

Engineering Explorations

IOT Workshop Using Arduino UNO

Prof. CVS N
Dr. Vidya M J

Topics Covered

- Blinking an on board LED
- Connecting external LED and blinking it.
- Blinking an LED using Breadboard
- LED FADEIN/FADEOUT
- Controlling LED using PUSH button
- Traffic Light Controller
- Ultrasonic Sensor
- Controlling LED using ultrasonic sensor
- Buzzer program
- Program with ultrasonic sensor, led and buzzer (intrusion detection)
- Controlling multiple LED's with ultrasonic sensor
- Controlling Buzzer using push button
- PIR sensor

Blinking an On-Board LED

```
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  
}
```

Connecting External LED and Blink

```
int led1=13;

void setup() {

    // initialize digital pin LED_BUILTIN as an output.
    pinMode(13, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

    digitalWrite(led1, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000);              // wait for a second

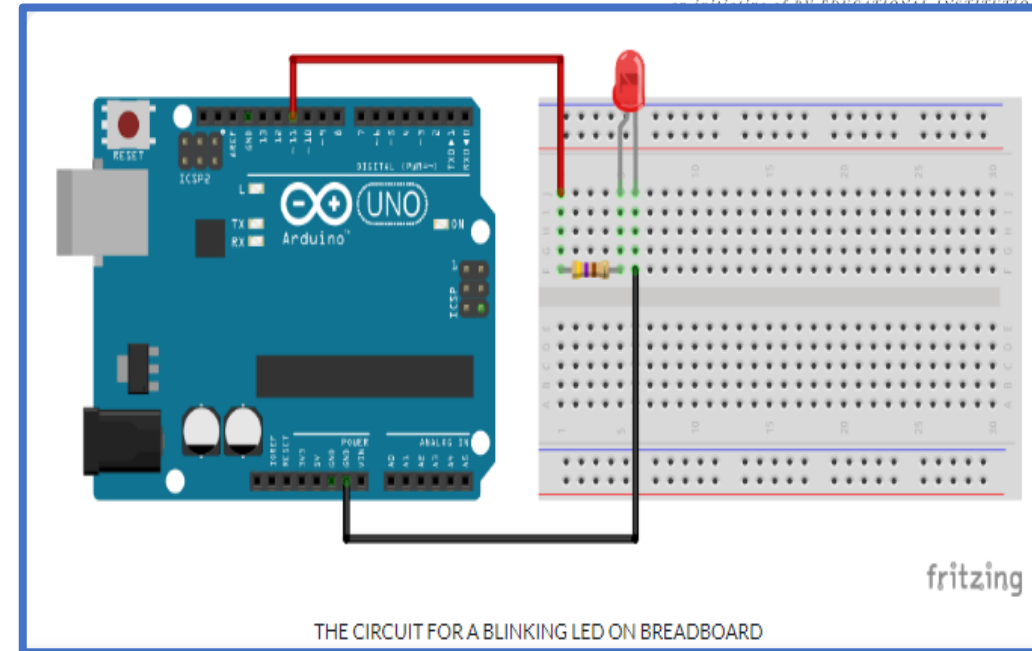
    digitalWrite(led1, LOW);  // turn the LED off by making the voltage LOW
    delay(1000);              // wait for a second

}
```

Note: Increase and decrease delay to see the variation in blinking.

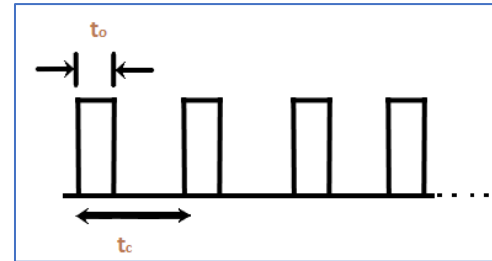
Blink an LED Connected to Breadboard

```
void setup() {  
    // initialize digital pin 11 as an output.  
    pinMode(11, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(11, HIGH); // turn the LED on (HIGH is the voltage level)  
    delay(1000);           // wait for a second  
    digitalWrite(11, LOW);  // turn the LED off by making the voltage LOW  
    delay(1000);  
}
```

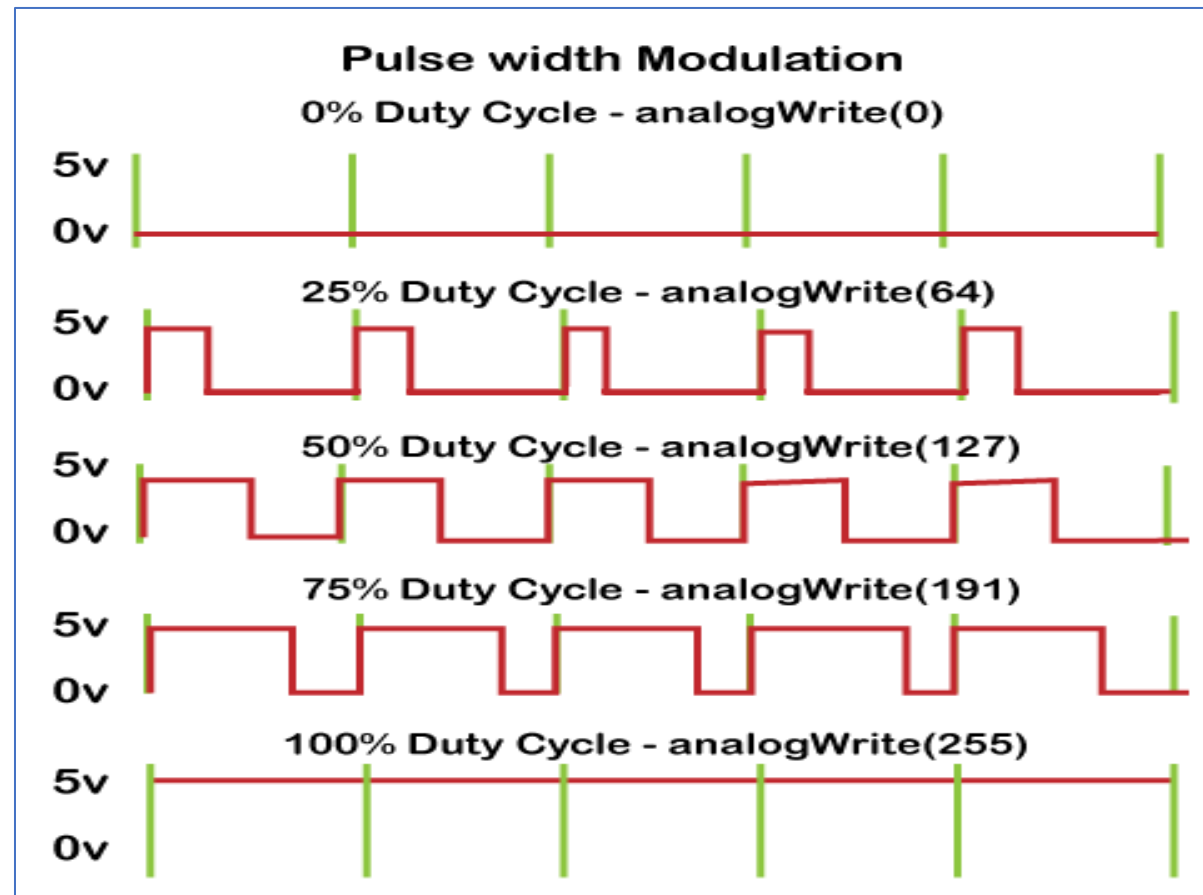


LED Fade-In Fade-Out

- The PWM (**Pulse Width Modulation**)
 - Method of controlling the average voltage.
 - It is a **stream of voltage pulses** that reduces the electric power supplied by the electrical signal.
 - The effective voltage is controlled by the width of individual pulses in a stream of voltage pulses of a PWM signal.
 - The common use of PWM pins includes controlling **LEDs** and **DC Motors**.



