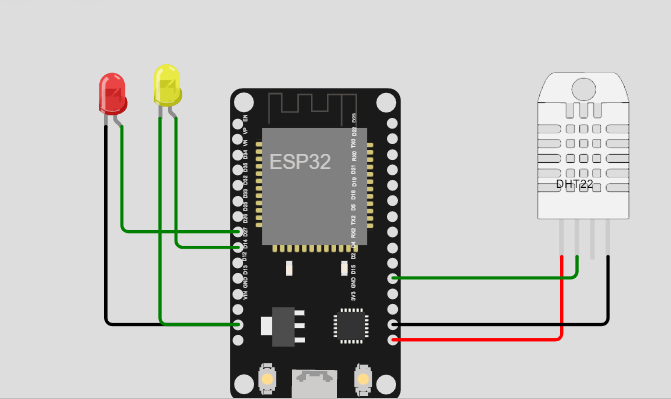
**1. Truyền nhận dữ liệu DHT**



#include <WiFi.h>

#include "PubSubClient.h"

#include <DHT.h>

#define DHT\_PIN 4

#define DHT\_TYPE DHT22

DHT dht(DHT\_PIN,DHT\_TYPE);

const char \* MQTTServer = "broker.hivemq.com";

const char \* MQTT\_Topic = "honghaihehe";

// Tạo ID ngẫu nhiên tại: https://www.guidgen.com/

const char \* MQTT\_ID = "honghai123";

int Port = 1883;

WiFiClient espClient;

PubSubClient client(espClient);

const int ledPin = 25;

const int ledPin2 = 26;

void WIFIConnect() {

**Serial**.println("Connecting to SSID: Wokwi-GUEST");

  WiFi.begin("Wokwi-GUEST", "");

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

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("");

**Serial**.print("WiFi connected");

**Serial**.print(", IP address: ");

**Serial**.println(WiFi.localIP());

}

void MQTT\_Reconnect() {

  while (!client.connected()) {

    if (client.connect(MQTT\_ID)) {

**Serial**.print("MQTT Topic: ");

**Serial**.print(MQTT\_Topic);

**Serial**.print(" connected");

      client.subscribe(MQTT\_Topic);

**Serial**.println("");

    } else {

**Serial**.print("failed, rc=");

**Serial**.print(client.state());

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

      delay(5000);

    }

  }

}

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

**Serial**.print("Message arrived on topic: ");

**Serial**.println(topic);

**Serial**.print("Message: ");

  String stMessage;

  for (int i = 0; i < length; i++) {

**Serial**.print((char)message[i]);

    stMessage += (char)message[i];

  }

**Serial**.println();

  if (stMessage == "on") {

    digitalWrite(ledPin, HIGH);

  }

  else if (stMessage == "off") {

    digitalWrite(ledPin, LOW);

  }

  if (stMessage == "on2") {

    digitalWrite(ledPin2, HIGH);

  }

  else if (stMessage == "off2") {

    digitalWrite(ledPin2, LOW);

  }

}

void setup() {

**Serial**.begin(115200);

  dht.begin();

  WIFIConnect();

  client.setServer(MQTTServer, Port);

  client.setCallback(callback);

  pinMode(ledPin, OUTPUT);

  pinMode(ledPin2, OUTPUT);

}

void loop() {

  delay(500);

  int nhietdo=dht.readTemperature();

  int doam=dht.readHumidity();

  char chuoidoam[8];

  dtostrf(doam,1,2,chuoidoam);

  char chuoithoitiet[8];

  dtostrf(nhietdo,1,2,chuoithoitiet);

**Serial**.println(chuoithoitiet);

**Serial**.println(chuoidoam);

  client.publish("honghai/nd",chuoithoitiet);

  client.publish("honghai/da",chuoidoam);

  if (!client.connected()) {

    MQTT\_Reconnect();

  }

  client.loop();

