**TRẢ LỜI TỰ ĐỘNG**

**I/Mô tả dự án:**

Tạo ra một thiết bị gồm esp32 và module sim 4g trả lời tự động thông qua MQTT (sử dụng node js (node red))

A diagram of a phone call

Description automatically generated

[Board - The research and development of a fire monitoring - Jira (atlassian.net)](https://firealert.atlassian.net/jira/core/projects/RDFM/board?atlOrigin=eyJpIjoiOGE0MTZlYWI1ZWVkNGRiMTgyMzU2NDkwNDhmNDgxOWMiLCJwIjoiaiJ9&selectedIssue=RDFM-5) (link phần mềm Hercules chứa các lệnh trong module sim)

**II/ Giải thích rõ từng bước**

* Tạo một tag node red user và intern \* topic Aler\_service
* Tạo một tag node red esp32 \* topic Aler\_service
* Tải thư viện [GitHub - adafruit/Adafruit\_MQTT\_Library: Arduino library for MQTT support](https://github.com/adafruit/Adafruit_MQTT_Library) cấu hình cho esp bằng 2 lệnh SUBSCRIBER và PUBLISH để lấy dữ liệu cũng như gửi dữ liệu từ sever xuống
* Sử dụng uart để giao tiếp với module simp thực hiện gọi cũng như nhắn tin [Board - The research and development of a fire monitoring - Jira (atlassian.net)](https://firealert.atlassian.net/jira/core/projects/RDFM/board?atlOrigin=eyJpIjoiOGE0MTZlYWI1ZWVkNGRiMTgyMzU2NDkwNDhmNDgxOWMiLCJwIjoiaiJ9&selectedIssue=RDFM-5) (link phần mềm Hercules chứa các lệnh trong module sim)

IV/ Cài đặt phần mềm và môi trường

* Tải node js và sau đó tải npm sau đó tải node red

**BÁO CÁO KẾT QUẢ NGHIÊN CỨU**

**I.Sơ đồ MQTT**

A screenshot of a computer

Description automatically generated

CTRL + z =

**II.Các thư viện sử dụng**

#include <ArduinoJson.h>



#include <ArduinoJson.hpp>

#include <PubSubClient.h>

#include<WiFi.h> 

#include <SoftwareSerial.h>

**II.Code mẫu :**

#include <ArduinoJson.h>

#include <ArduinoJson.hpp>

#include <PubSubClient.h>

#include<WiFi.h>

#include <SoftwareSerial.h>

int dem=0;

int count=0;

String msg;

// RX 16

// TX 17

// khởi tạo  SoftwareSerial

SoftwareSerial myserial(16,17);

// cấu hình wifi

const char\* ssid = "Thinh";

const char\* pass = "01052005T";

// Cấu hình MQTT Server

const char\* mqttsever = "mqtt.cusc.vn";

const int mqttport = 1883;

const char\* mqttuser = "fire\_alert";

const char\* mqttpass = "fire@123";

WiFiClient espClient;

PubSubClient client(espClient);

// cấu hình json

DynamicJsonDocument doc(1024);

// cài đặt wifi

void setup\_wifi(){

  Serial.print("conecting to ");

  Serial.println(ssid);

  WiFi.begin(ssid,pass);

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

    delay(500);

    Serial.print('.');

  }

  Serial.println("");

  Serial.println("Wifi conected");

}

// kết nối lại mqtt server

void reconnectmqtt(){

  while(!client.connected()){

    Serial.print("Attempting MQTT connection..");

    String clientID= "ESP32Client-";

    clientID += String(random(0xffff),HEX);

    if(client.connect(clientID.c\_str(),mqttuser,mqttpass)){

      Serial.println("connected");

      client.subscribe("Alert\_Service");

    }

    else{

      Serial.print("failed, rc=");

      Serial.print(client.state());

      Serial.println(" try again in 5 seconds");

      delay(5000);

    }

  }

}

// nhận message từ mqtt server

void callbackmqtt(char\* topic, byte\* message, unsigned int length){

  for(int i =0;i<length;i++){

    msg+=(char)message[i];

  }

}

// gọi số điện thoại

void callsim(String number){

  switch(dem){

    case 0:

      myserial.println("AT");

      while(myserial.read()!=65){

        delay(500);

      }

      Serial.println("OK");

      Serial.println(dem);

      dem++;

      break;

    delay(1000);

    case 1:

      myserial.println("AT+CREG?");

      delay(1000);

     while(myserial.read()!=65){

        delay(500);

      }

      Serial.println("OK");

      Serial.println(dem);

      dem++;

      delay(1000);

      break;

    case 2:

      myserial.println("AT+CGREG?");

      delay(1000);

      while(myserial.read()!=65){

        delay(500);

      }

      Serial.println("OK");

      Serial.println(dem);

      dem++;

      delay(1000);

      break;

    case 3:

      myserial.print("ATD");myserial.print(number);myserial.println(';');

      while(myserial.read()!=65){

        delay(500);

      }

      Serial.println("OK");

      Serial.println(dem);

      delay(5000);

      dem=0;

      Serial.println("\n");

      break;

  }

}

// gửi tin nhắn

void sendsim(String number, String send){

  switch(dem){

    case 0:

      myserial.println("AT");

      while(myserial.read()!=65){

        delay(500);

      }

      Serial.println("OK");

      Serial.println(dem);

      dem++;

      break;

    delay(1000);

    case 1:

      myserial.println("AT+CMGF=1");

      delay(1000);

     while(myserial.read()!=65){

        delay(500);

      }

      Serial.println("OK");

      Serial.println(dem);

      dem++;

      delay(1000);

      break;

    case 2:

      myserial.println("AT+CMGS=\"0839488362\"");

      delay(1000);

      myserial.println();

      while(myserial.read()!=65){

        delay(500);

      }

      Serial.println("OK");

      Serial.println(dem);

      dem=0;

      myserial.print("helloworld");

      delay(500);

      myserial.print("");

      delay(1000);

      msg="";

      break;

    }

  }

void setup(){

  Serial.begin(9600);

  myserial.begin(9600); // khởi tạo softwareSerial

  setup\_wifi(); // khởi tạo wifi

  client.setServer(mqttsever,mqttport);// khởi tạo thông tin server

  client.setCallback(callbackmqtt);// nhận tin nhắn từ server

}

void loop(){

  if (!client.connected()) {

    reconnectmqtt();

  }

  client.loop();

  deserializeJson(doc,msg); // thêm msg vào json doc

   if(msg!=""){

    callsim(doc["call"].as<String>()); // gọi

    sendsim(doc["call"].as<String>(),doc["message"].as<String>()); // nhắn tin

  }

}