

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**NIÊN LUẬN NGÀNH
TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH**

Đề tài

**LẬP TRÌNH IoT BẰNG NGÔN NGỮ JAVA TRÊN
RASPBERRY PI
HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ - ĐỘ ẨM
TRONG NHÀ**

**Sinh viên thực hiện :
Nguyễn Thanh Huy
Khóa : K43**

Cần Thơ, 01/2021

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**NIÊN LUẬN
NGÀNH TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH**

Đề tài

**LẬP TRÌNH IoT BẰNG NGÔN NGỮ JAVA TRÊN
RASPBERRY PI
HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ - ĐỘ ẨM
TRONG NHÀ**

**Giáo viên hướng dẫn:
TS. Ngô Bá Hùng**

**Sinh viên thực hiện:
Nguyễn Thanh Huy
Khóa : K43**

Cần Thơ, 01/2021

[illegible]

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Ngô Bá Hùng – Giảng viên hướng dẫn bộ môn Niên Luận Cơ Sở đã giành thời gian quý báu để hướng dẫn cho em trong môn học này. Cảm ơn thầy, đã tạo điều kiện và cung cấp những kiến thức quan trọng để học tập và thực hiện đề tài một cách tốt nhất.

Thông qua quá trình thực hiện niên luận, em phân nào củng cố và tích lũy được những kiến thức về lập trình ngôn ngữ Java trên Raspberry Pi. Mặc dù, đã cố gắng hết sức trong tất cả quá trình từ học tập cho đến thực hiện đề tài này. Nhưng em đã không thể tránh khỏi những sai sót nhất định. Em rất mong được sự thông cảm, bỏ qua và góp ý tận tình từ Thầy và các bạn.

Em xin chân thành cảm ơn !

Cần Thơ, ngày 14 tháng 01 năm 2021
Người viết

Nguyễn Thanh Huy

MỤC LỤC

PHẦN I: GIỚI THIỆU	1
1. Đặt vấn đề.....	1
2. Mục tiêu đề tài.....	1
2.1 Đối tượng nghiên cứu	2
2.2 Nội dung nghiên cứu.....	2
PHẦN II: NỘI DUNG.....	3
Chương 1: Mô Tả Bài Toán	3
1.1. Đặt tả yêu cầu	3
1.2. Đối tượng nghiên cứu.....	3
1.3. Phần cứng và phần mềm được sử dụng.....	3
1.3.1. Sensor DHT12.....	3
1.3.1.1 Tổng quan.....	3
1.3.1.2 Thông số kỹ thuật.....	3
1.3.1.3 Nguyên lý hoạt động	4
1.3.2. Module ISD1820	5
1.3.2.1 Tổng quan.....	5
1.3.2.2. Thông số kỹ thuật.....	5
1.3.3. Raspberry Pi 3	6
1.3.3.1. Tổng quan.....	6
1.3.3.2. Phần cứng	6
1.3.3.3. Cổng GPIO	6
1.3.4. Pi4J Library	8
1.3.4.1. Tổng quan.....	8
1.3.4.2. Một số tính năng cơ bản	8
Chương 2: Thiết Kế Giải Pháp	9
1. Kiến trúc hệ thống:	9
2. Thiết kế giải thuật xử lý	10
3. Thiết kế.....	11
3.1. Giao thức I2C	11
3.2. Raspberry Pi đọc dữ liệu từ DHT12.....	12
3.3. Apache2 chạy trên Raspberry Pi	13
3.4. Xử lý và lưu trữ dữ liệu	14
3.5. Đọc dữ liệu lưu trữ lên Web Server	15
3.6 Biểu diễn dữ liệu lên biểu đồ Website	16

Chương 3: Giải Pháp Thiết Kế	17
1. Đọc và lưu vào mảng thông qua giao thức I2C	17
2. Nối mảng byte để trả kết quả nhiệt độ và độ ẩm	17
3. Hàm trả kết quả ngày giờ	18
4. Hàm ghi dữ liệu vào file txt.....	18
5. Hàm điều kiện vượt ngưỡng cho phép	18
6. Đọc dữ liệu từ file txt xử lý trên Web Server	19
Chương 4: Cài Đặt Giải Pháp.....	20
4.1. Cài đặt và quản trị hệ điều hành Raspbian trên Raspberry Pi	20
4.2. Thiết lập môi trường phát triển ứng dụng Java trên Raspberry Pi	20
4.3. Cài đặt NetBeans trên máy trạm.....	20
4.4. Cài đặt Apache2 và PHP trên Raspberry Pi 3	26
4.5. Cài đặt Jpgraph Library	27
4.6. Tạo file TXT để lưu trữ dữ liệu nhiệt độ và độ ẩm	28
CHƯƠNG 5: Đánh Giá Và Kiểm Thử	29
1. Đánh giá	29
2. Kiểm thử.....	29
PHẦN III. KẾT LUẬN	30
1. Kết quả đạt được	30
2. Hướng phát triển.....	30
TÀI LIỆU THAM KHẢO	31

