TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

BÁO CÁO NIÊN LUẬN CƠ SỞ
NGÀNH TRUYỀN THÔNG & MẠNG MÁY TÍNH

HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ & ĐỘ ẨM TRONG NHÀ

GVHD:
TS. Ngô Bá Hùng

Sinh Viên Thực Hiện:
Nguyễn Thanh Huy – B1709280

CANTHO UNIVERSITY

Nội Dung

I: Giới thiệu

1. Giới thiệu các module

II: Mục tiêu đề tài

III: Nội dung nghiên cứu

IV: Đặc tả bài toán

1. Đặc tả các module

V: Giải pháp công nghệ

VI: Phân tích thiết kế giải pháp

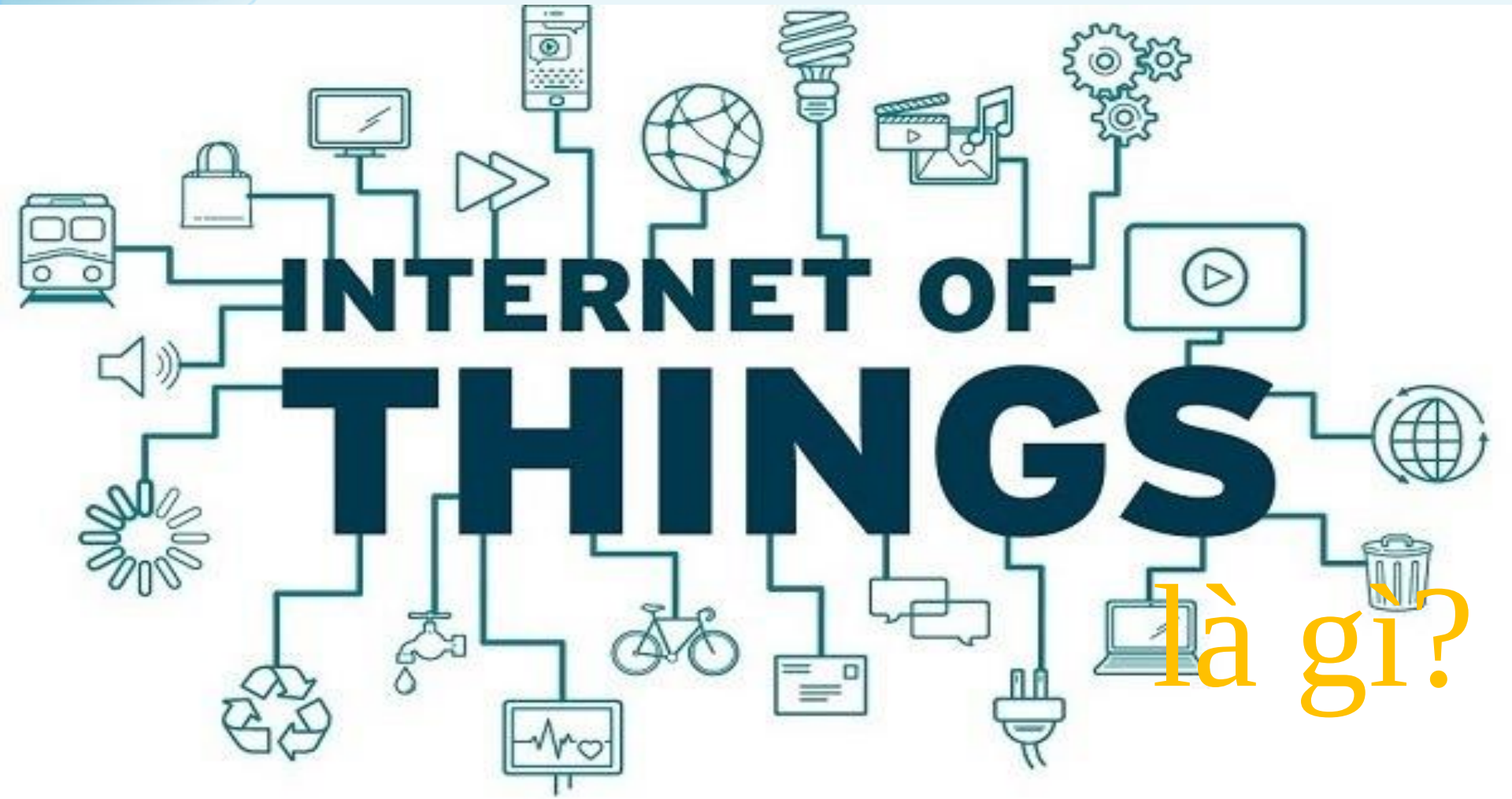
VII: Đánh giá kiểm thử giải pháp

VIII: Kết luận

