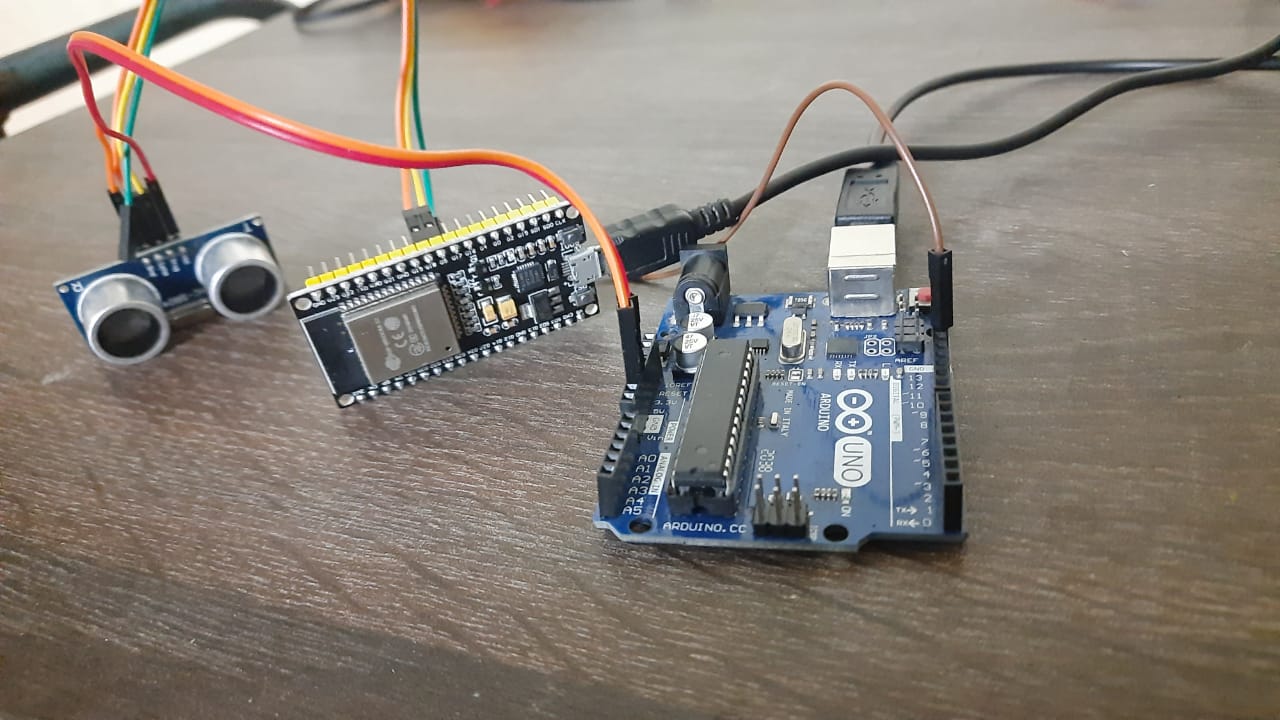
|  |  |
| --- | --- |
| Name | Shaunak Deshpande |
| Division | IC-C |
| Roll number | 39 |
| Batch | 2 |
| GR Number | 11911180 |

**Lab Assignment No.6**

**Title: Ultrasonic Sensor interfacing with ESP32 and sends this data to the Thingspeak.**

**Hardware implementation :**



**Program Code :**

#include <WiFi.h>

#include "ThingSpeak.h" // always include thingspeak header file after other header files and custom macros

#define SECRET\_SSID "Galaxy M21A398" // replace MySSID with your WiFi network name

#define SECRET\_PASS "55555555" // replace MyPassword with your WiFi password

#define SECRET\_CH\_ID 1603613 // replace 0000000 with your channel number

#define SECRET\_WRITE\_APIKEY "1EJRT7KAKIR8YZ1M" // replace XYZ with your channel write API Key

char ssid[] = SECRET\_SSID; // your network SSID (name)

char pass[] = SECRET\_PASS; // your network password

int keyIndex = 0; // your network key Index number (needed only for WEP)

WiFiClient client;

unsigned long myChannelNumber = SECRET\_CH\_ID;

const char \* myWriteAPIKey = SECRET\_WRITE\_APIKEY;

const int pingPin = 5;

const int echoPin = 18;

void setup() {

Serial.begin(115200); //Initialize serial

while (!Serial) {

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

}

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client); // Initialize ThingSpeak

