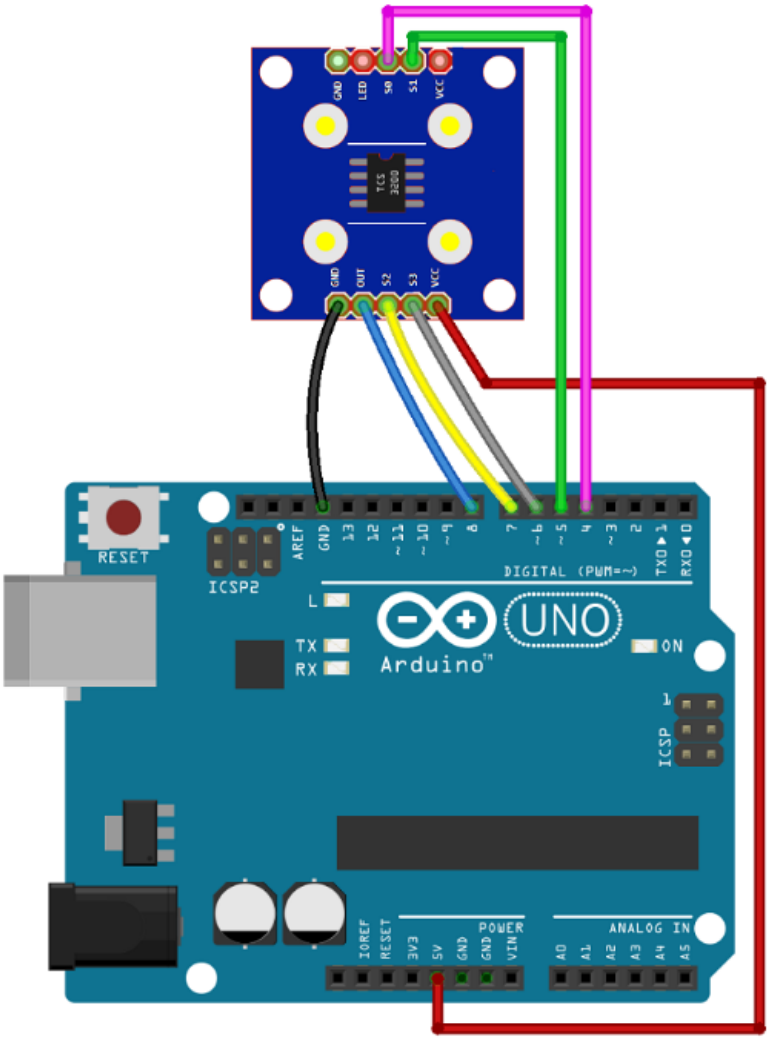


Figure 27: Schematic of Color Detection Sensor with Arduino

Code:

```
#define s0 8
#define s1 9
#define s2 10
#define s3 11
#define out 12

int data=0;

void setup()
{
   pinMode(s0,OUTPUT);     s
   pinMode(s1,OUTPUT);
   pinMode(s2,OUTPUT);
   pinMode(s3,OUTPUT);
   pinMode(out,INPUT);

   Serial.begin(9600);

   digitalWrite(s0,HIGH);
   digitalWrite(s1,HIGH);

}

void loop()
{

   digitalWrite(s2,LOW);
   digitalWrite(s3,LOW);
   Serial.print("Red value= ");
   GetData();

   digitalWrite(s2,LOW);
   digitalWrite(s3,HIGH);
   Serial.print("Blue value= ");
   GetData();

   digitalWrite(s2,HIGH);
   digitalWrite(s3,HIGH);
   Serial.print("Green value= ");
   GetData();

   Serial.println();

   delay(2000);
}

void GetData(){
```

```
    data=pulseIn(out,LOW);
    Serial.print(data);
    Serial.print("\t");
    delay(20);
}
```

Congratulation!!

★★★★★★★★★