IX: Tài liệu tham khảo

X: Demo

I: Giới Thiệu

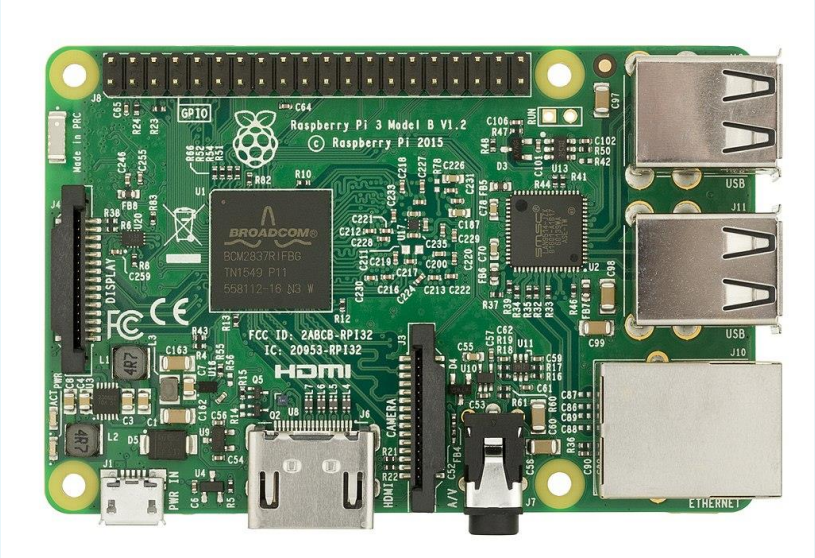


là gì?

I: Giới Thiệu

Tại sao lập trình IoT trên Raspberry lại phổ biến?

- Thích hợp với nhiều nền tảng
- Chạy đa ngôn ngữ lập trình
- Hiệu năng ổn định
- Giá thành rẻ

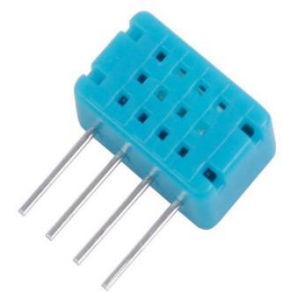


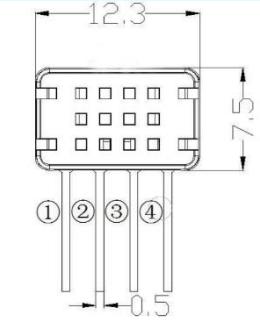
Raspberry Pi 3 Model B

I: Giới Thiệu

Giới thiệu các module

1: Sensor DHT12





1: Chân VDD (Điện áp vào từ 2.7V – 5.5V)

2: Chân SDA (Chân xuất tính hiệu)

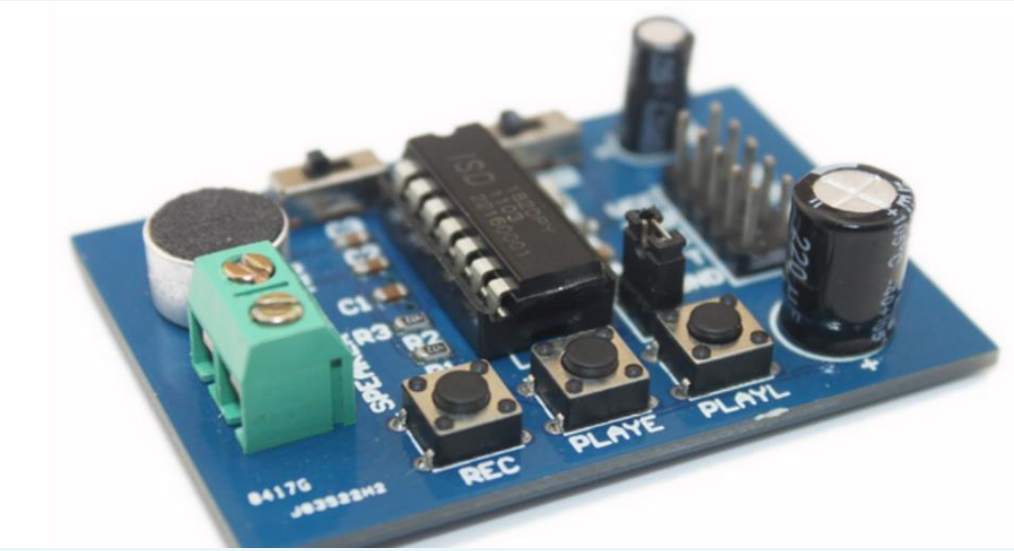
3: Chân GND (Điện áp vào từ nguồn âm)