PHẦN I: GIỚI THIỆU

1. Đặt vấn đề

Với sự bùng nổ của cuộc cách mạng công nghiệp 4.0, đi cùng với sự phát triển nhanh chóng tốc độ của Internet hiện nay, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Chính vì điều này, đã thúc đẩy sự khai sinh ra mạng lưới thiết bị kết nối Internet(IoT) làm thay đổi đáng kể bộ mặt thế giới. Các thiết bị IoT có khả năng kết nối mạng máy tính giúp các thiết bị thu thập và truyền tải dữ liệu cũng như cung cấp cho chúng ta hạ tầng cơ sở toàn cầu phục vụ cho xã hội thông tin, mỗi thiết bị IoT có thể được lập trình bằng các ngôn ngữ lập trình khác nhau như Python, Java, C... Một trong những thiết bị IoT hỗ trợ chạy đa ngôn ngữ lập trình đó là máy tính nhúng Raspberry Pi. Không những lập trình được đa ngôn ngữ, giá thành rẻ mà còn có thể ứng dụng vào các dự án như là làm Server quản lý nhà thông minh, Robot, Web Server... Trong số đó thì giám sát nhiệt độ - độ ẩm thông qua biểu đồ trên website được chú trọng, vì không những cho ta biết nhiệt độ và độ ẩm trong nhà mà có thể nhận biết được nhiệt độ bất thường thông qua biểu đồ giúp chúng ta phát hiện hỏa hoạn cũng như bảo vệ tính mạng, sức khỏe cho chính bản thân mình và gia đình.

2. Mục tiêu đề tài

Chúng ta xây dựng hệ thống giám sát và cung cấp giao diện Web trực quan cho người dùng để biết được nhiệt độ và độ ẩm trong nhà cũng như cho ta độ bảo mật cao và tiết kiệm chi phí khi so với các thiết bị của những nhà cung cấp bên ngoài.

Cách vận hành và quản lý các cảm biến một cách đơn giản, tự động chỉ với những cổng **GPIO** trên **Raspberry Pi**.

2.1 Đối tượng nghiên cứu

Với nhu cầu của con người về các thiết bị trong nhà cần tự động hóa và thông minh hơn cũng như dữ liệu luôn được cập nhật nhanh chóng. Vì vậy ta cần tập chung giải pháp lưu trữ dữ liệu nhiệt độ - độ ẩm thường xuyên, đọc dữ liệu từ sensor DHT12 lên các chân GPIO trên Raspberry và xử lý rồi tải lên Web liên tục.

Tìm hiểu về Sensor DHT12 bao gồm:

Các chân kết nối xuất tính hiệu, phương thức thức kết nối, giao thức kết nối với GPIO trên Raspberry Pi.

Tìm hiểu về máy tính nhúng Raspberry Pi bao gồm:

Các chân kết nối nhận tính hiệu trên GPIO, hệ điều hành tương thích, services được sử dụng, cài đặt biến môi trường.

Tìm hiểu về thư viện Pi4J để lập trình giả lập trên các cổng GPIO trên Raspberry Pi.

2.2 Nội dung nghiên cứu

Tìm hiểu và cài đặt biến môi trường trên Raspberry Pi và máy trạm.

Thu thập dữ liệu từ sensor sau đó xử lý và lưu trữ.

Tải dữ liệu và hiển thị biểu đồ nhiệt độ - độ ẩm trên Web Server.

PHẦN II: NỘI DUNG

Chương 1: Mô Tả Bài Toán

1.1. Đặt tả yêu cầu

Câu hỏi đặt ra, chúng ta làm thế nào để xây dựng hệ thống giám sát nhiệt độ - độ ẩm trong nhà, dùng thư viện gì để giao tiếp cũng như nhập xuất tín hiệu. Để làm được những điều đó chúng ta cần nghiên cứu về mặt lý thuyết trên thư viện Pi4J, máy tính nhúng Raspberry Pi, các module như ISD1820, DHT12...

1.2. Đối tượng nghiên cứu

Với nhu cầu của con người về các thiết bị trong nhà cần tự động hóa và thông minh hơn cũng như dữ liệu luôn được cập nhật nhanh chóng. Vì vậy ta cần tập chung giải pháp lưu trữ dữ liệu nhiệt độ - độ ẩm thường xuyên, đọc dữ liệu từ sensor DHT12 lên các chân GPIO trên Raspberry và xử lý rồi tải lên Web liên tục.

