**WATER LEVEL MONITORING SYSTEM WITH DISPLAY AND VISUALIZING THE AGGREGATED DATA IN THINGSPEAK**

**AIM:**

To implement a water level monitoring system with display device and visualizing the aggregated data in thingspeak

**COMPONENTS REQUIRED:**

Hardware requirement:

1. Arduino UNO
2. ESP8266 NodeMCU
3. Ultrasonic sensor HC
4. LCD Display
5. buzzer
6. Potentiometer
7. Jumper wires

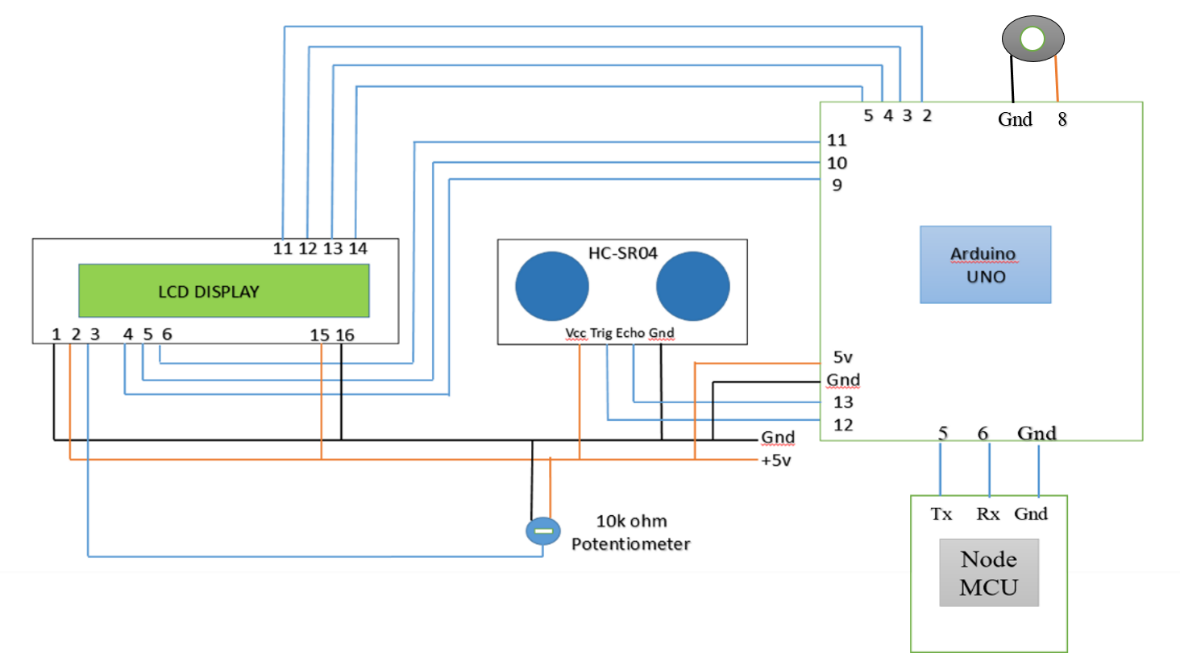
Software requirement:

1. Arduino IDE
2. MathWorks account
3. Thingspeak

**PROCEDURE:**

1. Connect the components as per the circuit diagram.
2. Connect Arduino UNO and NodeMCU with PC via cable.
3. Upload the Arduino Code and NodeMCU code to its respective board. Make sure that the Board and port are selected correctly in Arduino IDE.
4. Login thingspeak through MATLAB account.
5. Create a channel in thingspeak to view the output.
6. Adjust the potentiometer if required to clearly view the water level display in LCD.