}

WEB

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js" type="text/javascript"></script>

    <script src="https://code.jquery.com/jquery-3.6.1.min.js" type="text/javascript"></script>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="container">

        <textarea name="text1" id="text1" cols="30" rows="10"></textarea>

        <div style="position: relative;">

            <button id="guitext">GỬI DỮ LIỆU</button>

        </div>

    </div>

    <button class="bat">Bật LED</button>

    <button class="tat">Tắt LED</button>

    <h1><strong>NHIỆT ĐỘ HIỆN TẠI: <span id="nhietdo"></span> ĐỘ C</strong></h1>

        <h2>Khi nhiệt độ lớn hơn <input type="number" value="35" id="t1" min="0" max="80"> thì bật đèn led 1</h2>

        <h1><strong>ĐỘ ẨM HIỆN TẠI: <span id="doam"></span> %</strong></h1>

        <h2>Khi độ ẩm lớn hơn <input type="number" value="20" id="t2" min="0" max="80"> thì bật đèn led 2</h2>

</body>

</html>

<script>

    // $('.bat').click(function(){

    //     message = new Paho.MQTT.Message("on");

    //     message.destinationName = "trungsonpro";

    //     client.send(message);

    // });

    // $('.tat').click(function(){

    //     message = new Paho.MQTT.Message("off");

    //     message.destinationName = "trungsonpro";

    //     client.send(message);

    // });

    // $("#guitext").click(function(){

    //     var matkhau=$("#text1").val();

    //     console.log(matkhau)

    //     if(matkhau=="123456"){

    //         message = new Paho.MQTT.Message("on");

    //         console.log(matkhau)

    //         message.destinationName = "trungsonpro";

    //         client.send(message)

    //     }else{

    //         message = new Paho.MQTT.Message("off");

    //         console.log(matkhau)

    //         message.destinationName = "trungsonpro";

    //         client.send(message)

    //     }

    // })

    client = new Paho.MQTT.Client("broker.hivemq.com", *Number*(8000), "honghaihehe");

    client.onConnectionLost = onConnectionLost;

    client.onMessageArrived = onMessageArrived;

    client.connect({onSuccess:onConnect});

*function* onConnect() {

    console.log("onConnect");

    client.subscribe("honghai/nd");

    client.subscribe("honghai/da");

    }

*function* onConnectionLost(*responseObject*) {

    if (*responseObject*.errorCode !== 0) {

        console.log("onConnectionLost:"+*responseObject*.errorMessage);

    }

    }

*function* onMessageArrived(*message*) {

        console.log(*message*.payloadString);

*var* so=parseInt($("#t1").val())

        if(*message*.destinationName=="honghai/nd"){

*var* nhietdo=parseInt(*message*.payloadString);

            $("#nhietdo").text(nhietdo);

*var* nhietdo=parseInt(*message*.payloadString);

            if(nhietdo>so){

*message* = new Paho.MQTT.Message("on");

*message*.destinationName = "honghaihehe";

                client.send(*message*);

            }else{

*message* = new Paho.MQTT.Message("off");

*message*.destinationName = "honghaihehe";

                client.send(*message*);

            }

        }

*var* so2=parseInt($("#t2").val())

        if(*message*.destinationName=="honghai/da"){

*var* doam=parseInt(*message*.payloadString);

            $("#doam").text(doam);

            if(doam>so2){

*message* = new Paho.MQTT.Message("on2");

*message*.destinationName = "honghaihehe";

                client.send(*message*);

            }else{

*message* = new Paho.MQTT.Message("off2");

*message*.destinationName = "honghaihehe";

                client.send(*message*);

            }

        }

    }

</script>

**2.Keypad**

#include <Keypad.h>

#include <WiFi.h>

#include <WiFi.h>

#include "PubSubClient.h"

const uint8\_t ROWS = 4;

const uint8\_t COLS = 4;

