1. **LED BLINKING:**

void setup() {

// put your setup code here, to run once:

pinMode(5,OUTPUT);

pinMode(16,OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(5,HIGH);

delay(1000);

digitalWrite(5,LOW);

delay(1000);

digitalWrite(16,HIGH);

delay(1000);

digitalWrite(16,LOW);

delay(1000);

}

1. **Ultra Sonic Sensor**

#include "NewPing.h"

#define TRIGGER\_PIN 16

#define ECHO\_PIN 17

// Maximum distance we want to ping for (in centimeters).

#define MAX\_DISTANCE 400

// NewPing setup of pins and maximum distance.

NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANCE);

float duration, distance;

void setup()

{

Serial.begin(115200);

}

void loop()

{

// Send ping, get distance in cm

distance = sonar.ping\_cm();

// Send results to Serial Monitor

Serial.print("Distance = ");

Serial.print(distance);

if (distance >= 400 || distance <= 2)

{

Serial.println("Out of range");

}

else

{

Serial.print(distance);

Serial.println(" cm");

}

delay(1000);

}

1. **DHT11 on Serial Monitor:**

#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht (DHTPIN, DHTTYPE);

float h,t;

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

}

void loop() {

// put your main code here, to run repeatedly:

h = dht.readHumidity();

t = dht.readTemperature();

Serial.print("temperature:");

Serial.println(t);

Serial.print("Humidity:");

Serial.println(h);

delay(10000);

}

1. **Bluetooth on and off LED**

#include "BluetoothSerial.h"

#if !defined(CONFIG\_BT\_ENABLED) || !defined(CONFIG\_BLUEDROID\_ENABLED)

#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it

#endif

BluetoothSerial SerialBT;

String state;

void setup() {

pinMode(16, OUTPUT);

Serial.begin(115200);

SerialBT.begin("bluetooth practice"); //Bluetooth device name

Serial.println("The device started, now you can pair it with bluetooth!");

}

void loop() {

if (Serial.available())

{

SerialBT.write(Serial.read());

Serial.println("hello");

}

if (SerialBT.available()) {

state=SerialBT.read();

Serial.print("State :");

Serial.println(state);

if (state.equals("53")) {

digitalWrite(16, HIGH);

Serial.println("Light On");

}

// if the state is 'LED1OFF' the led1 will turn off

else if (state.equals("54")){

digitalWrite(16, LOW);

Serial.println("Light Off");

}

}

state="";

//delay(1000);

}

1. **LED On/Off Wifi with app**

#include <ThingSpeak.h>

#include <WiFi.h>

WiFiClient client;

const char\* ssid = "2nd Floor";//Enter the ssid of your router

const char\* password = "Sai56789";//Enter the password of your router

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

const char\* privateKey = "YRN193XOF406W7NP";//read key

const char\* privateKey1 = "NG4SL8DBCEZDBSPS";//write key

void setup() {

Serial.begin(115200);

pinMode(4, OUTPUT);//setting led as output

//pinMode(fan, OUTPUT);//setting led as output

//dht.begin();

ThingSpeak.begin(client);

delay(10);

// We start by connecting to a WiFi network

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void loop() {

//float h= ThingSpeak.readFloatField( 626040,2);

int d= ThingSpeak.readIntField( 1313948,1);

Serial.print(d);

if(d==1)

{

digitalWrite(4,HIGH);

Serial.print("LED ON");

Serial.println("");

}

if(d==0)

{digitalWrite(4,LOW);

Serial.print("LED OFF");

Serial.println("");

}

delay(5000);

}

1. **DHT11 data retrieval from thingspeak to app**

#include <WiFi.h>

#include "DHT.h"

#define DHTPIN 15 // what pin we're connected to

#define DHTTYPE DHT11 // define type of sensor DHT 11

DHT dht (DHTPIN, DHTTYPE);

const char\* ssid = "Lucky";//Enter the ssid of your router

const char\* password = "viniluckyram@123";//Enter the password of your router

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

const char\* privateKey = "QQNJ0Q6HAZEUEJAM";//read key

const char\* privateKey1 = "0TWPYXHW92CG2FTO";//write key

String line,line1;

float h,t;

void setup() {

Serial.begin(115200);

pinMode(DHTPIN, OUTPUT);//setting led as output

//pinMode(fan, OUTPUT);//setting led as output

dht.begin();

delay(10);

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void loop(){

h = dht.readHumidity();

t = dht.readTemperature();

Serial.print("temperature:");

Serial.println(t);

Serial.print("Humidity:");

Serial.println(h);

delay(10000);

retrieve\_from\_Cloud();

delay(10000);

}

void retrieve\_from\_Cloud(){

Serial.print("connecting to ");

Serial.println(host); // Use WiFiClient class to create TCP connections

WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

Serial.println("connection failed");

return;

}// We now create a URI for the request

String url = "/update";

url += "?api\_key=";

url += privateKey1;

url += "&field1=";

url += t;

url += "&field2=";

url += h;

Serial.print("Requesting URL: ");

Serial.println(url);

// This will send the request to the server

client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

delay(1000);

// Read all the lines of the reply from server and print them to Serial

while(client.available()){

String line1 = client.readStringUntil('\r');

Serial.print(line1);

}

Serial.println("closing connection");

}

1. **DHT11 data upload to thingspeak**

#include <WiFi.h>

#include "DHT.h"

#define DHTPIN 15 // what pin we're connected to

#define DHTTYPE DHT11 // define type of sensor DHT 11

DHT dht (DHTPIN, DHTTYPE);

const char\* ssid = "Lucky";//Enter the ssid of your router VSES

const char\* password = "viniluckyram@123";//Enter the password of your router gnir33nignEtr@mS

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

//const char\* privateKey = "AUCY5TZ02GDI5POO";//read key

const char\* privateKey1 = "9VZL995DRHB3A4NT";//write key

//String line,line1;

float h,t;

void setup() {

Serial.begin(115200);

dht.begin();

delay(10);

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void loop()

{

h = dht.readHumidity();

t = dht.readTemperature();

Serial.print("temperature:");

Serial.println(t);

Serial.print("Humidity:");

Serial.println(h);

upload();

delay(10000);

}

void upload()

{

Serial.print("connecting to ");

Serial.println(host);

// Use WiFiClient class to create TCP connections

WiFiClient client;

const int httpPort = 80;

if (!client.connect(host, httpPort)) {

Serial.println("connection failed");

return;

}

// We now create a URI for the request

String url = "/update";

url += "?api\_key=";

url += privateKey1;

url += "&field1=";

url += t;

url += "&field2=";

url += h;

Serial.print("Requesting URL: ");

Serial.println(url);

// This will send the request to the server

client.print(String("GET ") + url + " HTTP/1.1\r\n" +

"Host: " + host + "\r\n" +

"Connection: close\r\n\r\n");

delay(1000);

// Read all the lines of the reply from server and print them to Serial

while(client.available())

{

String line1 = client.readStringUntil('\r');

Serial.print(line1);

}

Serial.println();

Serial.println("closing connection");

}