**CIRCUIT DIAGRAM:**



**CODE:**

**Arduino code: (transmitter side code)**

#include <LiquidCrystal.h> //Load Liquid Crystal Library

LiquidCrystal LCD(11,10,9,2,3,4,7); //Create Liquid Crystal Object called LCD

#include <SoftwareSerial.h>

#define SOUND\_SPEED 0.034

#define CM\_TO\_INCH 0.393701

#define trigPin 13 //Sensor Echo pin connected to Arduino pin 13

#define echoPin 12 //Sensor Trig pin connected to Arduino pin 12

#define buzPin 8 // buzzer positive to Arduino pin 8

SoftwareSerial espSerial(5, 6);

String str;

void setup()

{

Serial.begin(115200);

espSerial.begin(115200);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(buzPin, OUTPUT);

LCD.begin(16,2); //Tell Arduino to start your 16 column 2 row LCD

LCD.setCursor(0,0); //Set LCD cursor to upper left corner, column 0, row 0

LCD.print("Water level:"); //Print Message on First Row

}

void loop() {

long duration, distance;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* SOUND\_SPEED/2;

if (distance <=1 && distance >=0) {

digitalWrite(buzPin, HIGH);}

else{

digitalWrite(buzPin, LOW);}

LCD.setCursor(0,1); //Set cursor to first column of second row

LCD.print(" "); //Print blanks to clear the row

LCD.setCursor(0,1); //Set Cursor again to first column of second row

LCD.print(distance); //Print measured distance

LCD.print(" cm"); //Print your units.

Serial.println(distance);

str =String(distance);

espSerial.println(str);

delay(1000); //pause to let things settle

}

**NodeMCU code: (Receiver side code)**

#include "ThingSpeak.h"

#include <ESP8266WiFi.h>

//----------- Enter you Wi-Fi Details---------//

char ssid[] = "Manda patharam..."; //SSID

char pass[] = "06060606"; // Password

//-------------------------------------------//

WiFiClient client;

unsigned long myChannelField = 1669091; // Channel ID

const int ChannelField = 1; // Which channel to write data

const char \* myWriteAPIKey = "FQFHJ7AMQMNH1B4K"; // Your write API Key

int distance;

void setup() {

// Open serial communications and wait for port to open:

Serial.begin(9600);

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client);

while (!Serial) {

; // wait for serial port to connect. Needed for native USB port only

}

}

void loop()

{

// run over and over

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

{

Serial.print("Attempting to connect to SSID: ");

Serial.println(ssid);

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

{

WiFi.begin(ssid, pass);

Serial.print(".");

delay(5000);

}

Serial.println("\nConnected.");

if (Serial.available())

{

distance=Serial.read();

Serial.write(Serial.read());

}

}

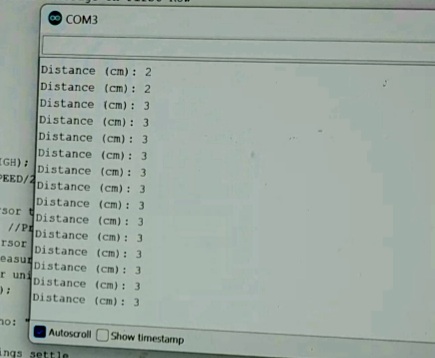
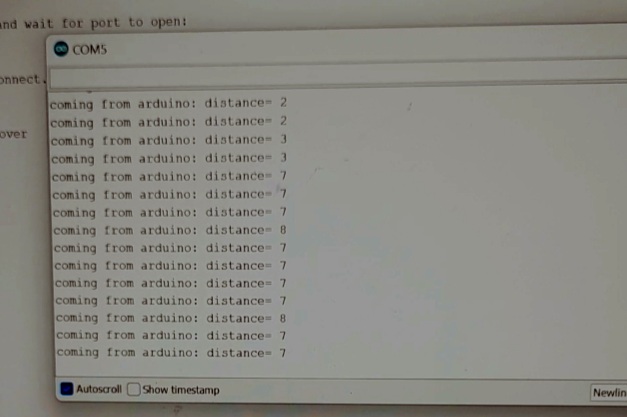
ThingSpeak.writeField(myChannelField, ChannelField, distance, myWriteAPIKey);

delay(1000);

}

**OUTPUTS:**

Ardiuno Serial Monitor: Node Mcu Serial Monitor:

Thingspeak: LCD Display:

