

ProjectDevelopment-DeliveryOfSprint-2

TeamID	PNT2022TMID19077
ProjectName	IOTbasedsafetygadget forchildsafetymonitoringa nd notification
Date	14-11-2022

NOTIFICATION:

This coding will make connection between IoT Device & Parent's application. When the child cross across the geofence message will be notified on parent's application.

Coding:

```
#include<WiFi.h>//library for
```

```
wifi#include<PubSubClient.h>//libraryfo
```

```
rMQTT
```

```
voidcallback(char*subscribetopic,byte*payload,unsignedintpayloadlength);
```

```
//-----credentialsofIBMAccount-----
```

```
#defineORG"frpi8s"// IBMORGANIZATIONID
```

```
#defineDEVICE_TYPE"NodeMCU"//DEVICETYPEMENTIONEDINIOTW  
ATSON
```

```
PLATFORM#defineDEVICE_ID"12345"//DEVICEIDMENTIONEDINIOTWATSON  
PLATEFORM
```

```
#defineTOKEN"12345678"//TokenStringdata3;floatd
```

```
ist;
```

```
//-----customizetheabovevalue-----
```

```

char server[] = "ORG".messaging.internetofthings.ibmcloud.com"; // server name

char publishTopic[] = "ultrasonic/evt/Data/fmt/json"; // topic name and type of event performed and format in which data to be send */

char subscribeTopic[] = "ultrasonic/cmd/test/fmt/String"; // command to represent command type and command in test of format string */

char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:"ORG":DEVICE_TYPE":DEVICE_ID"; // client ID
//
WiFiClient wifiClient; // creating an instance for wifi client
PubSubClient client(server, 1883, callback, wifiClient); // calling the predefined client ID by passing parameter like server ID, port and wifi credential */
int LED = 4;
int trig = 5;
int echo = 18;
void setup() {
  Serial.begin(115200);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(LED, OUTPUT);
  delay(10);
  Serial.println();
  wifiConnect();
  mqttConnect();

  void loop() {
    digitalWrite(trig, LOW);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    float dur = pulseIn(echo, HIGH);
    float dist = (dur * 0.0343) / 2;
    Serial.print("distance in cm");
    Serial.println(dist);
    PublishData(dist);
    delay(1000);
    if (!client.loop()) {
      mqttConnect();
    }
  }
  /* ..... retrieving to cloud ..... */
  void PublishData(float dist) {
    mqttConnect(); // function call for connecting to IBM
    /* creating the string in form of JSON to update the data to IBM cloud */
    String object;
  }

```

```

        if(dist<100)
        {
            digitalWrite(LED,HIGH);Serial.println("noobjectisnear");object="Near";
        }
        else
        {
            digitalWrite(LED,LOW);Serial.println("noobjectfound");object="No";
        }
        String payload="{\"distance\":"; payload
        +=dist;payload+=",\"\"object\":\":";payload+=object;payload+="\"}";
        Serial.print("Sendingpayload:");
        Serial.println(payload);

        if(client.publish(publishtopic,(char*)payload.c_str())){
            Serial.println("Publish ok");/* if its sucessfully upload data
on thecloud then it will print publish ok in serial monitor or else it will
print publishfailed*/
        }else{
            Serial.println("Publishfailed");
        }
    }
    voidmqttconnect(){if(!client.connected()){

        Serial.print("Reconnectingclientto");Serial.println(server);while(!client.connect(clientid,authMethod,
token)){Serial.print(".");delay(500);
        }

        initManagedDevice();
        Serial.println();
    }
    voidwificonnect();//functiondefenitionforwificonnect
    {
        Serial.println();Serial.print("Connectingto");

