

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA THÀNH PHỐ HỒ CHÍ MINH
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



Phát triển Ứng dụng Internet of Things (CO3037)

MINI PROJECT

Sinh viên thực hiện: Tô Dịu Quang - 2014251

Ngày nộp báo cáo: 9th May 2024



Contents

1	Lập trình trên vi xử lý	2
1.1	Các thành phần	2
1.2	Mã nguồn của Arduino	2
2	Nhận dữ liệu từ vi xử lý và chuyển lên server	3
2.1	Kết nối với server	3
2.2	Nhận dữ liệu từ vi xử lý bằng UART	4
3	Tạo Android App	5
3.1	Kết nối MQTT với Adafruit IO	5
3.2	Chuyển đổi dữ liệu	5
3.3	Thiết kế UI	6
3.3.1	Giao diện TextView trên Android Studio	7
3.3.2	Giao diện Toggle Button trên Android Studio	7
4	Link GITHUB	7
	List of Figure	8

1 Lập trình trên vi xử lý

Dùng Arduino Uno R3 kết hợp với DHT22 và 1 switch để xử lý, thu thập dữ liệu và điều khiển dữ liệu. Trao đổi dữ liệu bằng UART.

1.1 Các thành phần

- Arduino Uno R3 là một bảng mạch vi điều khiển mã nguồn mở dựa trên vi điều khiển Microchip ATmega328. Dòng mạch này thường được sử dụng để tự thiết kế các mạch điện tử, như điều khiển đèn LED, đo nhiệt độ - độ ẩm và hiển thị lên màn hình LCD, hay các ứng dụng khác.
- Cảm biến DHT22 là một cảm biến nhiệt độ và độ ẩm kỹ thuật số. Nó sử dụng cảm biến độ ẩm điện dung và điện trở nhiệt để đo không khí xung quanh và phát ra tín hiệu kỹ thuật số. DHT22 hiệu chỉnh tín hiệu kỹ thuật số đầu ra và sử dụng kỹ thuật thu thập tín hiệu kỹ thuật số và công nghệ cảm biến độ ẩm độc quyền.

1.2 Mã nguồn của Arduino

```
1 #include <DHT.h>
2 #define DHTPIN 4
3 #define DHTTYPE DHT22
4
5 DHT dht(DHTPIN, DHTTYPE);
6
7 const int switchPin = 3;
8 int counter_button = 0;
9 int counter_th = 0;
10 int previousState = HIGH;
11
12 void setup() {
13     Serial.begin(115200);
14     dht.begin();
15     pinMode(switchPin, INPUT_PULLUP);
16 }
17 void loop() {
18     counter_button += 1;
19     counter_th += 1;
20     // Đọc trạng thái nút nhấn
21     int currentState = digitalRead(switchPin);
22     if (currentState != previousState) {
23         Serial.print("!1:B:");
24         Serial.print(currentState);
25         Serial.print(":N:");
26         Serial.print("None");
27         Serial.print("#");
28         Serial.println();
29         previousState = currentState;
30     }
31     // gửi dữ liệu qua UART
32     if (counter_th == 5000) {
33         // Đọc dữ liệu từ cảm biến DHT22
34         float humidity = dht.readHumidity();
35         float temperature = dht.readTemperature();
36         Serial.print("!1:T:");
37         Serial.print(String(temperature, 1));
```

```
38     Serial.print(":H:");
39     Serial.print(String(humidity, 1));
40     Serial.print("#");
41     Serial.println();
42     counter_th = 0;
43 }
44 delay(1);
45 }
```