String pass="";

char keys[ROWS][COLS] = {

  { '1', '2', '3', 'A' },

  { '4', '5', '6', 'B' },

  { '7', '8', '9', 'C' },

  { '\*', '0', '#', 'D' }

};

uint8\_t colPins[COLS] = { 16, 4, 2, 15 };

uint8\_t rowPins[ROWS] = { 19, 18, 5, 17 };

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

const char \* MQTTServer = "broker.emqx.io";

const char \* MQTT\_Topic = "vlute\_pass";

// Tạo ID ngẫu nhiên tại: https://www.guidgen.com/

const char \* MQTT\_ID = "4aceedc2-ae01-4091-b4b3-9db9bd4c8703";

int Port = 1883;

WiFiClient espClient;

PubSubClient client(espClient);

const int ledPin = 12;

void WIFIConnect() {

**Serial**.println("Connecting to SSID: Wokwi-GUEST");

  WiFi.begin("Wokwi-GUEST", "");

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

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("");

**Serial**.print("WiFi connected");

**Serial**.print(", IP address: ");

**Serial**.println(WiFi.localIP());

}

void MQTT\_Reconnect() {

  while (!client.connected()) {

    if (client.connect(MQTT\_ID)) {

**Serial**.print("MQTT Topic: ");

**Serial**.print(MQTT\_Topic);

**Serial**.print(" connected");

      client.subscribe("vlute\_trave");

**Serial**.println("");

    } else {

**Serial**.print("failed, rc=");

**Serial**.print(client.state());

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

      delay(5000);

    }

  }

}

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

**Serial**.print("Message arrived on topic: ");

**Serial**.println(topic);

**Serial**.print("Message: ");

  String stMessage;

  for (int i = 0; i < length; i++) {

**Serial**.print((char)message[i]);

    stMessage += (char)message[i];

  }

**Serial**.println();

  if (stMessage == "on") {

   digitalWrite(ledPin, HIGH);

  }

  else if (stMessage == "off") {

    digitalWrite(ledPin, LOW);

  }

}

void setup() {

**Serial**.begin(115200);

  WIFIConnect();

  client.setServer(MQTTServer, Port);

  client.setCallback(callback);

  pinMode(ledPin, OUTPUT);

}

void loop() {

  char key = keypad.getKey();

  client.loop();

    if (key) {

**Serial**.println(key);

     if(key != '#' ){

    pass += key ;

  }else{

**Serial**.print(pass);

   client.publish("vlute\_matkhau",String(pass).c\_str());

    pass ="";

  }

  }

  if (!client.connected()) {

    MQTT\_Reconnect();

  }

}

Web

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js" type="text/javascript"></script>

    <script src="https://code.jquery.com/jquery-3.6.1.min.js" type="text/javascript"></script>

</head>

<body>

    <h1  style="color: blue ">Ứng dụng thay đổi mật khẩu </h1>

    </h1><h1 id="passmd" style="color: red ">Mật khẩu hiện tại là : 012345</h1>

    <h1 style="color: red ;" id="textpass"> </h1>

   <input type="text" id="pass" >

   <input type="button" value="Đổi mật khẩu" onclick="doimk()">

</body>

</html>

<script>

*var* pass="012345";

    client = new Paho.MQTT.Client("broker.emqx.io", *Number*(8083), "vlute\_pass");

    client.onConnectionLost = onConnectionLost;

    client.onMessageArrived = onMessageArrived;

    client.connect({onSuccess:onConnect});

*function* onConnect() {

    console.log("onConnect");

    client.subscribe("vlute\_matkhau");

    }

*function* onConnectionLost(*responseObject*) {

    if (*responseObject*.errorCode !== 0) {

        console.log("onConnectionLost:"+*responseObject*.errorMessage);

    }

    }

*function* onMessageArrived(*message*) {

    console.log("onMessageArrived:"+*message*.payloadString);

    if(*message*.payloadString===pass){

        console.log("dung")

*message* = new Paho.MQTT.Message("dung roi");

*message*.destinationName = "vlute\_trave";

        client.send(*message*);

    }

    else{

        console.log("ko")

*message* = new Paho.MQTT.Message("sai roi");

*message*.destinationName = "vlute\_trave";

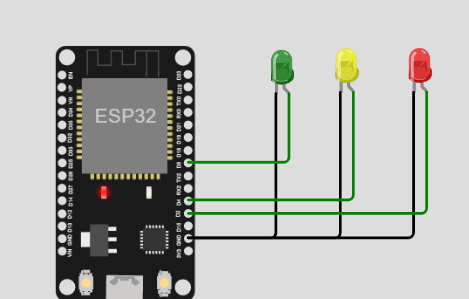
        client.send(*message*);

    }

    }

</script>

**3.Led**



#include <WiFi.h>

#include "PubSubClient.h"

const char \* MQTTServer = "broker.emqx.io";

const char \* MQTT\_Topic = "HD/den";

const char \* MQTT\_ID = "HD-kt";

int Port = 1883;

#define LED1 2

#define LED2 4

#define LED3 5

WiFiClient espClient;

PubSubClient client(espClient);

void WIFIConnect() {

**Serial**.println("Connecting to SSID: Wokwi-GUEST");

  WiFi.begin("Wokwi-GUEST", "");

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

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("");

**Serial**.print("WiFi connected");

**Serial**.print(", IP address: ");

**Serial**.println(WiFi.localIP());

}

void MQTT\_Reconnect() {

  while (!client.connected()) {

    if (client.connect(MQTT\_ID)) {

**Serial**.print("MQTT Topic: ");

**Serial**.print(MQTT\_Topic);

**Serial**.print(" connected");

      client.subscribe(MQTT\_Topic);

**Serial**.println("");

    } else {

**Serial**.print("failed, rc=");

**Serial**.print(client.state());

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

      delay(5000);

    }

  }

}

String stMessage;

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

**Serial**.print("Message arrived on topic: ");

**Serial**.println(topic);

**Serial**.print("Message: ");

  for (int i = 0; i < length; i++) {

    stMessage += (char)message[i];

  }

**Serial**.println(stMessage);

  digitalWrite(LED1, HIGH);

  delay(((int)stMessage.charAt(0))\*100);

  digitalWrite(LED1, LOW);

  digitalWrite(LED2, HIGH);

  delay(((int)stMessage.charAt(1))\*100);

  digitalWrite(LED2, LOW);

  digitalWrite(LED3, HIGH);

  delay(((int)stMessage.charAt(2))\*100);

  digitalWrite(LED3, LOW);

}

void setup() {

**Serial**.begin(115200);

  pinMode(LED1, OUTPUT);

  pinMode(LED2, OUTPUT);

  pinMode(LED3, OUTPUT);

  WIFIConnect();

  client.setServer(MQTTServer, Port);

  client.setCallback(callback);

}

void loop() {

  if (!client.connected()) {

    MQTT\_Reconnect();

  }

  client.loop();

  if(stMessage!=""){

    client.publish("HD/ok",String("Thao tác thành công. ESP đã nhận dữ liệu").c\_str());

  }

  else{

    client.publish("HD/thatbai",String("Thao tác thất bại").c\_str());

  }

  delay(10);

}

**Web**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js" type="text/javascript"></script>

    <script src="https://code.jquery.com/jquery-3.6.1.min.js" type="text/javascript"></script>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <label for="">Thời gian đèn xanh bật(Đơn vị là giây)</label><br>

    <input type="text" id="xanh"><br>

    <label for="">Thời gian đèn vàng bật(Đơn vị là giây)</label><br>

    <input type="text" id="vang"><br>

    <label for="">Thời gian đèn đỏ bật(Đơn vị là giây)</label><br>

    <input type="text" id="do"><br>

    <br>

    <h5 id="trave"></h5>

    <input type="button" value="Lưu" onclick="guidulieu()">

</body>

</html>

<script>

    // $('.bat').click(function(){

    //     message = new Paho.MQTT.Message("on");