4: Chân SCL (Chân xuất tính hiệu)

I: Giới Thiệu

Giới thiệu các module

2: Module ISD1820



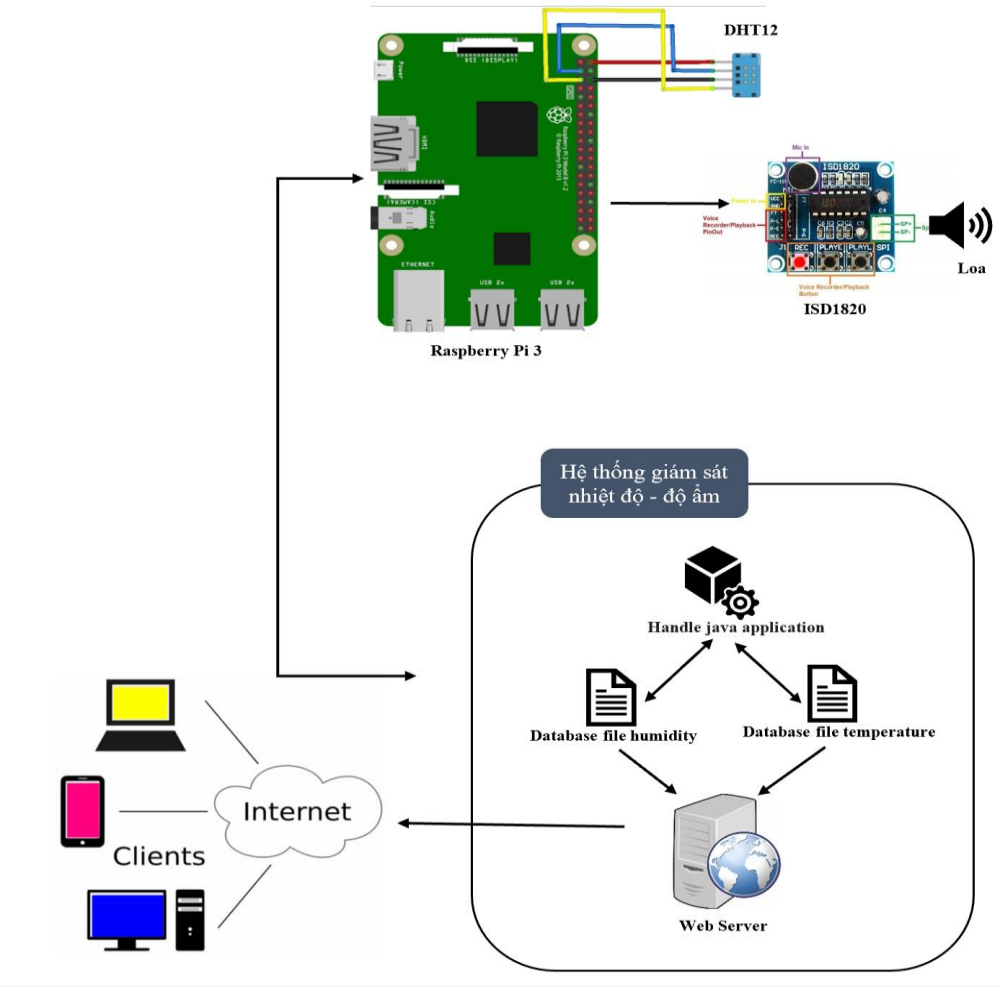
II: Mục Tiêu Đề Tài

- Nghiên cứu lập trình IoT bằng ngôn ngữ Java trên Raspberry Pi.
- Tiếp cận làm quen với thư viện Pi4J.
- Xây dựng mô hình nhỏ nhà thông minh rồi mở rộng ra các lĩnh vực khác.