$$\text{duty cycle} = \frac{t_o}{t_c}$$



LED Fade-In Fade-Out

```
#define LED_PIN 11

void setup()
{
    pinMode(LED_PIN, OUTPUT);
}

void loop()
{
    for (int i = 0; i <= 255; i++) {
        analogWrite(LED_PIN, i);
        delay(10);
    }
```

```
for (int i = 255; i >= 0; i--)
{
    analogWrite(LED_PIN, i);
    delay(10);
}
}
```

Controlling LED using PUSH button

```
int led = 13;

int PinButton = 4;

void setup()
{
    pinMode(PinButton, INPUT);
    pinMode(led, OUTPUT);
}

void loop()
{
    int stateButton = digitalRead(PinButton);
    if (stateButton == 1)
```

```
    {
        digitalWrite(led, HIGH); // Turn on led
    }
    else
    {
        digitalWrite(led, LOW); //Turn off led
    }
    delay(20);
}
```


Traffic Light Controller

```
#define LED_PIN_1 11
#define LED_PIN_2 10
#define LED_PIN_3 9

void setup()
{
    pinMode(LED_PIN_1, OUTPUT);
    pinMode(LED_PIN_2, OUTPUT);
    pinMode(LED_PIN_3, OUTPUT);
}

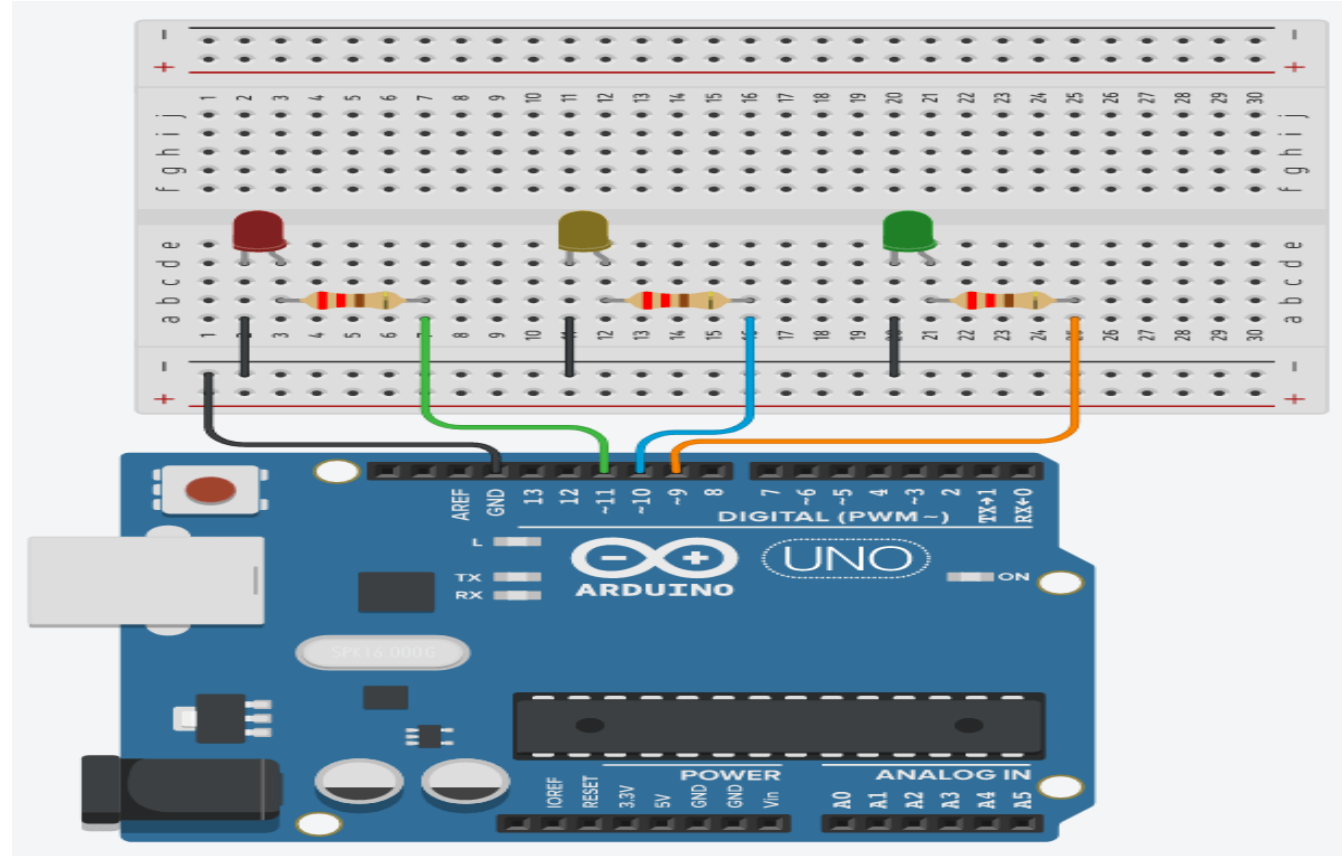
void loop()
{
```

```
    digitalWrite(LED_PIN_1, HIGH);
    digitalWrite(LED_PIN_2, LOW);
    digitalWrite(LED_PIN_3, LOW);
    delay(1000);

    digitalWrite(LED_PIN_1, LOW);
    digitalWrite(LED_PIN_2, HIGH);
    digitalWrite(LED_PIN_3, LOW);
    delay(1000);

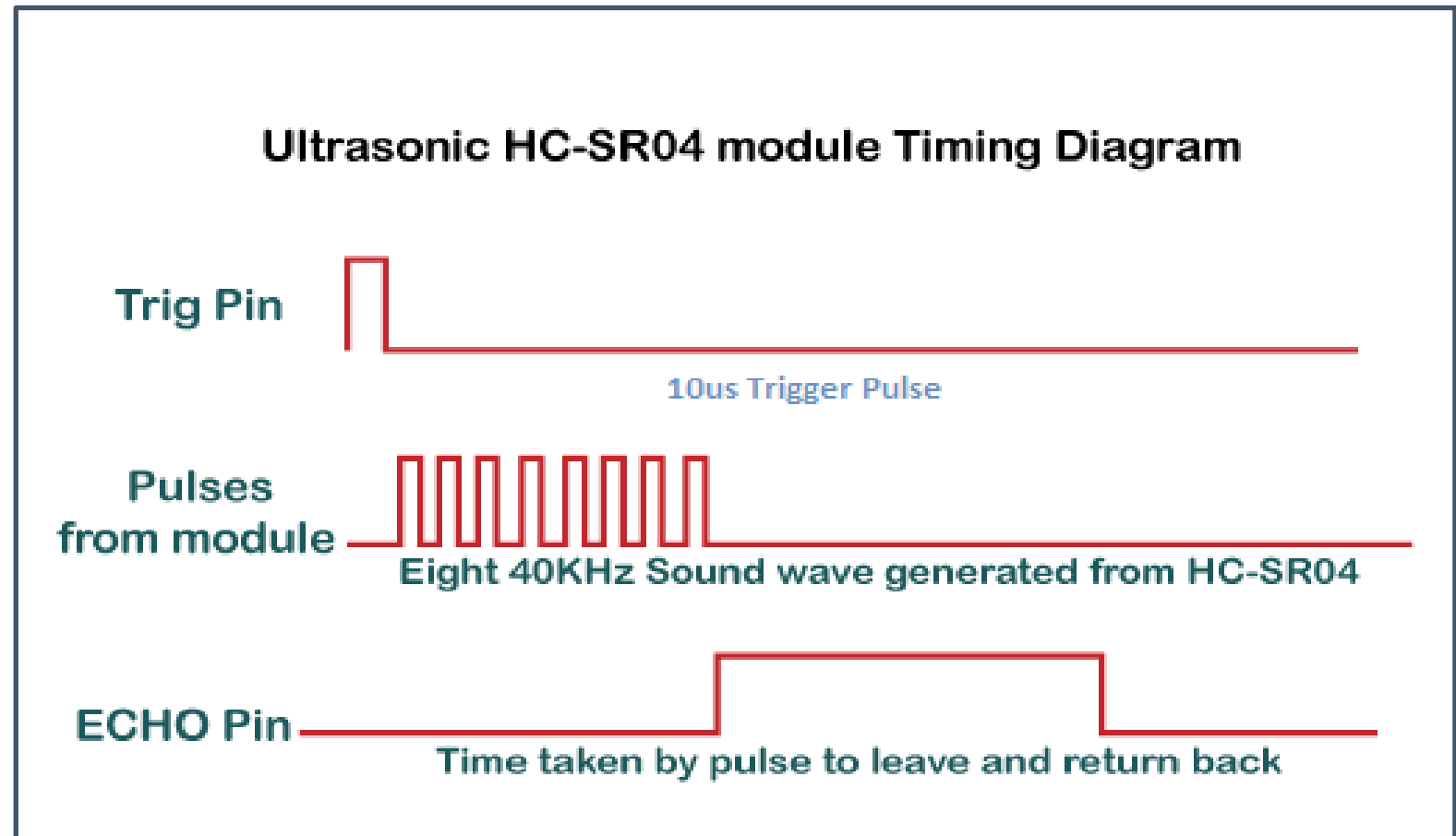
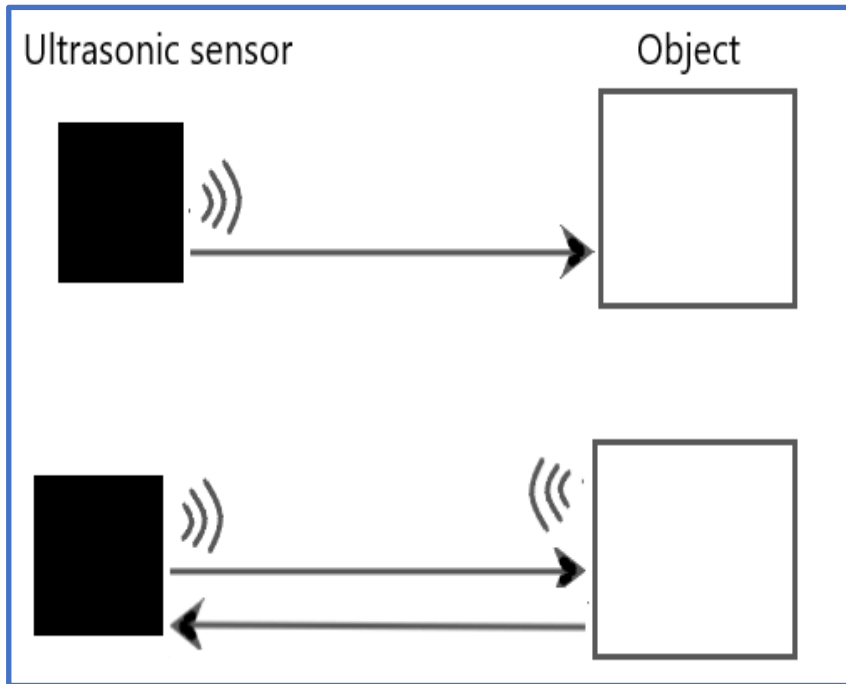
    digitalWrite(LED_PIN_1, LOW);
    digitalWrite(LED_PIN_2, LOW);
    digitalWrite(LED_PIN_3, HIGH); delay(1000); }
```