2 Nhận dữ liệu từ vi xử lý và chuyển lên server

2.1 Kết nối với server

Dùng thư viện MQTT để có thể kết nối với tài khoản trên server, sau đó subscribe vào các topic để có thể nhận được tín hiệu khi có sự thay đổi dữ liệu,

```
1 import sys
2 import datetime
3 from Adafruit_IO import MQTTClient
4 import time
5 # from simple_ai import *
6 from uart import *
7 AIO_FEED_ID = ["nutnhan1", "nutnhan2", "nutnhan3"]
8 AIO_USERNAME = "todiuquang123"
9 AIO_KEY = "aio_oRDa290TjTmnnoYvRkuhqgvtXcN"
10 def connected(client):
11     print("Ket noi thanh cong ...")
12     for topic in AIO_FEED_ID:
13         client.subscribe(topic)
14 def subscribe(client, userdata, mid, granted_qos):
15     print("Subscribe thanh cong ...")
16 def disconnected(client):
17     print("Ngat ket noi ...")
18     sys.exit(1)
19 def message(client, feed_id, payload):
20     print("Nhan du lieu: " + payload + " ,feed id:" + feed_id)
21 client = MQTTClient(AIO_USERNAME, AIO_KEY)
22 client.on_connect = connected
23 client.on_disconnect = disconnected
24 client.on_message = message
25 client.on_subscribe = subscribe
26 client.connect()
27 client.loop_background()
28 counter_time = 1
29 while True:
30
31     \\ tat den vao luc 17h hang ngay
32     counter_time = counter_time - 1
33     now = datetime.datetime.now()
34     current_time = now.strftime("%H:%M:%S")
35
36     if counter_time <= 0:
37         counter_time = 1
38         if current_time == "17:00:00":
39             client.publish("nutnhan1", 1)
40 readSerial(client)
41 time.sleep(1)
```

2.2 Nhận dữ liệu từ vi xử lý bằng UART

Kết nối với thiết bị thông qua COM8 sau đó nhận dữ liệu và giải mã dữ liệu, việc giải mã sẽ giúp có thể xác định được dữ liệu gửi qua thuộc topic nào, sau đó sẽ public lên server theo đúng topic đó.

```
1 import serial.tools.list_ports
2 def getPort():
3     ports = serial.tools.list_ports.comports()
4     N = len(ports)
5     commPort = "None"
6     for i in range(0, N):
7         port = ports[i]
8         strPort = str(port)
9         if "USB Serial Device" in strPort:
10             splitPort = strPort.split(" ")
11             commPort = (splitPort[0])
12     return "COM8"
13
14 if getPort() != "None":
15     ser = serial.Serial( port=getPort(), baudrate=115200)
16     print(ser)
17
18 def processData(client, data):
19     data = data.replace("!", "")
20     data = data.replace("#", "")
21     splitData = data.split(":")
22     print(splitData)
23     if splitData[1] == "T":
24         client.publish("cambien1", splitData[2])
25     if splitData[3] == "H":
26         client.publish("cambien2", splitData[4])
27     if splitData[1] == "B":
28         client.publish("nutnhan2", splitData[2])
29
30 mess = ""
31 def readSerial(client):
32     bytesToRead = ser.inWaiting()
33     if (bytesToRead > 0):
34         global mess
35         mess = mess + ser.read(bytesToRead).decode("UTF-8")
36         while ("#" in mess) and ("!" in mess):
37             start = mess.find("!")
38             end = mess.find("#")
39             processData(client, mess[start:end + 1])
40             if (end == len(mess)):
41                 mess = ""
42             else:
43                 mess = mess[end+1:]
44
45 def writeData(data):
46     ser.write(str(data).encode())
```

3 Tạo Android App

3.1 Kết nối MQTT với Adafruit IO

Tạo 1 class MQTTHelper để thực hiện việc kết nối lên server Adafruit IO. Trong class MQTTHelper tạo 1 hàm để kết nối vào các feed id trên Adafruit IO để có thể cập nhật vào các biến khi dữ liệu trên hệ thống bị thay đổi.

```
1 private void subscribeToTopic() {
2     for(int i = 0; i < arrayTopics.length; i++) {
3         try {
4             mqttAndroidClient.subscribe(arrayTopics[i], 0, null, new
5             IMqttActionListener() {
6                 @Override
7                 public void onSuccess(IMqttToken asyncActionToken) {
8                     Log.d("TEST", "Subscribed!");
9                 }
10                @Override
11                public void onFailure(IMqttToken asyncActionToken, Throwable
12                exception) {
13                    Log.d("TEST", "Subscribed fail!");
14                }
15            });
16        } catch (MqttException ex) {
17            System.err.println("Exceptionst subscribing");
18            ex.printStackTrace();
19        }
20    }
21 }
```

3.2 Chuyển đổi dữ liệu

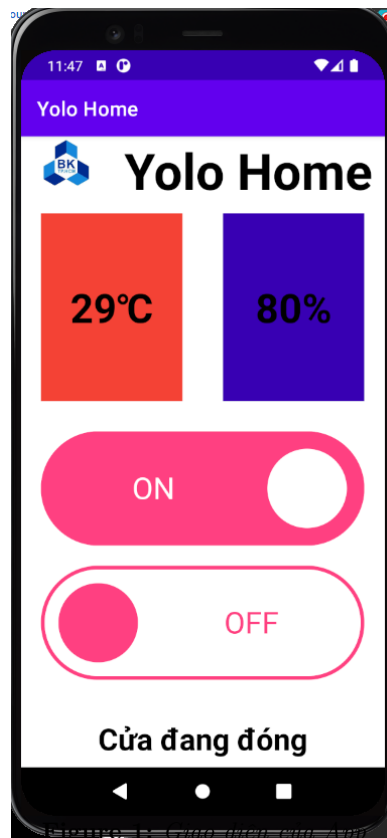
Dùng 1 function để chuyển đổi dữ liệu số hoặc chuỗi được lấy từ AdaFruit Io về thành dữ liệu thích hợp để hiển thị.

- Đối với nhiệt độ và độ ẩm thì sẽ thêm độ C và % vào sau.
- Đối với các nút nhấn nếu dữ liệu là 1 sẽ dùng hàm setOn(true) để bật nút nhấn, nếu là 0 thì setOn(false)
- Đối với cửa thì sẽ xem dữ liệu của nút nhấn và hiển thị TextView tương ứng

```
1 public void messageArrived(String topic, MqttMessage message) throws
2 Exception {
3     Log.d("TEST", topic + "****" +message.toString() );
4     if (topic.contains("cambien1")){
5         txtTemp.setText(message.toString()+"C");
6     } else if (topic.contains("cambien2")){
7         txtHumi.setText(message.toString()+"%");
8     } else if (topic.contains("nutnhan1")){
9         if (message.toString().equals("1")){
10             btnLED.setOn(true);
11         }else{
12             btnLED.setOn(false);
13         }
14     }
15 }
```

```
12     }
13     } else if (topic.contains("nutnhan2")){
14         if (message.toString().equals("1")){
15             btnPUMP.setOn(true);
16             txtDoor.setText(" C a      ang      m I ");
17         }else{
18             btnPUMP.setOn(false);
19             txtDoor.setText(" C a      ang      ng  ");
20         }
21     } else if (topic.contains("nutnhan3")){
22         if (message.toString().equals("1")){
23             btnSENSOR.setOn(true);
24         }else{
25             btnSENSOR.setOn(false);
26         }
27     }
28 }
```

3.3 Thiết kế UI



- Trên cùng là tên của ứng dụng và logo Bách khoa.
- Chia 2 cột TextView để hiển thị nhiệt độ và độ ẩm lấy được từ cảm biến.

- 2 nút nhấn được thiết kế để điều khiển lần lượt là đèn Led và Quạt.
- TextView cuối cùng để hiển thị trạng thái của cửa hiện tại.

3.3.1 Giao diện TextView trên Android Studio

Thực hiện giao diện Textview bằng các lệnh Android có sẵn trong Android Studio

```
1 <TextView
2     android:id="@+id/txtTemperature"
3     android:layout_width="0dp"
4     android:layout_height="match_parent"
5     android:layout_weight="1"
6     android:layout_margin="20dp"
7     android:background="#F44336"
8     android:text="20°C"
9     android:textColor="@color/black"
10    android:textSize="40dp"
11    android:textStyle="bold"
12    android:gravity="center">
13 </TextView>
```

- Các lệnh layout dùng để setting về vị trí và khoảng cách của TextView trong giao diện
- Lệnh background dùng để chọn màu nền
- Lệnh text dùng để khởi tạo giá trị đầu tiên hiển thị khi vừa bật ứng dụng, khi dữ liệu trên Adafruit cập nhật thì sẽ tự thay đổi
- Lệnh gravity = center để căn chỉnh cho phần text nằm giữa.

3.3.2 Giao diện Toggle Button trên Android Studio

Tham khảo giao diện nút nhấn ở <https://github.com/singhangadin/android-toggle> sau đó sử dụng và thêm 1 số lệnh về layout để phù hợp với giao diện trong bài tập

```
1 <com.github.angads25.toggle.widget.LabeledSwitch
2     android:layout_weight="20"
3     android:id="@+id/btnPUMP"
4     android:layout_width="match_parent"
5     android:layout_height="0dp"
6     android:layout_marginTop="10dp"
7     android:layout_marginLeft="20dp"
8     android:layout_marginBottom="10dp"
9     android:layout_marginRight="20dp"
10    android:textSize="30dp"
11    app:on="false"
12    app:colorBorder="@color/colorAccent"
13 />
```

4 Link GITHUB

Link chứa source code: [TẠI ĐÂY](#)

Link chứa video demo: [TẠI ĐÂY](#)



List of Figures

1	Giao diện của App	6
---	-----------------------------	---