III: Nội Dung Nghiên Cứu

Mô tả hệ thống

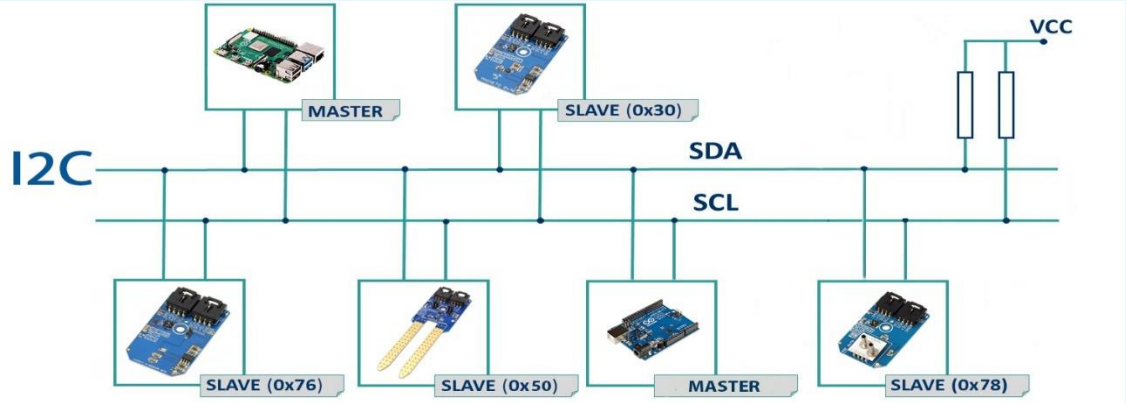
Hệ thống giám sát
nhiệt độ - độ ẩm trong nhà



III: Nội Dung Nghiên Cứu

Giao thức truyền tải dữ liệu I2C

- Bộ xử lý trung tâm truyền dữ liệu trên hai đường truyền tính hiệu.
- Các Bit dữ liệu được truyền theo các khoảng thời gian đều đặn



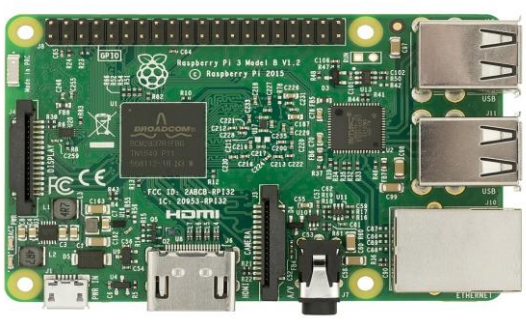
Giao tiếp I2C

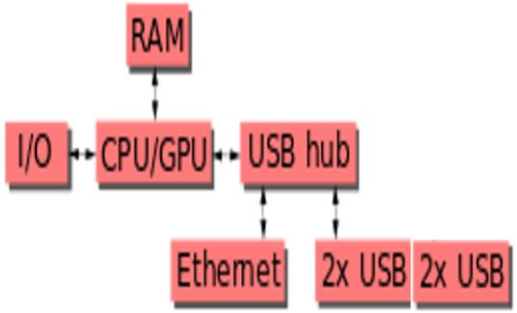
IV: Đặc Tả Bài Toán

Đặc tả máy tính nhúng Raspberry Pi 3

Raspberry Pi 3 với CPU ARM Cortex-A53 Quadcore 1.2GHz 64-bit, RAM 1GB…

- Lợi ích công nghệ
  - Sử dụng được đa ngôn ngữ lập trình.
  - Giá thành rẻ.
- Vấn đề bảo mật
  - Chứng thực trên trên Raspberry cần được sát thực.