Traffic Light Controller

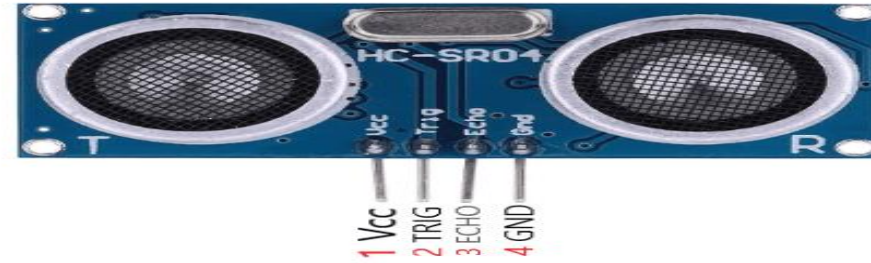
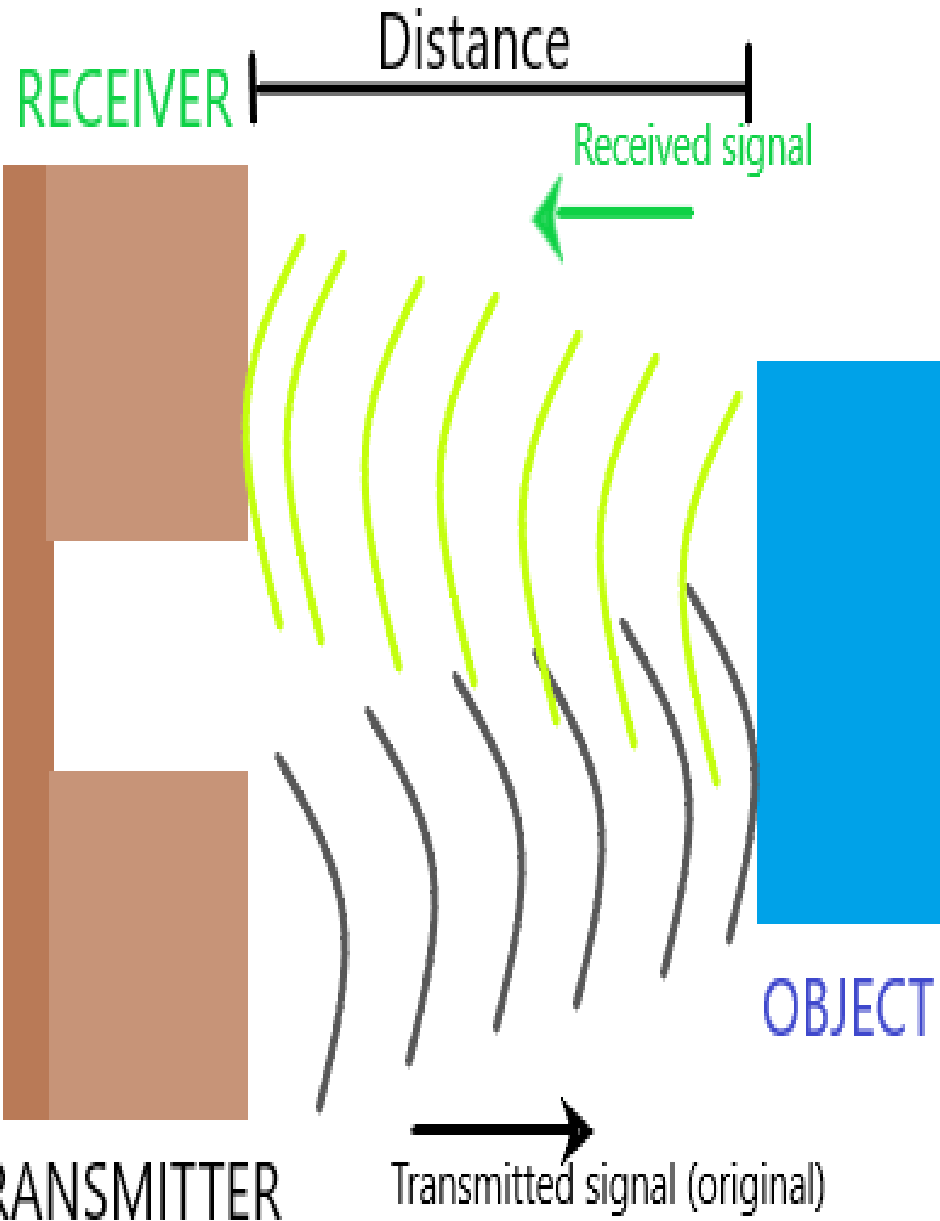


Arduino Ultrasonic Distance Sensor HC-SR04

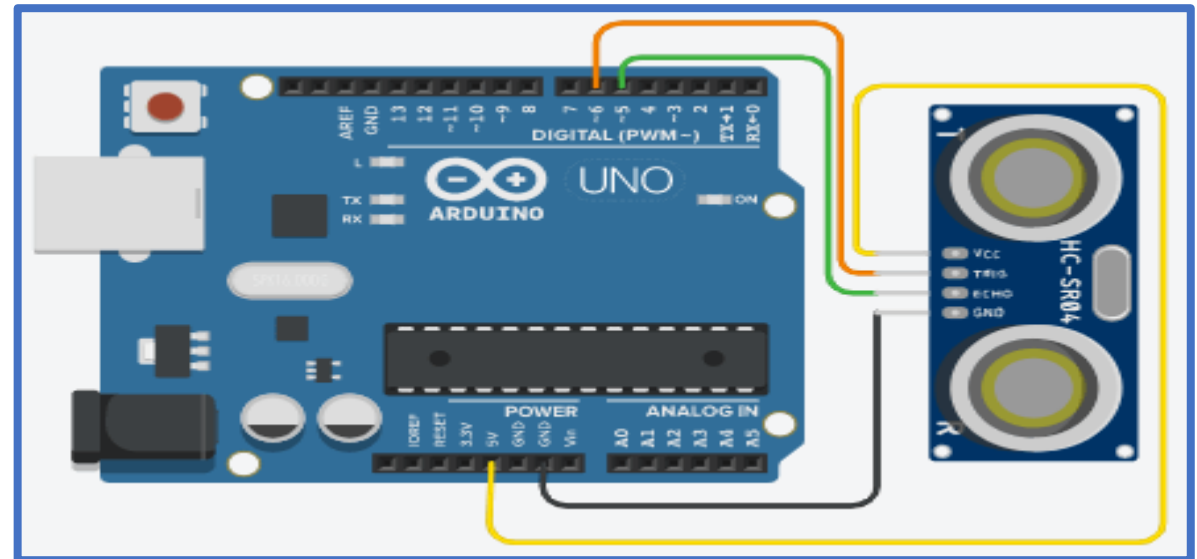
- Frequency of **40KHZ**
- The waves travels through the air and strikes the object on its path and bounce back to reach the module.



Arduino Ultrasonic Distance Sensor HC-SR04



- VCC pin to 5V
- GND pin to GND
- TRIG pin to pin 6
- ECHO pin to pin 5



Arduino Ultrasonic Distance Sensor

An object is 40cm away from the Ultrasonic sensor. The speed of sound in air is 340m/s. We need to calculate the time (in Microseconds).

$$\begin{aligned}v &= 340\text{m/s} = 0.034\text{cm/us} \\ \text{time} &= \text{distance/speed} \\ &= 40/0.034 \\ &= 1176 \text{ microseconds}\end{aligned}$$

The speed of sound from the echo pin will double because the wave travels forward and backward (bounces).

So, to calculate the distance, we need to divide it by 2

$$\begin{aligned}\text{distance} &= \text{time} \times \text{speed of sound}/2 \\ &= \text{time} \times 0.034/2\end{aligned}$$

Arduino Ultrasonic Distance Sensor

```
// define variables

long duration; // variable for the duration of sound wave
travel

int distance; // variable for the distance measurement

void setup() {

    pinMode(trigPin, OUTPUT); // Sets the trigPin as an
    OUTPUT

    pinMode(echoPin, INPUT); // Sets the echoPin as an
    INPUT

    Serial.begin(9600); // // Serial Communication is starting
    with 9600 of baudrate speed

    Serial.println("Ultrasonic Sensor HC-SR04 Test");

    // print some text in Serial Monitor

    Serial.println("with Arduino UNO R3");

}
```

```
void loop() {

    // Clears the trigPin condition

    digitalWrite(trigPin, LOW);

    delayMicroseconds(2);

    // Sets the trigPin HIGH (ACTIVE) for 10 microseconds

    digitalWrite(trigPin, HIGH);

    delayMicroseconds(10);

    digitalWrite(trigPin, LOW);

}
```

