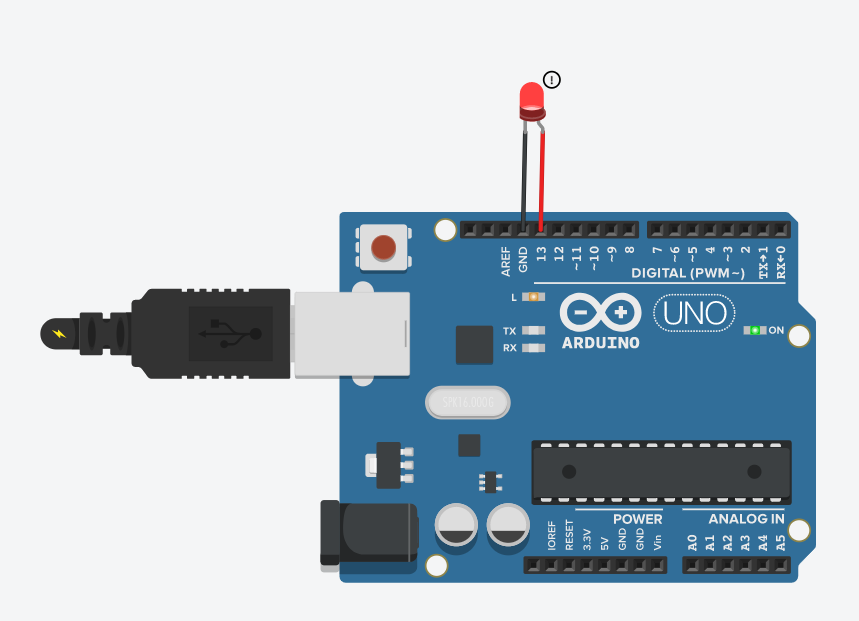
**1.PROBLEM STATEMENT:**

Manipulating LED light with the help of ArduinoUNO

**CIRCUIT DESIGN:-**



**CODE:**

int led = 13;

void setup()

{

pinMode(led, OUTPUT);

}

void loop()

{

digitalWrite(led, HIGH);

delay(1000); // Wait for 1000 millisecond(s)

digitalWrite(led, LOW);

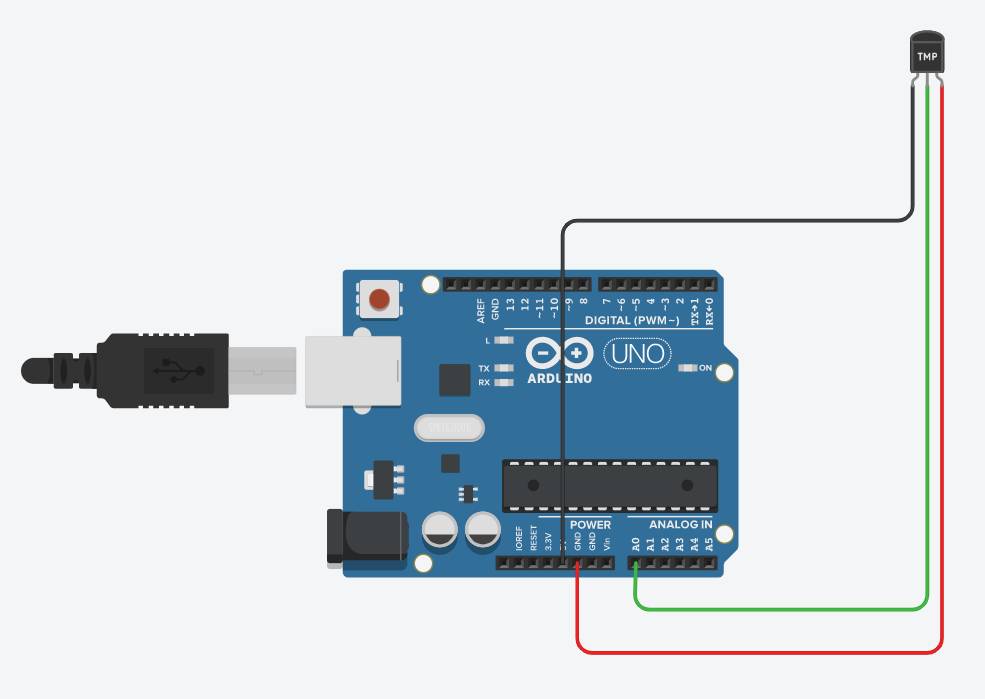
delay(1000); // Wait for 1000 millisecond(s)

}

**2.PROBLEM STATEMENT:**

Temperature sensor with the help of Arduino UNO.