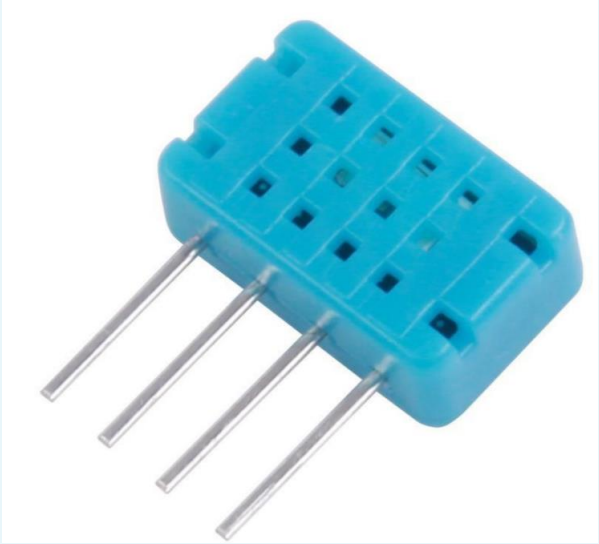




IV: Đặc Tả Bài Toán

Đặc tả Sensor DHT12

- Lợi ích công nghệ.
  - Là phiên bản nâng cấp của DHT11.
  - Dữ liệu được truyền trên một cổng GPIO.
  - Giao tiếp với I2C.
- Vấn đề bảo mật
  - Dữ liệu dễ bị đánh cắp.

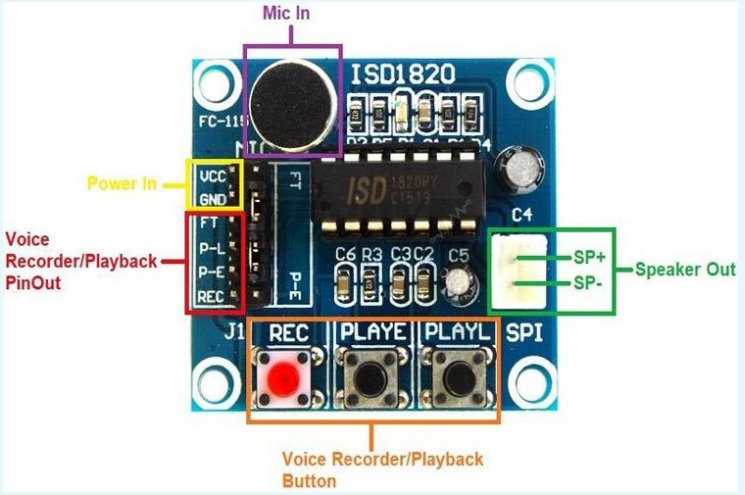


IV: Đặc Tả Bài Toán

Đặc tả module ISD1820

- Lợi ích công nghệ

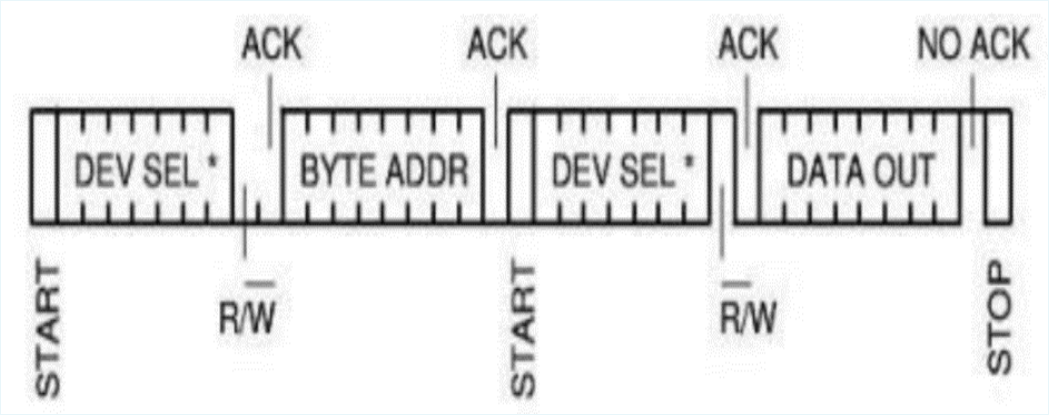
  - Ghi lại âm thanh phát.
  - Ứng dụng vào các hệ thống cần người chào hỏi.



V: Giải Pháp Công Nghệ

Truyền tính hiệu bằng I2C trên sensor DHT12

- Được thiết lập truyền bằng giáo tiếp I2C.
- Kết nối dễ dàng với Raspbery Pi.
- Dữ liệu đọc từ Raspberry Pi được lưu vào 5 mảng, mỗi mãng gồm 8 bit.