Arduino Ultrasonic Distance Sensor

```
// Reads the echoPin, returns the sound wave travel time in
microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance = duration * 0.034 / 2; // Speed of sound wave divided
by 2 (go and back)

// Displays the distance on the Serial Monitor

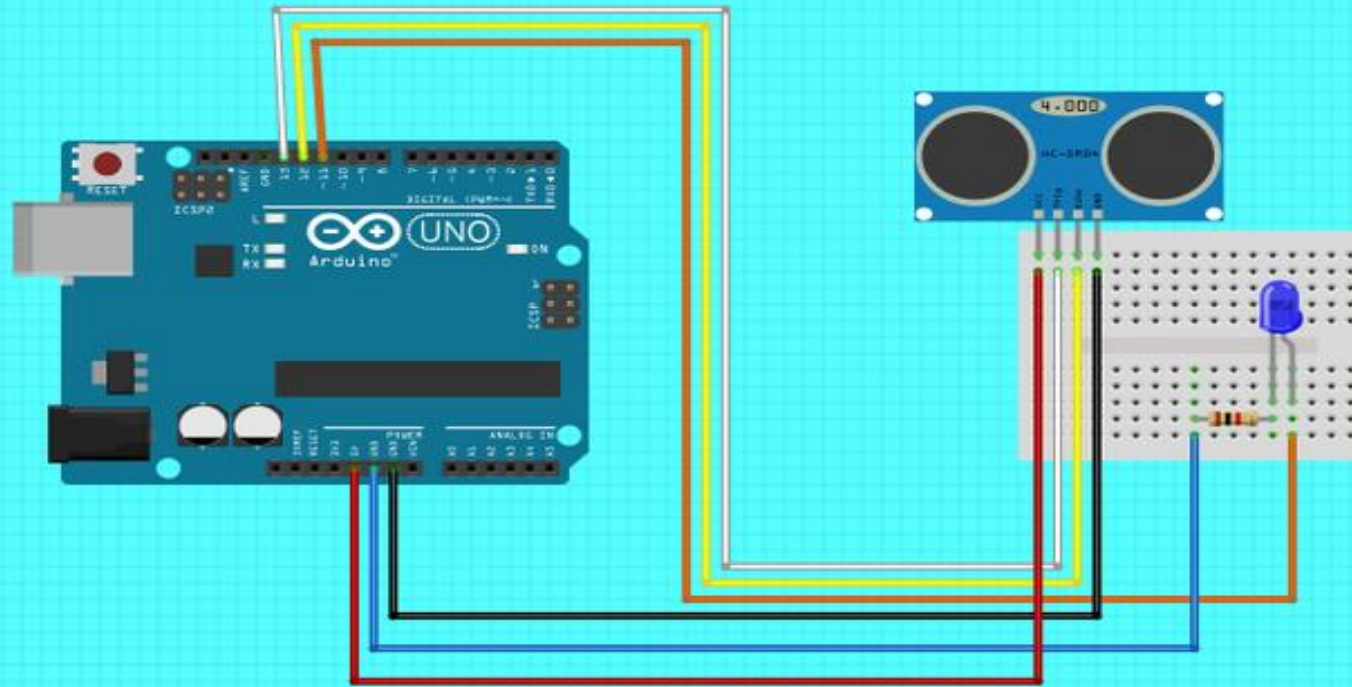
Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

}
```

Controlling LED using Ultrasonic Sensor

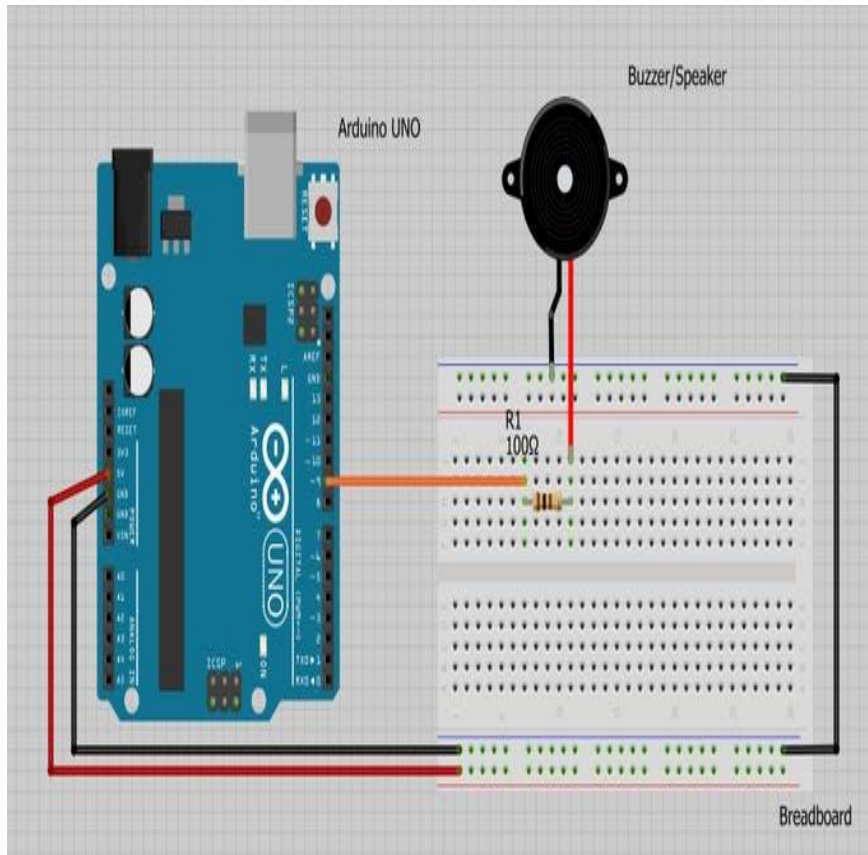


fritzing


```
#define echoPin 2 // attach pin D2 Arduino to pin Echo of HC-  
SR04  
  
#define trigPin 3 //attach pin D3 Arduino to pin Trig of HC-  
SR04  
  
// defines variables  
  
long duration; // variable for the duration of sound wave travel  
  
int distance; // variable for the distance measurement  
  
void setup() {  
  
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an  
OUTPUT  
  
    pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT  
  
    Serial.begin(9600); // // Serial Communication is starting with  
9600 of baudrate speed  
  
    Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print  
some text in Serial Monitor  
  
    Serial.println("with Arduino UNO R3");  
  
}
```

```
void loop() {  
  
    // Clears the trigPin condition  
  
    digitalWrite(trigPin, LOW);  
  
    delayMicroseconds(2);  
  
    // Sets the trigPin HIGH (ACTIVE) for 10 microseconds  
  
    digitalWrite(trigPin, HIGH);  
  
    delayMicroseconds(10);  
  
    digitalWrite(trigPin, LOW);  
  
    // Reads the echoPin, returns the sound wave travel time in  
microseconds  
  
    duration = pulseIn(echoPin, HIGH);  
  
    // Calculating the distance  
  
    distance = duration * 0.034 / 2; // Speed of sound wave divided  
by 2 (go and back)  
  
    if (distance < 10)  
  
    { digitalWrite(8,HIGH);}
```