```

```

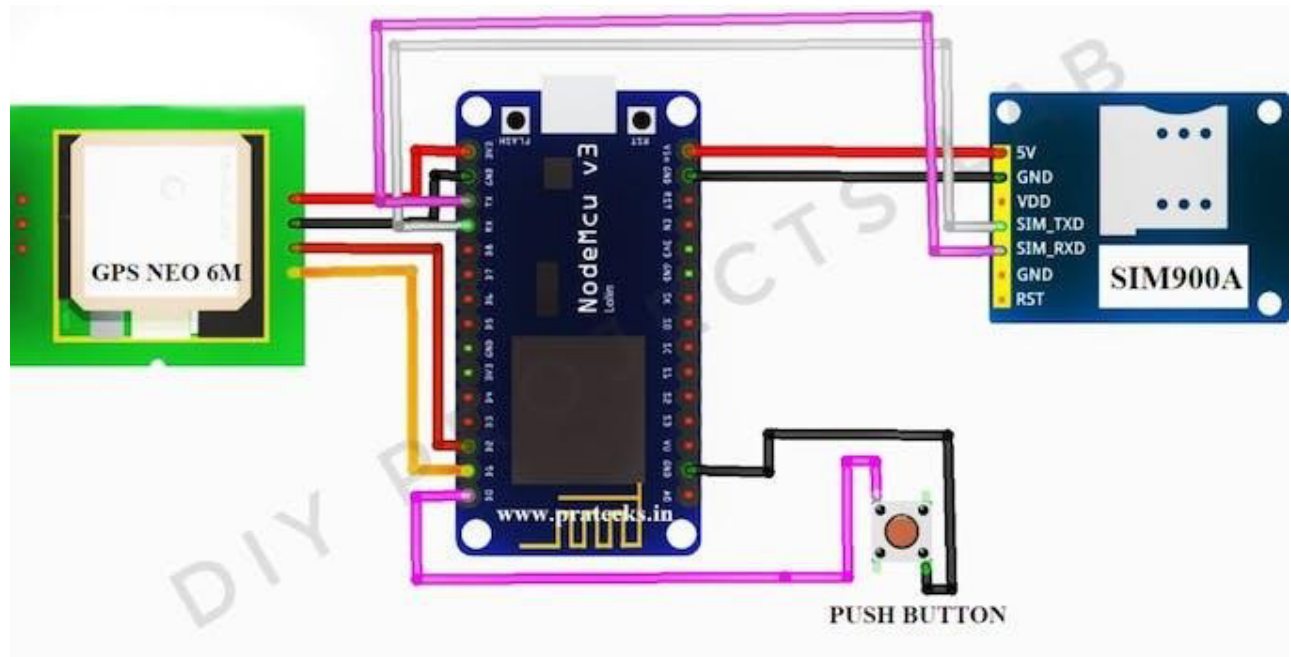
WiFi.begin("vivo 1816", "taetae95",6);//PASSING THE WIFI
CREDENTIALS TOESTABLISHCONNECTION
while(WiFi.status()!=WL_CONNECTED){delay(500);
    Serial.print(".");
}
    Serial.println("");Serial.println("WiFiconnected");Serial.println("IP
address");
    Serial.println(WiFi.localIP());
}
voidinitManagedDevice(){if(client.subscribe(subscribetopic)){
    Serial.println((subscribetopic));Serial.println("subscribetocmdOK");
}else
    {    Serial.println("subscribetocmdfailed");

    }

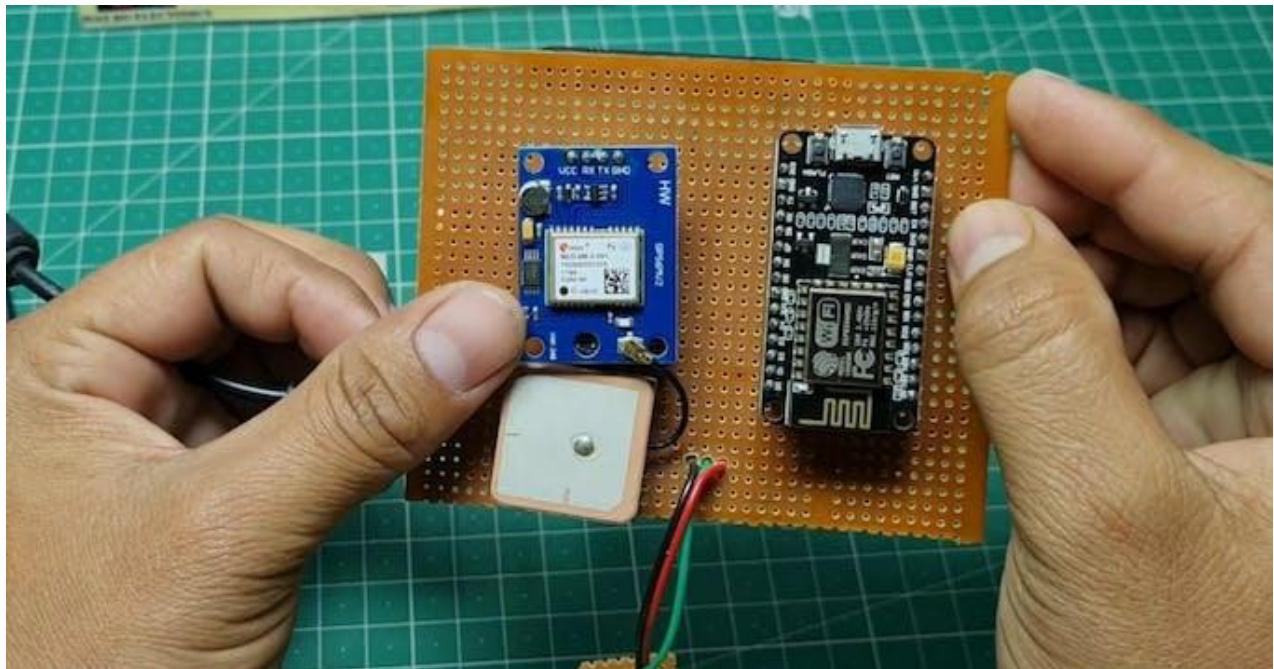
}
voidcallback(char*subscribetopic,byte*payload,unsignedintpayloadLength)
{
    Serial.print("callbackinvokedfortopic:");
    Serial.println(subscribetopic);for(inti=0;i<payloadLength;i++){
        //Serial.print((char)payload[i]);data3+=(char)payload[i];
    }
    //Serial.println("dta:"+data3);
    //if(data3=="Near")
    //{
    //Serial.println(data3);
    //digitalWrite(LED,HIGH);
    //}
    //else//{
    //Serial.println(data3);
    //digitalWrite(LED,LOW);//}data3="";
}

```

SCHEMATICDIAGRAM:



OUTPUT:



← Back

Device Drilldown - 12345

Device Credentials

Connection Information

Recent Events

State

Device Information

Metadata

Diagnostics

Connection Logs

Device Actions

Device Credentials

You registered your device to the organization. Add these credentials to the device to connect it to the platform. After the device is connected, you can navigate to view connection and event details.

Organization ID	frpi8s
Device Type	NodeMCU
Device ID	12345
Authentication Method	use-token-auth
Authentication Token	12345678

⚠

Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the device to generate a new authentication token.

Find out how to add these credentials to your device

➤

BrowseActionDevice TypesInterfaces

Add Device +

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
data	{"Warning":28.95}	json	a few seconds ago
data	{"Warning":28.95}	json	a few seconds ago
data	{"Warning":49.98}	json	a minute ago
data	{"Warning":49.98}	json	a minute ago
data	{"Warning":11.03}	json	a minute ago