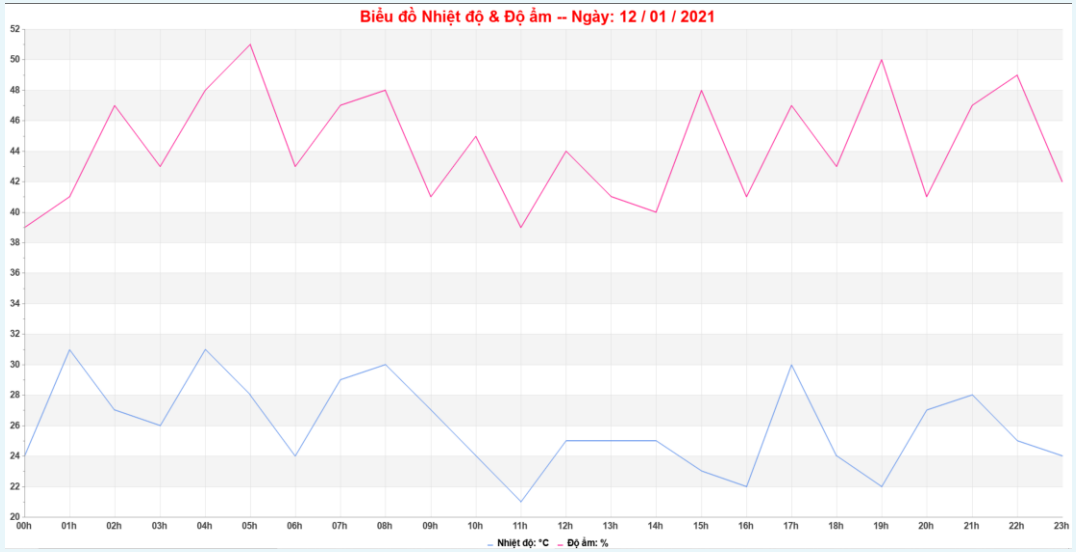


Giao thức I2C

V: Giải Pháp Công Nghệ

Vẽ biểu đồ bằng thư viện Jpgraph

- Dễ sử dụng.
- Cung cấp giao diện biểu đồ trực quan.
- Đọc và xử lý hiển thị dữ liệu tự động điều chỉnh tham số.