Arduino with Buzzer



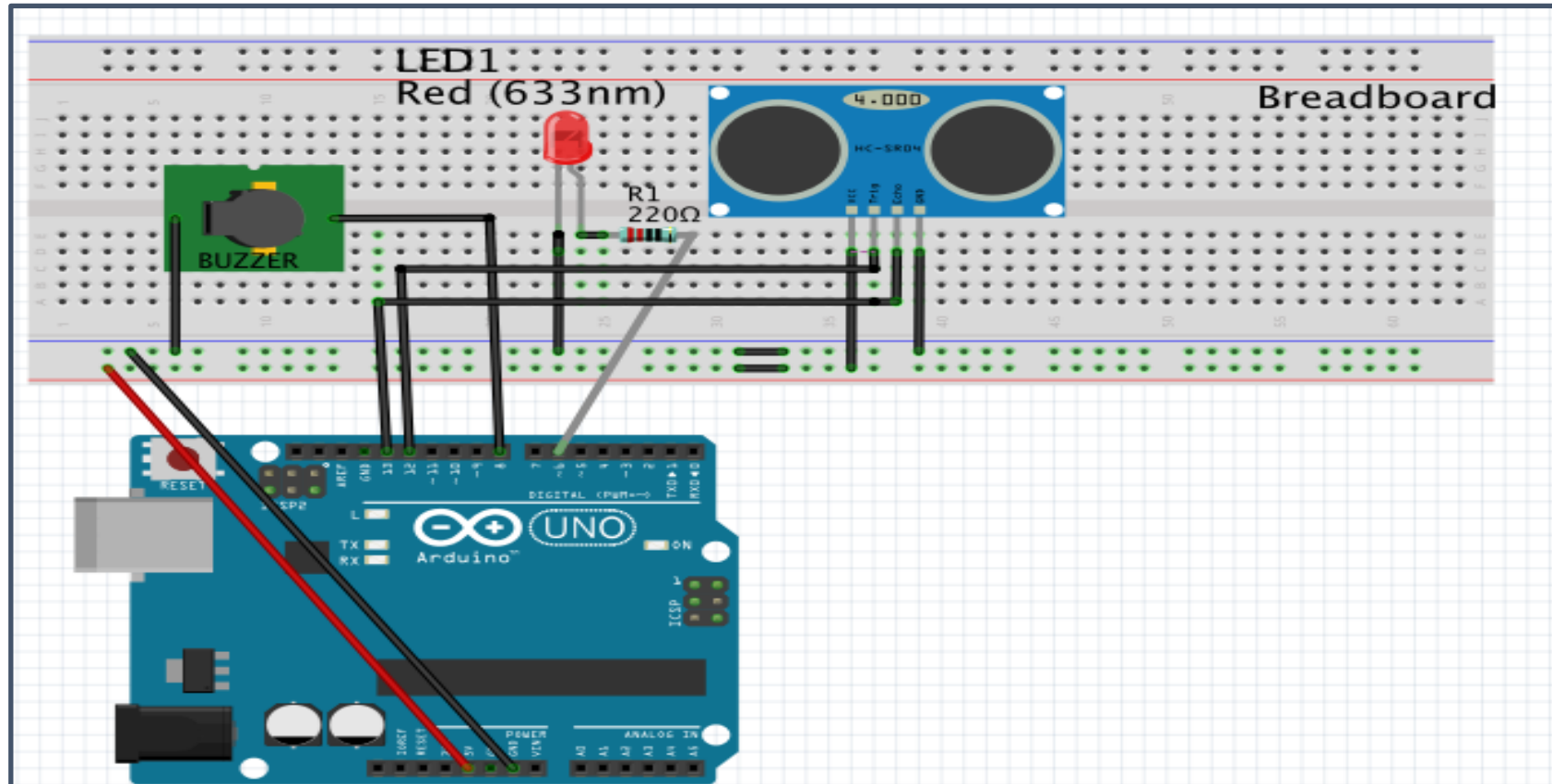
```
const int buzzer = 9; //buzzer to pin 9

void setup() {
  pinMode(buzzer, OUTPUT);
  // Set buzzer - pin 9 as an output
}

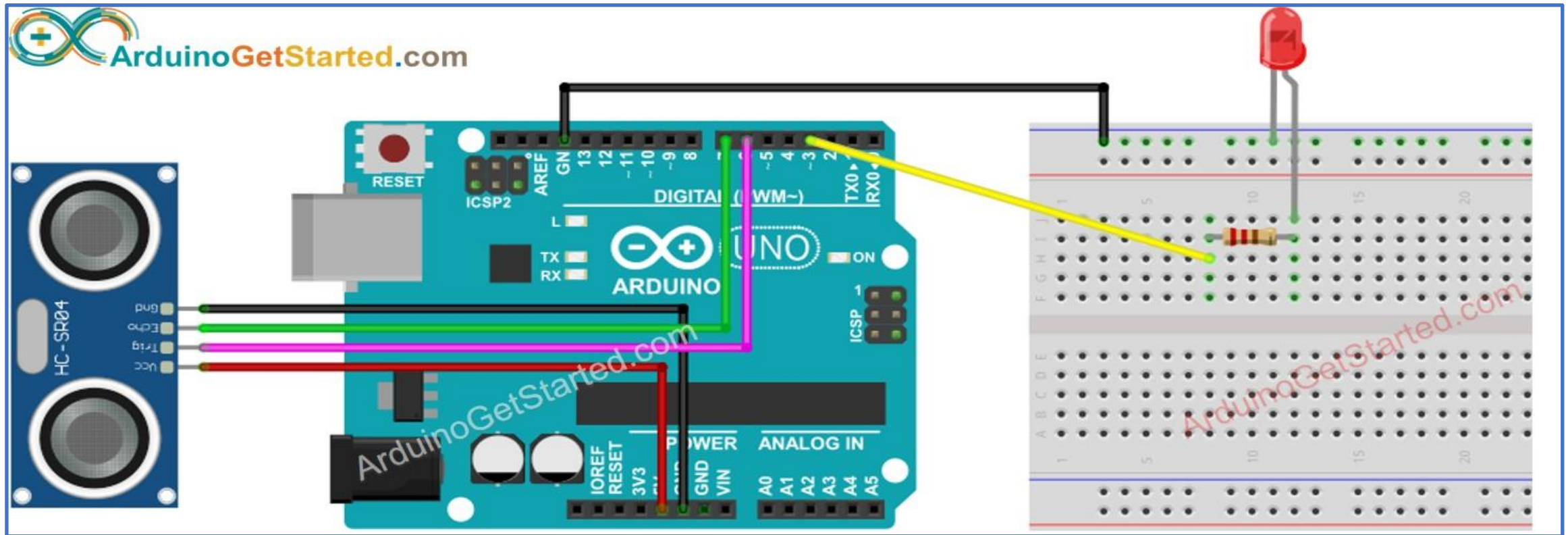
void loop() {
  tone(buzzer, 1000); // Send 1KHz sound signal...
  delay(1000);        // ...for 1 sec
  noTone(buzzer);     // Stop sound...
  delay(1000);        // ...for 1sec
}
```