**CIRCUIT DESIGN:-**



**CODE:**

float TempValue, TempValueC,TempValuemv;

void setup()

{

pinMode(A0, INPUT);

Serial.begin(9600);

}

void loop()

{

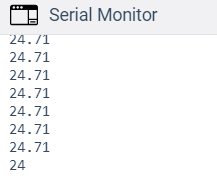
TempValue=analogRead(A0);

TempValuemv=TempValue\*5000/1024;

TempValueC=(TempValuemv/10+(-50));

Serial.println(TempValueC); }

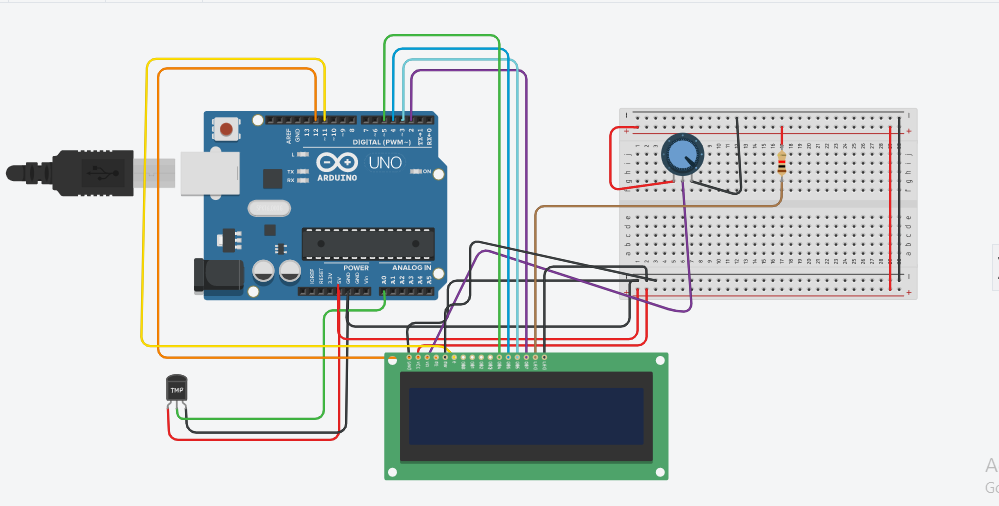
**OUTPUT:**



**3.PROBLEM STATEMENT:**

Temperature sensor with LCD display

**CIRCUIT DESIGN:-**



**CODE:**

#include<LiquidCrystal.h>

LiquidCrystal lcd(12,11,5,4,3,2);

#define pinTemp A0

void setup()

{

Serial.begin(9600);

pinMode(pinTemp,INPUT);

lcd.begin(16,2);

lcd.print("Temperature:");

}

void loop()

{

float temp=((analogRead(pinTemp)\*(5.0/1024))-0.5)/0.01;

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print(temp);

Serial.println(temp);

delay(1000);