    //     message.destinationName = "Vlute/led";

    //     client.send(message);

    // });

    // $('.tat').click(function(){

    //     message = new Paho.MQTT.Message("off");

    //     message.destinationName = "Vlute/led";

    //     client.send(message);

    // });

    client = new Paho.MQTT.Client("broker.emqx.io", *Number*(8083), "HD/den");

    client.onConnectionLost = onConnectionLost;

    client.onMessageArrived = onMessageArrived;

    client.connect({onSuccess:onConnect});

*function* onConnect() {

    console.log("onConnect");

    client.subscribe("HD/ok");

    client.subscribe("HD/thatbai");

    }

*function* onConnectionLost(*responseObject*) {

    if (*responseObject*.errorCode !== 0) {

        console.log("onConnectionLost:"+*responseObject*.errorMessage);

    }

    }

*function* onMessageArrived(*message*) {

    console.log("onMessageArrived:"+*message*.payloadString);

    if(*message*.destinationName=="HD/ok"){

        document.getElementById("trave").innerHTML=*message*.payloadString

    }

    if(*message*.destinationName=="HD/thatbai"){

        document.getElementById("trave").innerHTML=*message*.payloadString

    }

    }

*function* guidulieu(){

*var* denxanh=document.getElementById("xanh").value;

*var* denvang=document.getElementById("vang").value;

*var* dendo=document.getElementById("do").value;

*var* data=denxanh+denvang+dendo;

        console.log(data);

        message = new Paho.MQTT.Message(data);

        message.destinationName = "HD/den";

        client.send(message);

        document.getElementById("xanh").innerHTML=""

        document.getElementById("vang").innerHTML=""

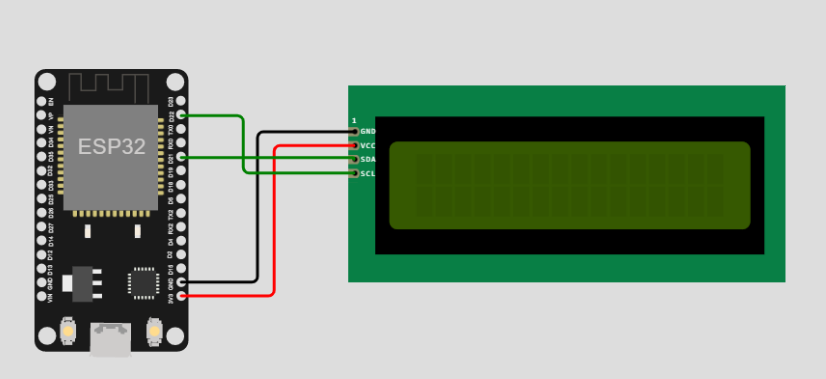
        document.getElementById("do").innerHTML=""

    }

</script>

esp32 và js ta đặt địa chỉ mqttserver là: broker.hivemq.com.Địa chỉ subscribe esp32 là ‘trungsonpro’, còn js là ‘trungson/nd’. Khi bắt đầu chạy esp32 thì dữ liệu sẽ publish đến server broker.hivemq.com và tìm địa chỉ subscribe là ‘trungson/nd’ để gửi dữ liệu. khi nhận data, js nhận in lên màn hình và so sánh data nhiệt độ đó có lớn hơn giá trị cài đặt, nếu lớn hơn hoặc khác thì js sẽ gửi dữ liệu đến địa chỉ ‘trungsonpro’, lúc đó hàm callback ở esp32 sẽ nhận được data và xét điều kiện đã xét để bật/tắt đèn.