Ultrasonic sensor, led and buzzer

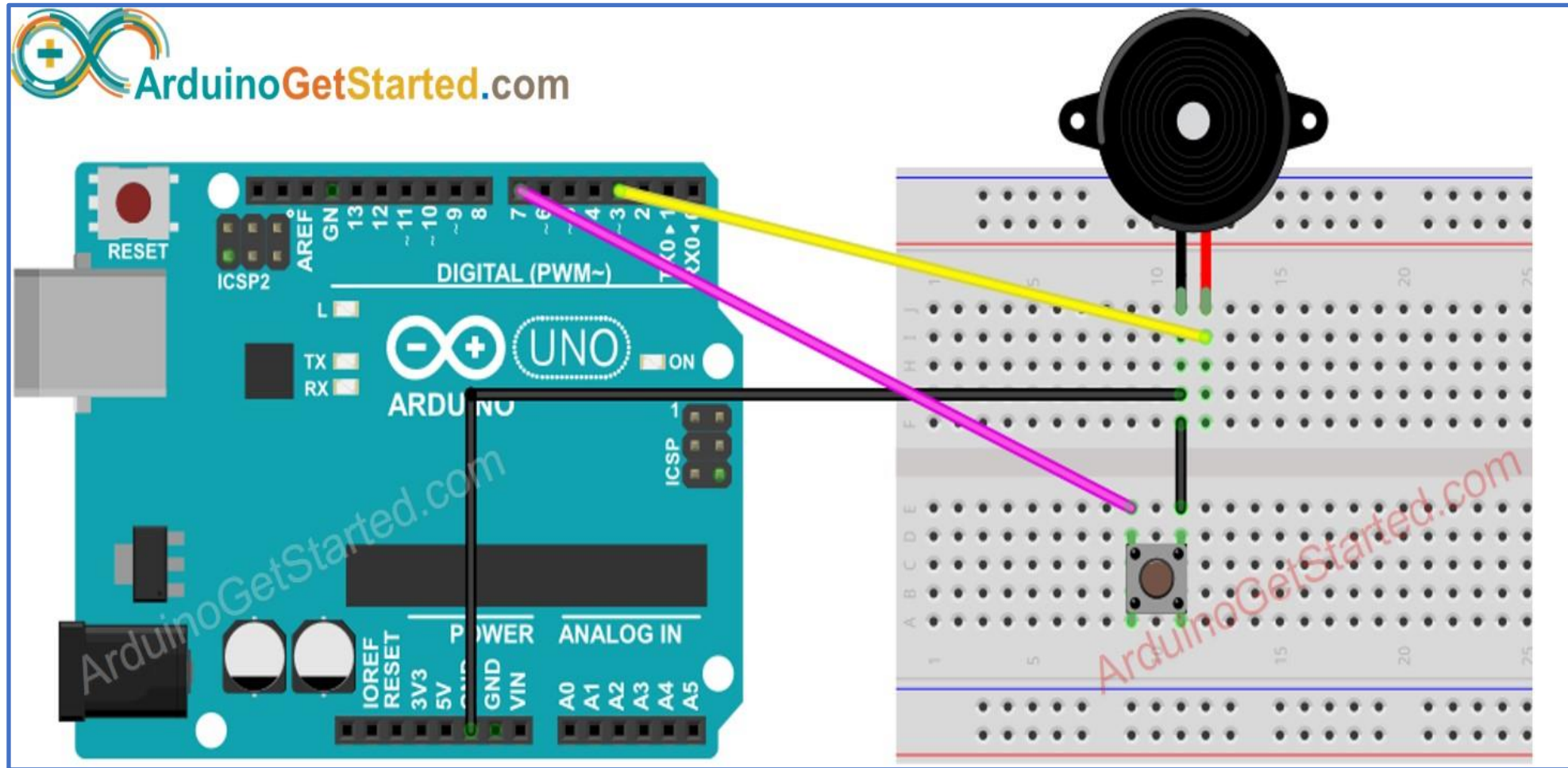
(Intrusion Detection)



Controlling Multiple LED's with Ultrasonic Sensor



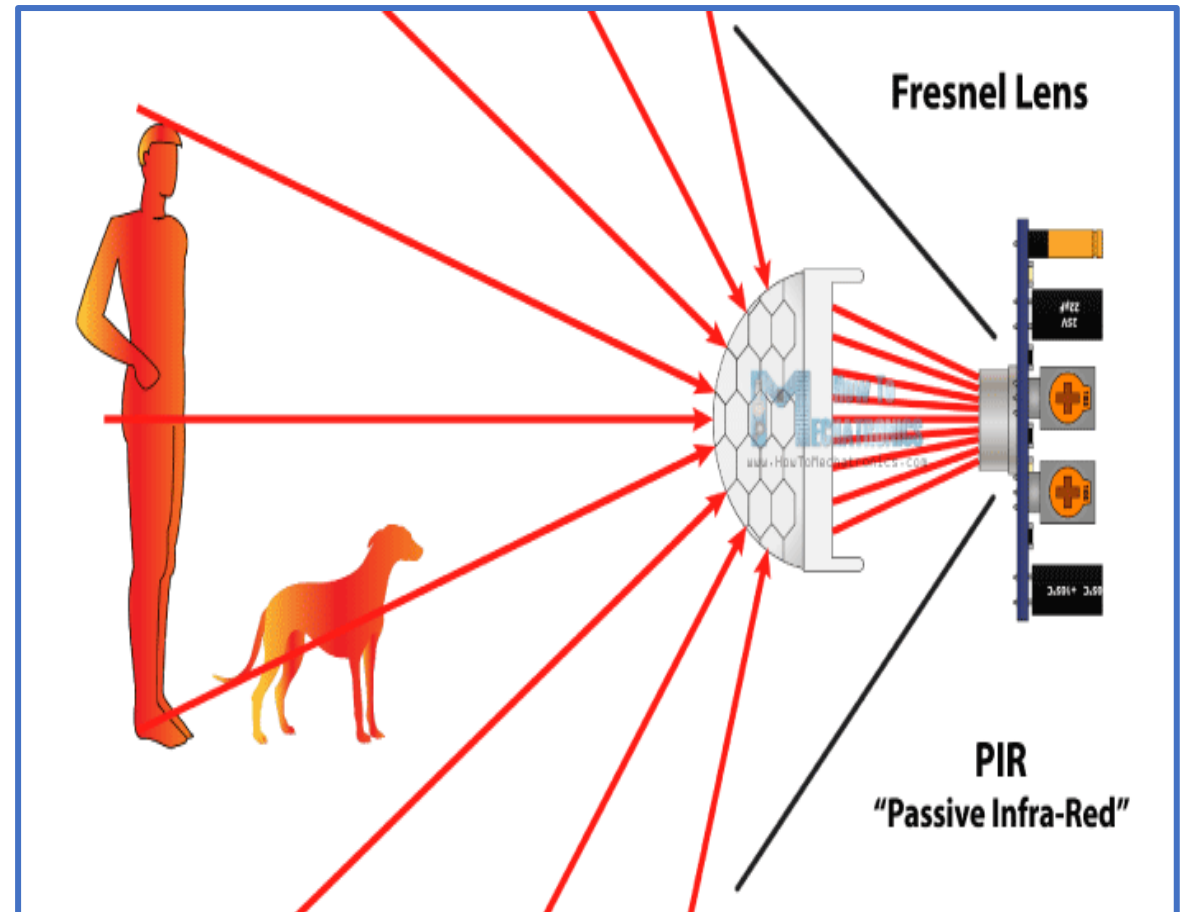
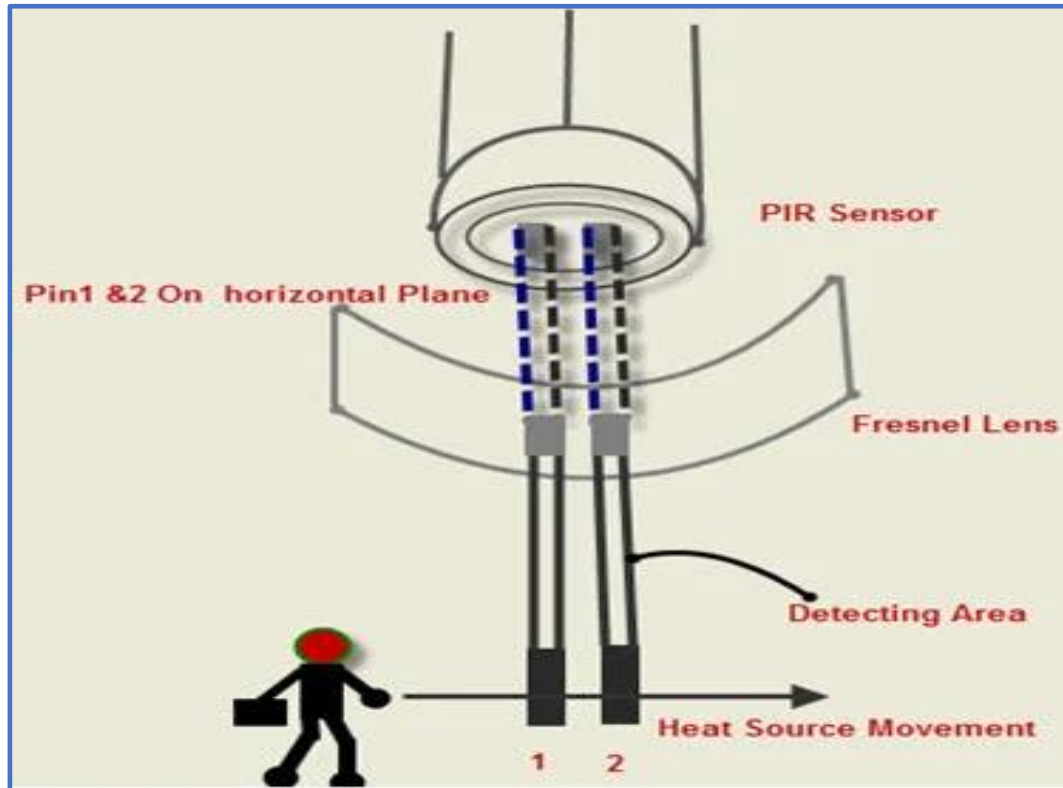
Controlling Buzzer Using Push Button



