Sử dụng ionic để lập trình app thời tiết IOT:

Chuẩn bị:

* Phần cứng

1 node mcu

1 cảm biến thời tiết DHT11

Dây đực, cái, bê-đê

1 dây cáp (sạc điện thoại)

* Phần mềm

Arduino

Vào file -> preferences -> <http://arduino.esp8266.com/stable/package_esp8266com_index.json> vào ABMU.

Tải các thư viện cần thiết cho DHT, ESP, MQTT, BLYNK

Lắp đặt:

* Chân VCC (DHT) -> 3.3V (NODE MCU)
* Chân GND(DHT) -> GND (NODE MCU)
* DAT (DHT) -> D5 (NODE MCU)

Lập trình:

Cấu hình:

Với Blynk:

* Tải Blynk trên CH play:
* Đăng nhập:
* Nhấn vào New Project:
* Cấu hình lấy mã auth ( kích vào biểu tượng hình lục giác thanh trên cùng màn hình, kéo xuống dưới và thấy mã token ở đó )
* Vào Widget Box chọn Gauge, kích vào biểu tượng -> Input -> Virtual -> chọn chân V0,V1, giới hạn nhiệt độ và đơn vị.

Với IONIC:

* Tải IONIC devapp trên ch play
* Vào đây để cấu hình, tải node.js đầu tiên: <https://ionicframework.com/getting-started>
* Vào trang chủ mqtt: <http://www.hivemq.com/demos/websocket-client/>
* Cài các thư viện cho mqtt: ngx-mqtt

Code:

Với Blynk:

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include "DHT.h" // including the library of DHT11 temperature and humidity sensor

#define DHTTYPE DHT11 // DHT 11

#define dht\_dpin 14 //D5

DHT dht(dht\_dpin, DHTTYPE);

char auth[] = "BU7a0r2FLMO\_kuYQl9MiAtJX-WD48E3R"; // You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char ssid[] = "Teutons"; // Your WiFi credentials.

char pass[] = "87654321"; // Set password to "" for open networks.

float t; // Declare the variables

float h;

void setup()

{

Serial.begin(115200);// Debug console

Blynk.begin(auth, ssid, pass);

dht.begin();

//timer.setInterval(2000, sendUptime);

}

void loop()

{

Blynk.run();

//timer.run();

float h = dht.readHumidity();

float t = dht.readTemperature();

Serial.println("Humidity and temperature\n\n");

Serial.print("Current humidity = ");

Serial.print(h);

Serial.print("% ");

Serial.print("temperature = ");

Serial.print(t);

Blynk.virtualWrite(V0, t);

Blynk.virtualWrite(V1, h);

}

IONIC:

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include "DHT.h"

#include <string>

#define DHTTYPE DHT11

#define dht\_dpin 14

DHT dht(dht\_dpin, DHTTYPE);

const char\* ssid = "Teutons";

const char\* password = "87654321";

const char\* mqttServer = "broker.mqttdashboard.com";

const int mqttPort = 1883;

const char\* mqttUser = "sonmt"; // if you don't have MQTT Username, no need input

const char\* mqttPassword = "123456"; // if you don't have MQTT Password, no need input

const char\* topictemp = "temp";

const char\* topichum = "humidity";

WiFiClient espClient;

PubSubClient client(espClient);

float t;

float h;

bool turn;

void reconnect()

{

while (!client.connected()) {

Serial.println("Connecting to MQTT...");

if (client.connect("clientId-tVBumA8h7y")) {

Serial.println("connected");

} else {

Serial.print("failed with state ");

Serial.print(client.state());

delay(2000);

}

}

}

void setup() {

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

Serial.begin(115200);

turn = true;

WiFi.begin(ssid, password);

dht.begin();

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

{

delay(500);

Serial.println("Connecting to WiFi...");

}

Serial.println("Connected to the WiFi network");

client.setServer(mqttServer, mqttPort);

client.setCallback(callback);

}

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

//turn = true;

Serial.print("Message arrived in topic: ");

Serial.println(topic);

Serial.print("Message:");

for (int i = 0; i < length; i++) {

Serial.print((char)payload[i]);

}

Serial.println();

Serial.println("-----------------------");

}

void loop() {

if(!client.connected())

{

reconnect();

}

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

//if(turn)

{

h = dht.readHumidity();

t = dht.readTemperature();

char buffer[10];

dtostrf(t,0,0,buffer);

client.publish(topictemp, buffer);

client.subscribe("sender1");

dtostrf(h,0,0,buffer);

client.publish(topichum, buffer);

//client.subscribe(topic);

//turn = false;

}

delay(300);

client.loop();

}

Trên app ionic:

App.module.ts:

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { RouteReuseStrategy } from '@angular/router';

import { IonicModule, IonicRouteStrategy } from '@ionic/angular';

import { SplashScreen } from '@ionic-native/splash-screen/ngx';

import { StatusBar } from '@ionic-native/status-bar/ngx';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import {

  IMqttMessage,

  MqttModule,

  IMqttServiceOptions,

  MqttService

} from 'ngx-mqtt'

export const MQTT\_SERVICE\_OPTIONS = {

  hostname: 'broker.mqttdashboard.com',

  port: 8000,

  path: '/mqtt',

};

export function mqttServiceFactory()

{

  return new MqttService(MQTT\_SERVICE\_OPTIONS);

}

@NgModule({

  declarations: [AppComponent],

  entryComponents: [],

  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule,  MqttModule.forRoot(MQTT\_SERVICE\_OPTIONS)],

  providers: [

    StatusBar,

    SplashScreen,

    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }

  ],

  bootstrap: [AppComponent]

})

export class AppModule {}

tab2.module.ts:

import { IonicModule, ToastController } from '@ionic/angular';

import { RouterModule } from '@angular/router';

import { NgModule, Component } from '@angular/core';

import { CommonModule } from '@angular/common';

import { FormsModule } from '@angular/forms';

import { Tab2Page } from './tab2.page';

import { IMqttMessage,MqttModule,MqttService } from 'ngx-mqtt'

import { Observable } from 'rxjs'

import { MQTT\_SERVICE\_OPTIONS } from '../app.module';

@NgModule({

  imports: [

    IonicModule,

    CommonModule,

    FormsModule,

    MqttModule.forRoot(MQTT\_SERVICE\_OPTIONS),

    RouterModule.forChild([{ path: '', component: Tab2Page }])

  ],

  declarations: [Tab2Page]

})

export class Tab2PageModule {

  static texttemp: string;

  static texthum: string;

  topic1 = 'temp';

  topic2 = 'humidity';

  constructor(private toastCtrl: ToastController, private \_mqttService: MqttService)

  {

    this.\_mqttService.observe(this.topic1).subscribe((message: IMqttMessage) => {

      Tab2PageModule.texttemp = message.payload.toString();

      //console.log(this.texttemp);

    });

    this.\_mqttService.observe(this.topic2).subscribe((message: IMqttMessage) => {

      Tab2PageModule.texthum = message.payload.toString();

      //console.log(this.texthum);

    });

  }

}

Tab2.page.html:

<ion-header>

    <ion-toolbar>

        <ion-title>

            Tab Two

        </ion-title>

    </ion-toolbar>

</ion-header>

<ion-content>

    <div style="margin-top: 75px;">

        <h2>gia tri {{texttemp}}</h2>

    </div>

    <div>

        <h2>gia tri {{texthum}}</h2>

    </div>

</ion-content>