**4.LCD**



#include <WiFi.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <WiFi.h>

#include "PubSubClient.h"

LiquidCrystal\_I2C lcd = LiquidCrystal\_I2C(0x27, 16, 2);

const char \* MQTTServer = "broker.emqx.io";

const char \* MQTT\_Topic = "19004222/Bai4";

const char \* MQTT\_ID = "0b02574b-4353-4fae-bde4-1a3f07568a28";

int Port = 1883;

WiFiClient espClient;

PubSubClient client(espClient);

void WIFIConnect() {

**Serial**.println("Connecting to SSID: Wokwi-GUEST");

  WiFi.mode(WIFI\_STA);

  WiFi.begin("Wokwi-GUEST", "");

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

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("");

**Serial**.print("WiFi connected");

**Serial**.print(", IP address: ");

**Serial**.println(WiFi.localIP());

}

void MQTT\_Reconnect() {

  while (!client.connected()) {

    if (client.connect(MQTT\_ID)) {

**Serial**.print("MQTT Topic: ");

**Serial**.print(MQTT\_Topic);

**Serial**.print(" connected");

      client.subscribe(MQTT\_Topic);

**Serial**.println("");

    } else {

**Serial**.print("failed, rc=");

**Serial**.print(client.state());

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

      delay(5000);

    }

  }

}

void Display(String msg){

  lcd.setCursor(0, 0);

  lcd.print(msg);

  if(msg.length() > 15){

    for(int i=0;i<msg.length();i++){

      delay(400);

      lcd.scrollDisplayLeft();

    }

  }

}

String stMessage;

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

**Serial**.print("Message arrived on topic: ");

**Serial**.println(topic);

**Serial**.print("Message: ");

  stMessage = length == 0 ? "" : stMessage;

  for (int i = 0; i < length; i++) {

**Serial**.print((char)message[i]);

    stMessage += (char)message[i];

  }

**Serial**.println();

}

void setup() {

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

**Serial**.begin(115200);

  lcd.init();

  lcd.backlight();;

  WIFIConnect();

  client.setServer(MQTTServer, Port);

  client.setCallback(callback);

}

void loop() {

  delay(10);

  if (!client.connected()) {

    MQTT\_Reconnect();

  }

  Display(stMessage);

  client.loop();

}

Web

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8" />

        <meta http-equiv="X-UA-Compatible" content="IE=edge" />

        <meta name="viewport" content="width=device-width, initial-scale=1.0" />

        <script

            src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.min.js"

            type="text/javascript"

        ></script>

        <script

            src="https://code.jquery.com/jquery-3.6.1.min.js"

            type="text/javascript"

        ></script>

    </head>

    <body>

        <h1 style="text-align: center">

            Ứng dụng quản lý màn hình thông báo thông minh

        </h1>

        <p>Nhap noi dung can hien thi (tieng viet khong dau)</p>

        <textarea id="msg" style="width: 100%" rows="10"></textarea>

        <button>Gửi và hiển thị lên màn hình thiết bị</button>

    </body>

</html>

<script>

    client = new Paho.MQTT.Client(

        'broker.emqx.io',

        8083,

        '19004222\_NguyenDuyTrung',

    );

    client.onConnectionLost = onConnectionLost;

    client.onMessageArrived = onMessageArrived;

    client.connect({ onSuccess: onConnect });

*function* onConnect() {

        console.log('onConnect');

        client.subscribe('19004222/Bai4');

    }

*function* onConnectionLost(*responseObject*) {

        if (*responseObject*.errorCode !== 0) {

            console.log(*responseObject*);

            console.log('onConnectionLost:' + *responseObject*.errorMessage);

        }

    }

*function* onMessageArrived(*message*) {

        console.log(*message*);

    }

    $('button').on('click', *function* () {

*const* val = $('#msg').val();

        if (!val) return;

        message = new Paho.MQTT.Message(val);

        message.destinationName = '19004222/Bai4';

        client.send(message);

        $('.msg').val('');

    });

</script>