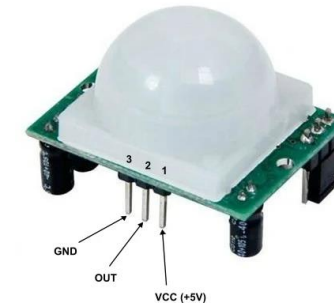
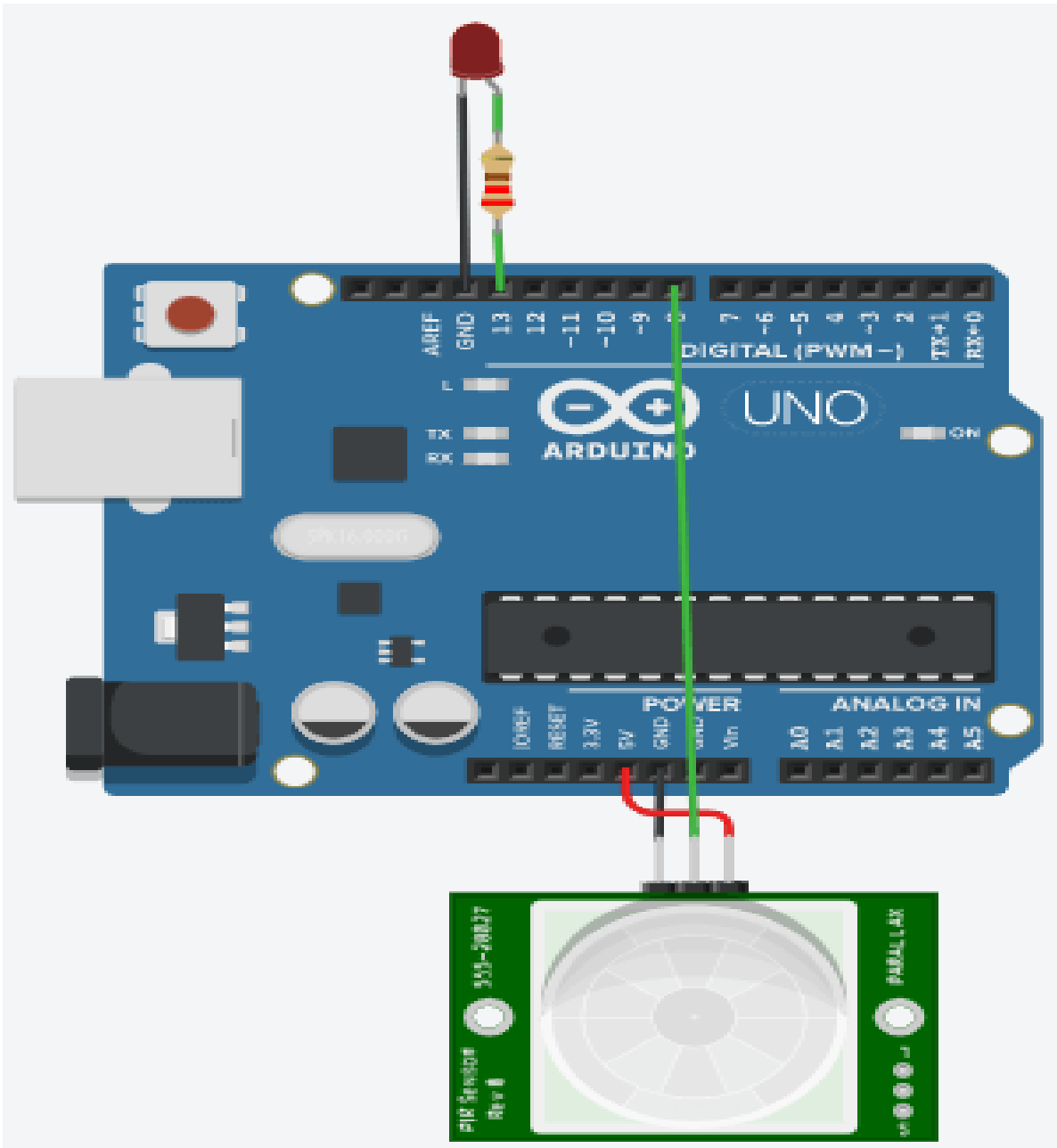
PIR Sensor

To sense motion.

They are small, inexpensive, low-power, easy to use and don't wear out
Appliances and gadgets used in homes or businesses.



Sensitivity Potentiometer Time Potentiometer.



THANK YOU