IOT ML BOOTCAMP DAY-4

Experiment 17:

Aim : program to detect the alcohol content?

Description:

->Connect the D0 to the digital and A0 to the analog A0 and then ground to the ground in the node and VCC to the VIN.

#define pm(a,b) pinMode(a,b)

#define dw(a,b) digitalWrite(a,b)

#define dr(a) digitalRead(a)

#define MQ2pin (0)

int digi = D2;

int ana = A0;

void setup()

{

//pm(digi,OUTPUT);

//pm(ana,OUTPUT);

Serial.begin(9600);

}

void loop()

{

float sensorval = analogRead(MQ2pin);//analog read

Serial.print("sensor val : ");

Serial.println(sensorval);

if(sensorval > 500)

Serial.print("Alcohol detected!!");

else

Serial.print("Alcohol not detected..");

delay(2000);

}

Experiment :18

Aim : program for moisture or soil sensor?

->connect the soil sensor to the flying fish..

#define pm(a,b) pinMode(a,b)

#define dw(a,b) digitalWrite(a,b)

#define dr(a) digitalRead(a)

int so = D0;

void setup()

{

pm(so,INPUT);

Serial.begin(9600);

}

void loop()

{

int m = dr(so);

//delay(500);

Serial.println(m);

if(m==1)

Serial.println("Dry soil..");

else

Serial.println("wet soil..");

delay(2000);

}

Example 19:

Aim: program to smoke sensor?

Experiment:20

Aim :

THING SPEAK:

* ->IOT is a 4D technology.

1.Data collection

2.Data storage

3.Data visualization

4.

Channel name :IOTdhtData

Experiment 21:

Aim : Program for wifi connectivity and to the ThingSpeak connectivity?

Code::::

#include "ESP8266WiFi.h"

const char\* SSID = "Harley";

const char\* PWD = "harley143";

void setup()

{

Serial.begin(9600);

WiFi.mode(WIFI\_STA); //wifi should be in station mode to connect to the network

WiFi.disconnect();

Serial.println("Connecting to :");

Serial.println(SSID);;

WiFi.begin(SSID,PWD);

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

{

Serial.println(".");

delay()

}

Serial.println("WiFi connected.");

}

TOTAL CODE TO UPLOAD INTO THE CLOUD….

#include "ESP8266WiFi.h" // connecting to wifi

#include "DHT.h" //readib the sensor data

#include "ThingSpeak.h" //connecting to the thing speak

WiFiClient client; //used to call the server

DHT dht(D2,DHT11);

const char\* SSID = "Harley";

const char\* PWD = "harley143";

void setup()

{

Serial.begin(9600);

dht.begin();

WiFi.mode(WIFI\_STA); //wifi should be in station mode to connect to the network

WiFi.disconnect();

Serial.println("Connecting to :");

Serial.println(SSID);;

WiFi.begin(SSID,PWD);

while(WiFi.status()!=WL\_CONNECTED) //checks wheather the wifi is connected of not.

{

Serial.println(".");

delay()

}

Serial.println("WiFi connected.");

ThingSpeak.begin(client); //it begins the client to the thingspeak

}

void loop()

{

float h = dht.readHumidity();

float t = dht.readTemperature();

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

{

return;

}

Serial.print("h : ");

Serial.print(h);

Serial.print("T : ");

Serial.println(t);

ThingSpeak.setField(1,h);

ThingSpeak.setField(2,t);

int s = ThingSpeak.writeFields(1575380,"JXSPNPVSNDAYDUNM");

if(s == 200)

Serial.println("Data uploaded.");

else

Serial.println("Not Sent.")

delay(20000);//minimum 2 seconds should be given to upload the data into the cloud....