}

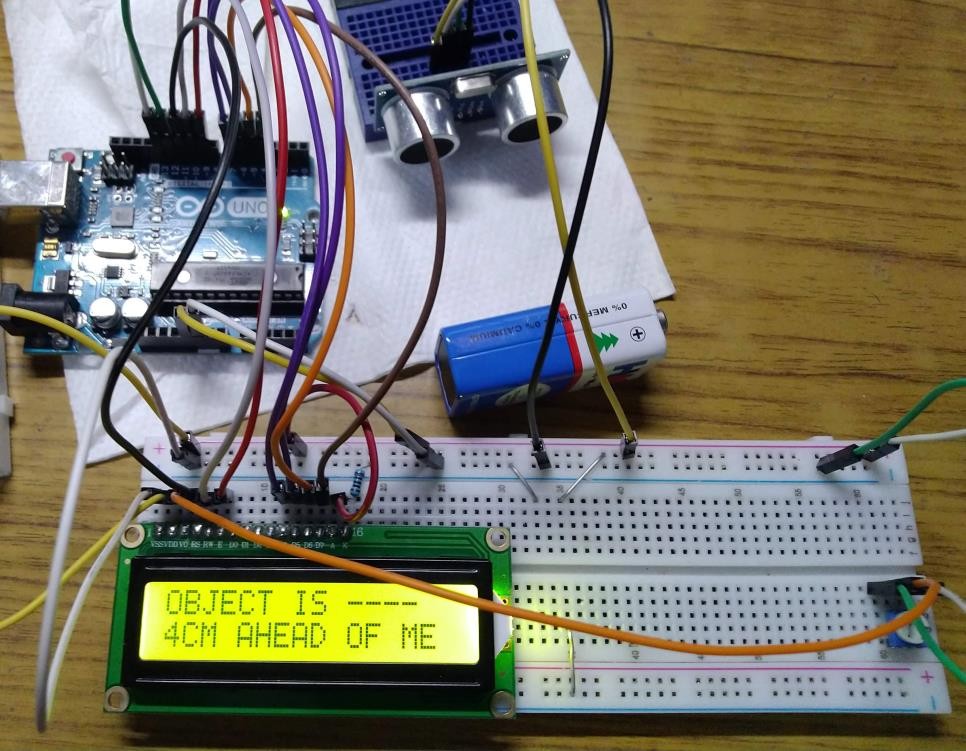
**OUTPUT:**



# 4.PROBLEM STATEMENT

Implementing ultrasonic sensor with lcd using Arduino uno.

**CIRCUIT DESIGN:-**

*+*

**CODE:**

#include <LiquidCrystal.h>LiquidCrystallcd(11, 10, 9, 7, 6, 5, 4); intpingPin =13;

intinPin = 12; void setup() { lcd.begin(16, 2);

lcd.clear(); lcd.setCursor(0, 0); lcd.print("---D.M.T.K.T---");

lcd.setCursor(0, 1); lcd.print(" Ver : MK-II"); delay(2500);

}

void loop(){ lcd.clear(); lcd.setCursor(0, 0);

lcd.print(" READING"); lcd.setCursor(0,1);

lcd.print(" ENVIRONMENT"); delay(1000);

long duration, inches, cm; pinMode(pingPin, OUTPUT); digitalWrite(pingPin, LOW); delayMicroseconds(2); digitalWrite(pingPin, HIGH);

delayMicroseconds(10); digitalWrite(pingPin, LOW); pinMode(inPin, INPUT); duration = pulseIn(inPin,HIGH);

cm = microsecondsToCentimeters(duration);

if (cm >400){lcd.clear(); lcd.setCursor(0,0);

lcd.print("$$$$ DANGER $$$$"); lcd.setCursor(0,1); lcd.print("$$$$ BEWARE $$$$");

}

else { lcd.clear();

lcd.setCursor(0, 0); lcd.print("OBJECTIS ");

lcd.setCursor(0, 1); lcd.print(cm);

lcd.print("CM AHEAD OF ME");

}

delay(3000);

}

long microsecondsToInches(long microseconds)

{

return microseconds / 74 / 2;

}

long microsecondsToCentimeters(long microseconds)

{

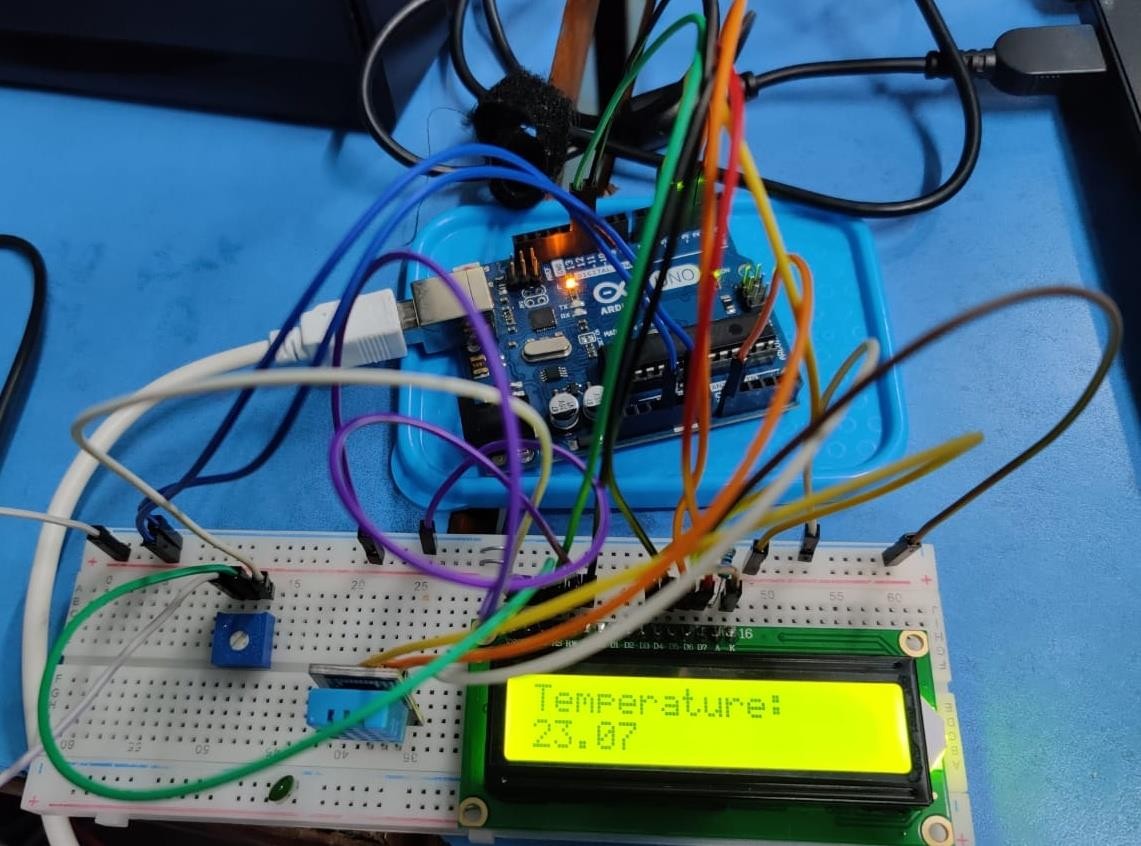
return microseconds / 29 / 2;