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

pinMode(echoPin, INPUT);}

void loop() {

// Connect or reconnect to WiFi

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

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

Serial.println(SECRET\_SSID);

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

WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network

Serial.print(".");

delay(5000);

}

Serial.println("\nConnected.");

}

long duration, inches, cm;

pinMode(pingPin, OUTPUT);

digitalWrite(pingPin, LOW);

delayMicroseconds(2);

digitalWrite(pingPin, HIGH);

delayMicroseconds(10);

digitalWrite(pingPin, LOW);

pinMode(echoPin, INPUT);

duration = pulseIn(echoPin, HIGH);

inches = microsecondsToInches(duration);

cm = microsecondsToCentimeters(duration);

Serial.print(inches);

Serial.print("in, ");

Serial.print(cm);

Serial.print("cm");

Serial.println();

delay(100);

// Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different

// pieces of information in a channel. Here, we write to field 1.

int x = ThingSpeak.writeField(myChannelNumber, 1, cm, myWriteAPIKey);

if(x == 200){

Serial.println("Channel update successful.");

}

else{

Serial.println("Problem updating channel. HTTP error code " + String(x));

}

// change the value

delay(15000); // Wait 20 seconds to update the channel again

}

long microsecondsToInches(long microseconds) {

return microseconds / 74 / 2;