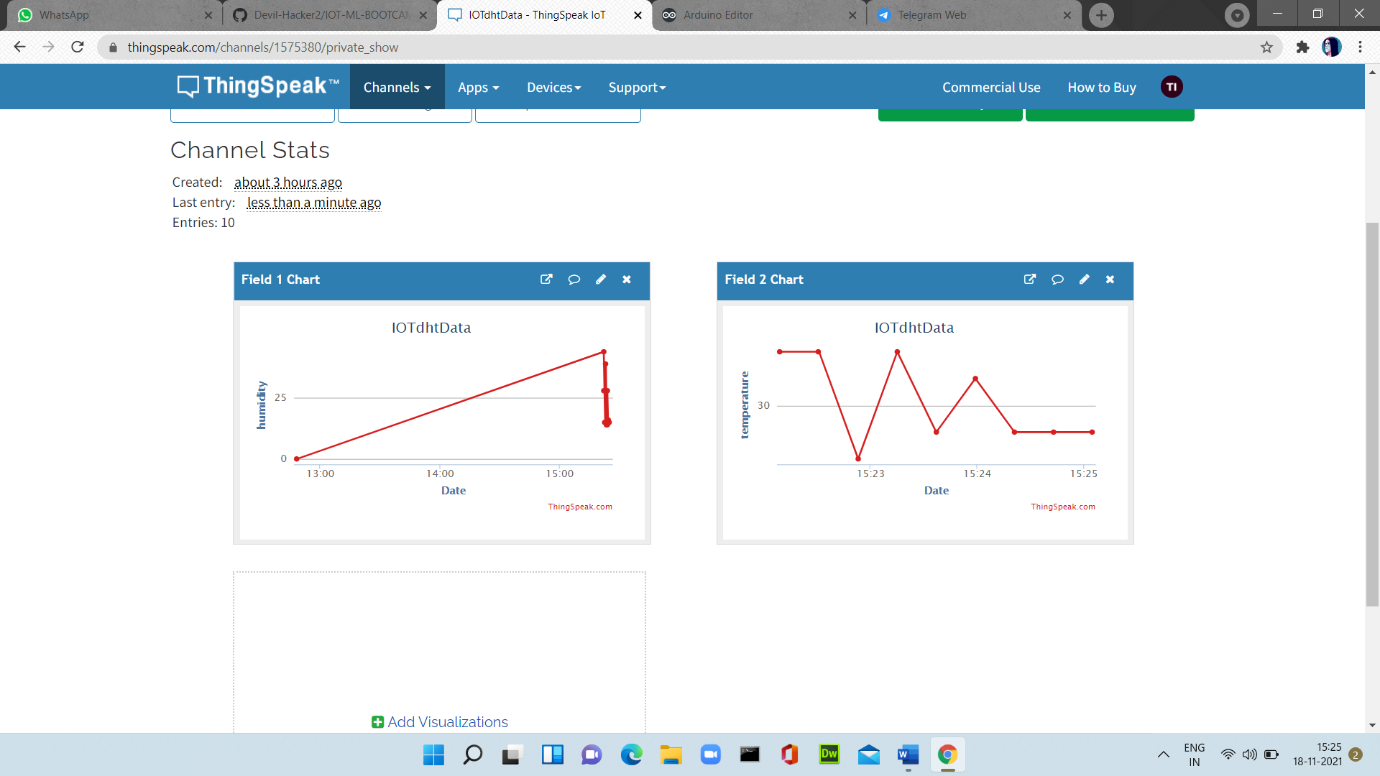
}

this is used to upload the data and analyse the data in the cloud.

->ThingSpeak has built in class in C++ i.e “ThingSpeak.h”

->By connecting the HDT sensor to the MCU and the embedding the above code into NodeMCU ,we can upload the data into the cloud.

->connect VCC of DHT sensor to VIN of the MCU,DATA pin to the D2 in MCU.



>MQTT PROTOCOL

* MQTT – message queue telemetry transport.
* It is an IOT protocol.
* It is used in communication b/w sender and receiver.
* Open the google colab to write the code for mqtt.
* Pip install paho-mqtt – it is an open source of the eclipe open source project.
* CODE to send the request to the receiver:
* import paho.mqtt.client as mqtt
* client = mqtt.Client() #client called
* while True:
* client.connect("broker.hivemq.com",1883) #it connects to the client through the broker using the default port number
* print('Broker connected')
* k = input('Enter msg:')
* client.publish('kits/iot',k) #it the name of the reciever to send the msg.