Tìm hiểu về Sensor DHT12 bao gồm: Các chân kết nối xuất tính hiệu, phương thức thức kết nối, giao thức kết nối với GPIO trên Raspberry Pi.

Tìm hiểu về máy tính nhúng Raspberry Pi bao gồm: Các chân kết nối nhận tính hiệu trên GPIO, hệ điều hành tương thích, services được sử dụng, cài đặt biến môi trường.

Tìm hiểu về thư viện Pi4J bao gồm: Các thư viện nhập xuất dữ liệu từ I2C.

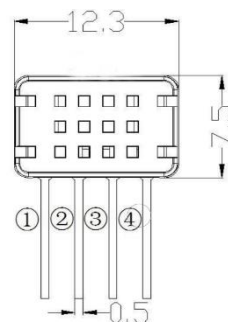
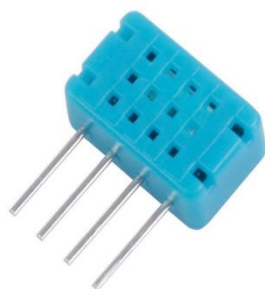
1.3. Phần cứng và phần mềm được sử dụng

1.3.1. Sensor DHT12

1.3.1.1 Tổng quan

DHT12 là cảm biến nhiệt độ và độ ẩm kỹ thuật số, là một bản ghi đầu ra kỹ thuật số đã hiệu chỉnh về nhiệt độ và độ ẩm và là một sản phẩm nâng cấp của DHT11.

1.3.1.2 Thông số kỹ thuật



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)

1.3.1.3 Nguyên lý hoạt động

DHT12 gửi và nhận dữ liệu với một dây tín hiệu **DATA**, với chuẩn dữ liệu truyền 1 dây này, chúng ta phải đảm bảo sao cho ở chế độ chờ (idle) dây DATA có giá trị ở mức cao, nên trong mạch sử dụng DHT12, dây DATA phải được mắc với một trở kéo bên ngoài (thông thường giá trị là $4.7k\Omega$).

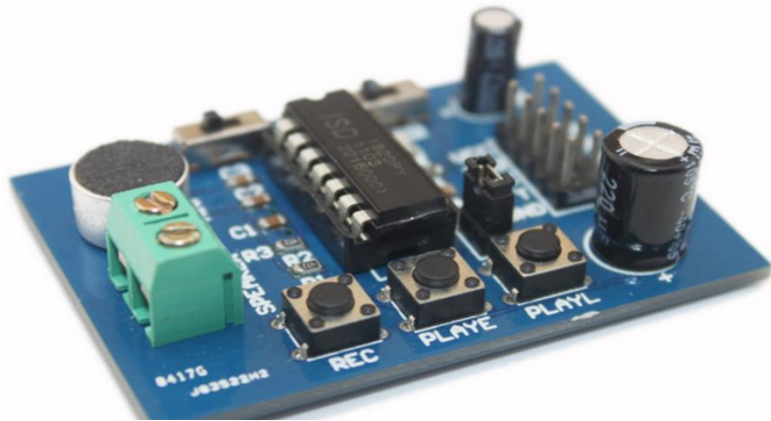
Dữ liệu truyền về của DHT12 gồm 40 bit dữ liệu theo thứ tự: 8 bit biểu thị phần nguyên của độ ẩm + 8 bit biểu thị phần thập phân của độ ẩm + 8 bit biểu thị phần nguyên của nhiệt độ + 8 bit biểu thị phần thập phân của nhiệt độ + 8 bit check sum.

1.3.2. Module ISD1820

1.3.2.1 Tổng quan

Mạch thu và phát âm thanh ISD1820 sử dụng ic ISD1820, có thể sử dụng để ghi nhiều đoạn âm thanh và phát lại. Module sử dụng loại chip ghi âm chuyên dụng, có khả năng lưu đoạn thu âm mà không bị mất đi. Một đoạn ghi âm đạt độ dài từ 8 đến 20 giây.

1.3.2.2. Thông số kỹ thuật



- IC khuếch đại 8Ω cho loa.
- Nguồn điện sử dụng 3.3V.
- Có thể điều khiển thủ công hoặc bằng vi điều khiển.
- Tần số lấy mẫu và thời gian điều chỉnh bằng điện trở.
- Ghi âm được 20s thời gian của đoạn ghi âm.
- Kích thước board 37 x 54mm.