}

# 5.PROBLEM STATEMENT

Implementing “DHT 11 Sensor “ with LCD using Arduino.

**CIRCUIT DESIGN:-**



**CODE:**

#include <DHT.h> #include <DHT\_U.h> #include<LiquidCrystal.h>

#define DHTPIN 8

LiquidCrystallcd(12, 11, 5, 4, 3, 2); #define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup()

{

lcd.begin(16, 2);

dht.begin(); lcd.print("Temperature(C) :");

}

void loop() { delay(500); lcd.setCursor(0, 1);

float h = dht.readHumidity(); float f = dht.readTemperature();

if (isnan(h) || isnan(f)) {

lcd.print("WAIT"); return; }lcd.print(f);

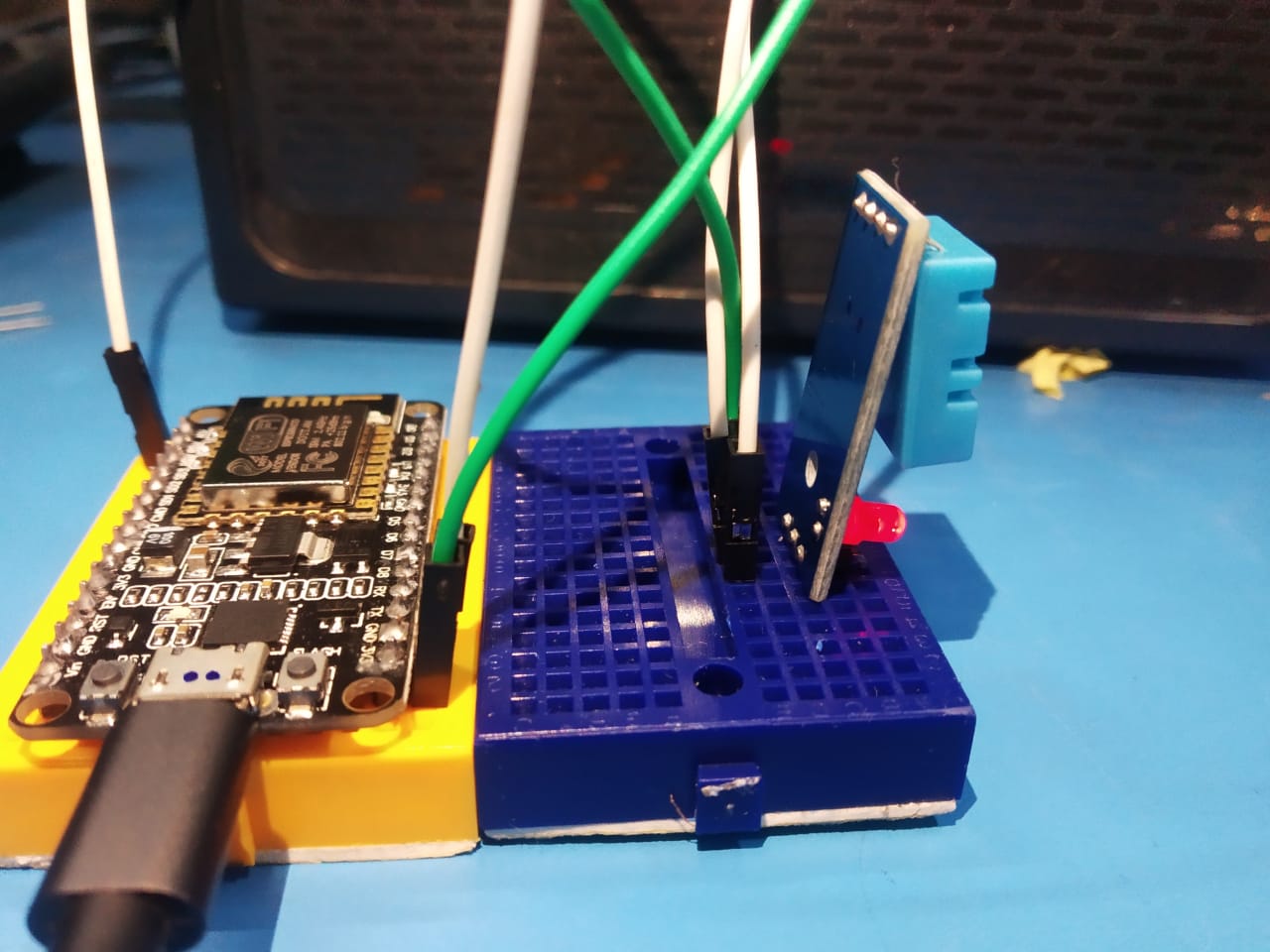
lcd.setCursor(7,1);