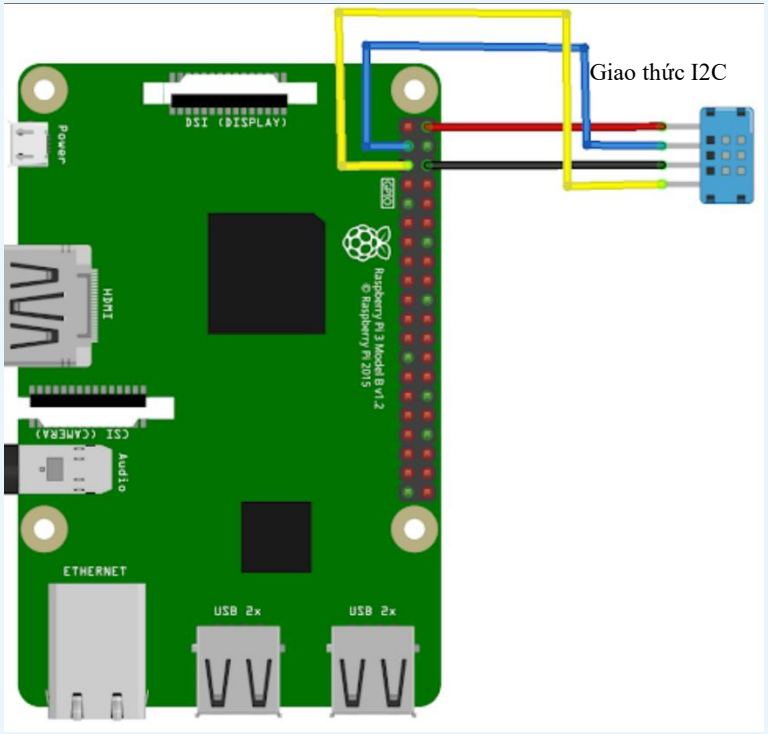


Biểu đồ Jpgraph

VI: Phân Tích Thiết Kế Giải Pháp

Raspberry Pi đọc dữ liệu từ DHT12

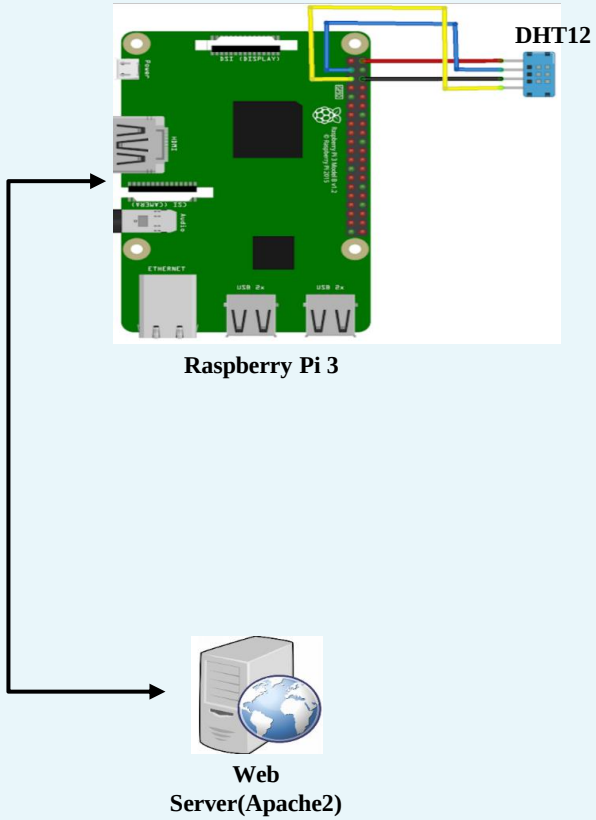
- Giao tiếp với nhau bằng giao tiếp I2C
- Dữ liệu được đọc và lưu tuần tự



VI: Phân Tích Thiết Kế Giải Pháp

Xây dựng Apache2 trên Raspberry Pi

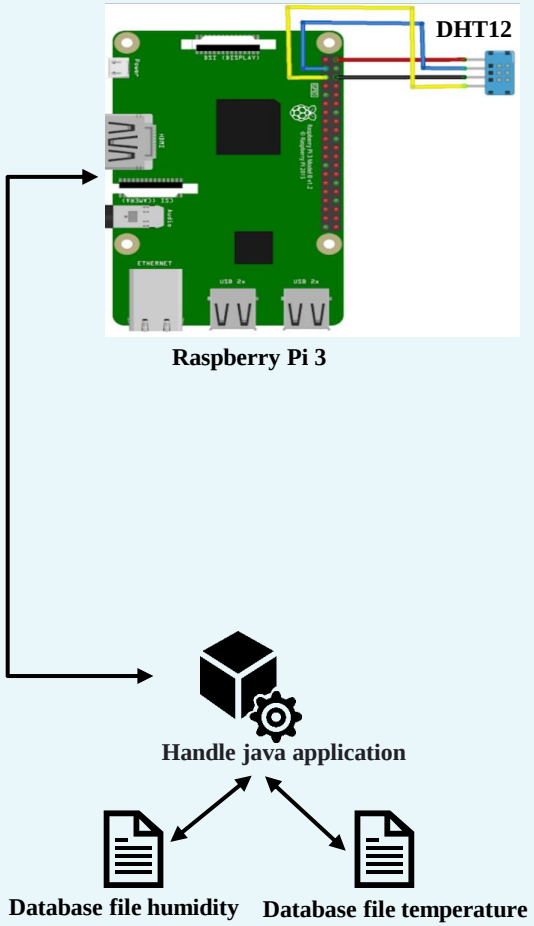
- Truyền dữ liệu lên Webserver từ Raspberry Pi



VI: Phân Tích Thiết Kế Giải Pháp

Xử lý và lưu trữ dữ liệu

- Cứ sau mỗi 30 giây sẽ đọc giá trị một lần.
- Webserver đọc dữ liệu và truyền lên hiển thị trên biểu đồ website.



VI: Phân Tích Thiết Kế Giải Pháp

Lập trình đoạn code trả về nhiệt độ, độ ẩm

```
if(data[4]==(data[0]+data[1]+data[2]+data[3])){

        double doam_=Double.parseDouble(data[0]+"."+data[1]);
        double nhietdo_=Double.parseDouble(data[2]+"."+data[3]);
        doam = (float)doam_;
        nhietdo = (float)nhietdo_;
        nhietdoint = (int)nhietdo;
        System.out.println("Nhiet Do : "+ nhietdo +" °C " + "--- Do am : "+ doam + " %");

}
else
        System.out.println("Khong nhan duoc Data");
```