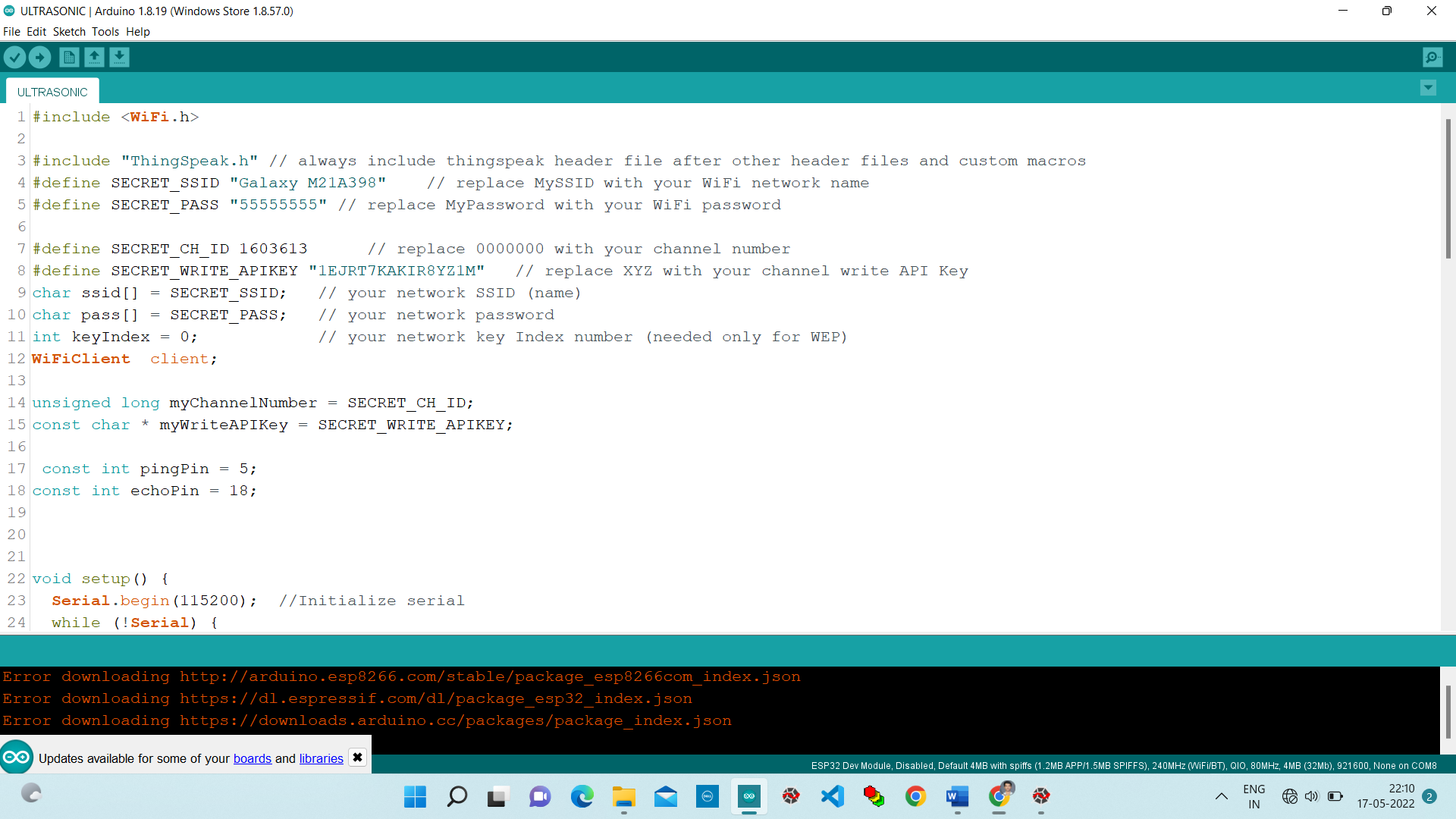
}

long microsecondsToCentimeters(long microseconds) {

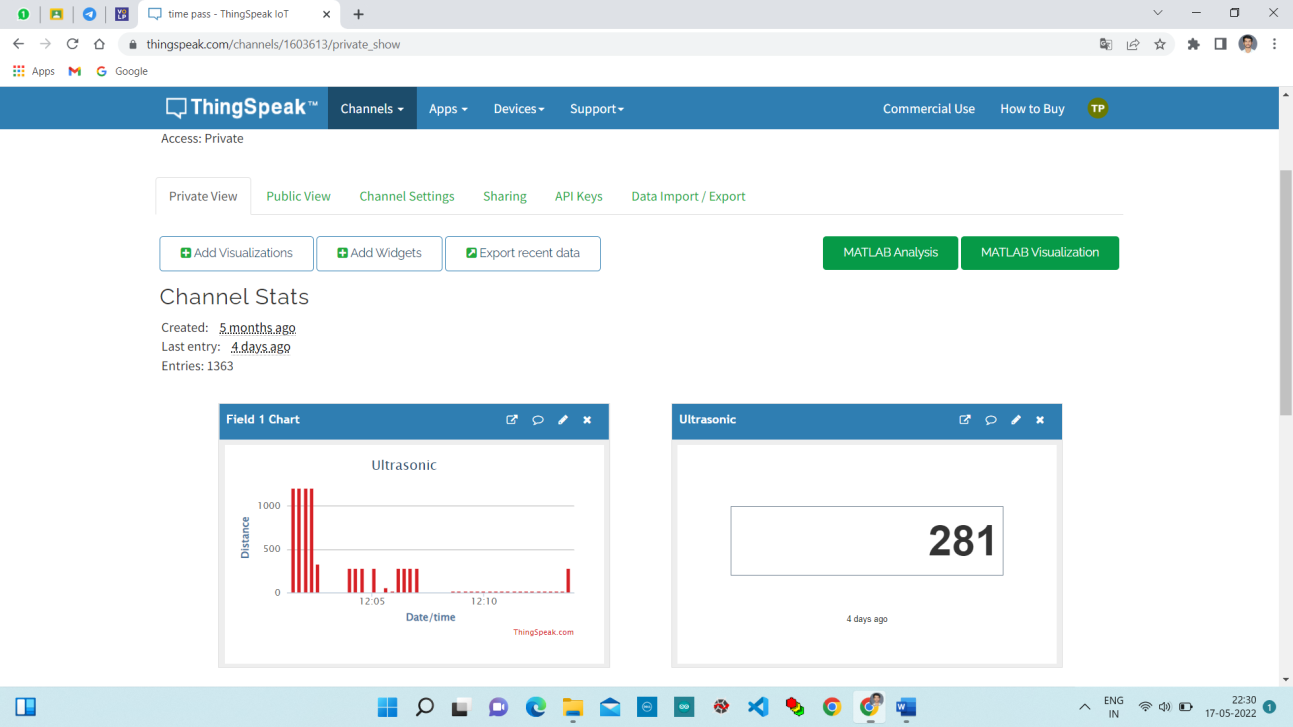
return microseconds / 29 / 2;

}

**Actual Code in Arduino IDE:**



**Thingspeak Result:**



**Conclusion:**

In this lab we studied about sensor interfacing with ESP32 and sends this signal to thingseak through cloud and observe the readings on it.