}

# 6.PROBLEM STATEMENT

Implementing DHT11 with NodeMCU on Thingspeak.

**CIRCUIT DESIGN:-**

**

**CODE:**

#include <ThingSpeak.h>

#include <DHT.h>

#include <DHT\_U.h>

#include <ESP8266WiFi.h>

String apiKey = "C4HVMAUGUITVR658";

const char \*ssid = "JIOFI X";

const char \*pass = "7003131847";

const char \*server = "api.thingspeak.com";

#define DHTPIN D1

DHT dht(DHTPIN, DHT11);

WiFiClient client;

void setup()

{

Serial.begin(9600);

delay(10);

dht.begin();

Serial.println("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

}

void loop()

{

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(h) || isnan(t))

{

Serial.println("Failed to read from DHT sensor!");

return;

}

if (client.connect(server,80)) // "184.106.153.149" or api.thingspeak.com

{

String postStr = apiKey;

postStr +="&field1=";

postStr += String(t);

postStr +="&field2=";

postStr += String(h);

postStr += "\r\n\r\n";

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

Serial.print("Temperature: ");

Serial.print(t);

Serial.print(" degrees Celcius, Humidity: ");

Serial.print(h);

Serial.println("%. Send to Thingspeak.");

}

client.stop();

Serial.println("Waiting...");

delay(1000);

}

# 7.PROBLEM STATEMENT

# Reading Ultrasonic Data using NodeMCU on Thingspeak.

**CIRCUIT DESIGN:-**



**CODE:**

#include <ESP8266WiFi.h>

#include <ThingSpeak.h>

// defines pins numbers

String apiKey = "X73Y7PMMP5EDAN0G"; // Enter your Write API key from ThingSpeak

const char \*ssid = "SSIDSSID"; // replace with your wifi ssid and wpa2 key

const char \*pass = "9433933920";

const char\* server = "api.thingspeak.com";

const int trigPin = 2; //D4

const int echoPin = 0; //D3

// defines variables

long duration;

float distance;

WiFiClient client;

void setup() {

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

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

Serial.begin(9600);

delay(10);

Serial.println("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");// Starts the serial communication

}

void loop() {

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

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;

// Prints the distance on the Serial Monitor

if (client.connect(server,80)) // "184.106.153.149" or api.thingspeak.com

{

String postStr = apiKey;

postStr +="&field1=";

postStr += String(distance);

postStr += "\r\n\r\n";

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

Serial.print("distance: ");

Serial.print(distance);

Serial.println("%. Send to Thingspeak.");

}

client.stop();

Serial.println("Waiting...");

// thingspeak needs minimum 15 sec delay between updates, i've set it to 1 second

delay(1000);

}

**OUTPUT:**