VI: Phân Tích Thiết Kế Giải Pháp

Lập trình đoạn code trả về ngày, giờ

```
//----------------------Ham tra ve nhiet do, do am, ngay, gio ----------

SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
Date date = new Date();
String ngaygio = formatter.format(date);
int len = ngaygio.length();
//System.out.println("Ngày đã được định dạng : "+ngaygio+"Do dai: "+len);
int z = ngaygio.indexOf(" ");
String ngay = ngaygio.substring(0,z);
String gio = ngaygio.substring(z+1,z+3);
System.out.println("Ngay: "+ ngay + "--Gio: " + gio);
int gioint = Integer.parseInt(gio);
```

VI: Phân Tích Thiết Kế Giải Pháp

Lập trình đoạn code điều kiện bật loa báo động

```
if(nhietdoint > 40 | nhietdoint<=0){
        try{
          System.out.println("Canh bao!! Nhiet do vuot qua nguong cho phep");
        pin.high();
        Thread.sleep(15000);
    // turn off gpio pin #01
        pin.low();
        Thread.sleep(15000);
        }//close try
        catch(Exception e){
            System.out.println("Loi nhap xuat!");
        }
    }
```

VI: Phân Tích Thiết Kế Giải Pháp

Lập trình đoạn code ghi nhận giá trị nhiệt độ và độ ẩm vào file

```
//---------------------- Dieu kien gio = 1 ----------------------------
        else if((gioint==1)){
                try{
                    for(int l=0; l<1; l++){
                        fw.write("" + result.get(l) + "\n");
                        fw2.write("" + result2.get(l) + "\n");
                    }//close for

                    fw.write("" + nhietdo + "\n");
                    fw2.write("" + doam + "\n");
                    for(int k=2; k<24; k++){
                        fw.write("" + result.get(k) + "\n");
                        fw2.write("" + result2.get(k) + "\n");
                    }//close for
                    fw.close();
                    fw2.close();
                }//close try
                 catch(Exception e){
                    System.out.println("Loi ghi file!");
                 }
            }//close else if
```

VII: Đánh Giá Kiểm Thử Giải Pháp

Kiểm thử giải pháp

Mô hình giám sát nhiệt độ - độ ẩm trong nhà.

```
filedoam.txt  filenhietdo.txt  lib  NLCS_DHT12.jar  README.TXT
pi@HuyGao:~/NetBeansProjects/NLCS_DHT12/dist $ java -jar NLCS_DHT12.jar
Project Nien luan co so: He thong giam sat nhiet do - do am trong nha
Nhiet Do : 25.8 °C --- Do am : 40.1 %
Ngay: 12/01/2021--Gio: 13
```

VII: Đánh Giá Kiểm Thử Giải Pháp

Đánh giá

- Mục tiêu kiểm thử xem lập trình IoT bằng ngôn ngữ Java trên Raspberry có chạy ổn định và khả năng tùy biến cao không thì cho được kết quả rất ổn định và tương thích với nhiều module, sensor…
- Có thể biến Raspberry Pi thành một máy chủ quản lý nhà thông minh trong nhà, có khả năng tương thích cao và lập trình trên nhiều ngôn ngữ. Ít tiêu tốn điện, chi phí vận hành rẻ.

VIII: Kết luận

- Mô hình giám sát nhiệt độ - độ ẩm trong nhà đã cho kết quả chính sát, chu trình của mô hình khép kín hoạt động liên tục, dữ liệu nhiệt độ và độ ẩm được cập như theo thời gian thực sau mỗi 30 giây.
- Giao diệp web trực quan có biểu đồ hiển hiển thị nhiệt độ và độ ẩm theo thời gian thực, giúp người dùng dễ giám sát và trực quan khi xem.
- Giám sát nhiệt độ trong nhà theo thời gian thực, nếu nhiệt độ trong nhà vượt quá ngưỡng cho phép (cháy nhà, …) sẽ thông báo cho chủ nhà bằng loa.

IX: Tài Liệu Tham Khảo

[1] https://pi4j.com/0.0.5/apidocs/com/pi4j/wiringpi/Gpio.html

[2] https://www.instructables.com/Efficient-Development-of-Java-for-the-Raspberry-Pi/

[3] _ https://jpgraph.net/features/gallery.php#line1

X: Demo

Demo mô hình giám sát nhiệt độ - độ ẩm trong nhà

CẢM ƠN THẦY VÀ CÁC BẠN ĐÃ LẮNG NGHE