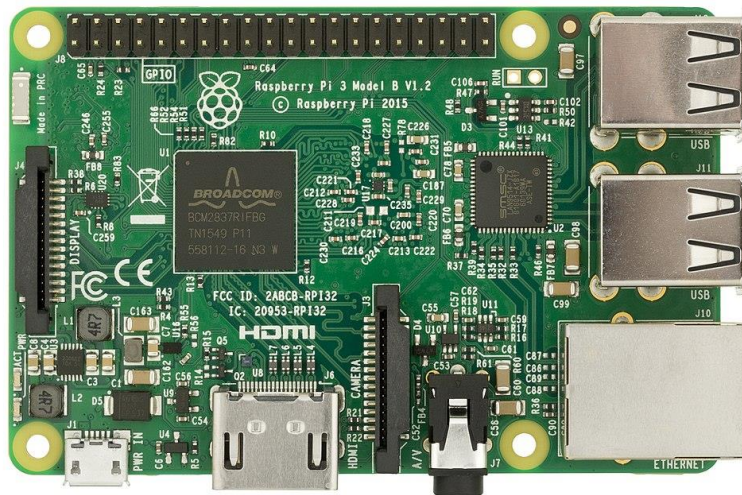
1.3.3. Raspberry Pi 3

1.3.3.1. Tổng quan














Raspberry Pi là từ để chỉ các máy tính chỉ có một board mạch (hay còn gọi là máy tính nhúng) kích thước chỉ bằng một thẻ tín dụng, được phát triển tại Anh bởi Raspberry Pi Foundation. Raspberry Pi chủ yếu sử dụng các hệ điều hành dựa trên nhân Linux.

1.3.3.2. Phần cứng

Raspberry Pi 3 với CPU ARM Cortex-A53 Quadcore 1.2GHz 64-bit, RAM 1GB và đặc biệt hỗ trợ chuẩn Wifi 802.11n cùng Bluetooth 4.1. Bên cạnh đó, Raspberry Pi 3 hoàn toàn tương thích ngược với Raspberry Pi 2.



1.3.3.3. Cổng GPIO

Raspberry Pi Model A & B (P1 Header)					
PIN #	NAME			NAME	PIN #
	3.3 VDC Power	1		2	5.0 VDC Power
8	SDA0 (I2C)	3		4	DNC
9	SCL0 (I2C)	5		6	0V (Ground)
7	GPIO 7	7		8	TxD (UART) 15
	DNC	9		10	RxD (UART) 16
0	GPIO 0	11		12	GPIO1 1
2	GPIO2	13		14	DNC
3	GPIO3	15		16	GPIO4 4
	DNC	17		18	GPIO5 5
12	MOSI	19		20	DNC
13	MISO	21		22	GPIO6 6
14	SCLK	23		24	CE0 10
	DNC	25		26	CE1 11
<p>Attention! The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.</p> <p>http://www.pi4j.com</p>					

1.3.4. Pi4J Library

1.3.4.1. Tổng quan

Pi4J cung cấp một cầu nối giữa phần cứng và Java để có quyền truy cập vào Raspberry Pi với cách tiếp cận hướng đối tượng thân thiện với Java. Pi4J là một dự án mã nguồn mở được phát triển bởi các kỹ sư phần mềm chuyên nghiệp. Ngoài chức năng truy cập phần cứng thô cơ bản, dự án này cũng cố gắng cung cấp một tập hợp các tính năng nâng cao giúp làm việc với Raspberry Pi trở thành trải nghiệm dễ thực hiện và thuận tiện hơn cho các nhà phát triển Java.

1.3.4.2. Một số tính năng cơ bản

NUM_PINS: Tổng số chân GPIO có sẵn trong thư viện Pi4J.

INPUT: Chân GPIO cho hướng INPUT để đọc trạng thái chân.

OUTPUT: Chân GPIO cho hướng OUTPUT để ghi trạng thái chân.

PWM_OUTPUT: Chân GPIO cho hướng PWM_OUTPUT để ghi trạng thái chân analog.

LOW: Chân GPIO có trạng thái OFF

HIGH: Chân GPIO có trạng thái ON

wiringPiSetup: Gọi sơ đồ đánh số chân wiringPi ánh xạ từ số pin ảo sang chân GPIO của Raspberry Pi.

wiringPiSetupGpio: Gọi sử dụng trực tiếp số chân GPIO của Raspberry Pi mà không cần phải ánh xạ.

GpioController: Tạo bảng điều khiển GPIO mới.

GpioPinDigitalOutput: Cung cấp chân GPIO làm chân đầu ra.

GpioPinDigitalInput: Cung cấp chân GPIO làm chân đầu vào.

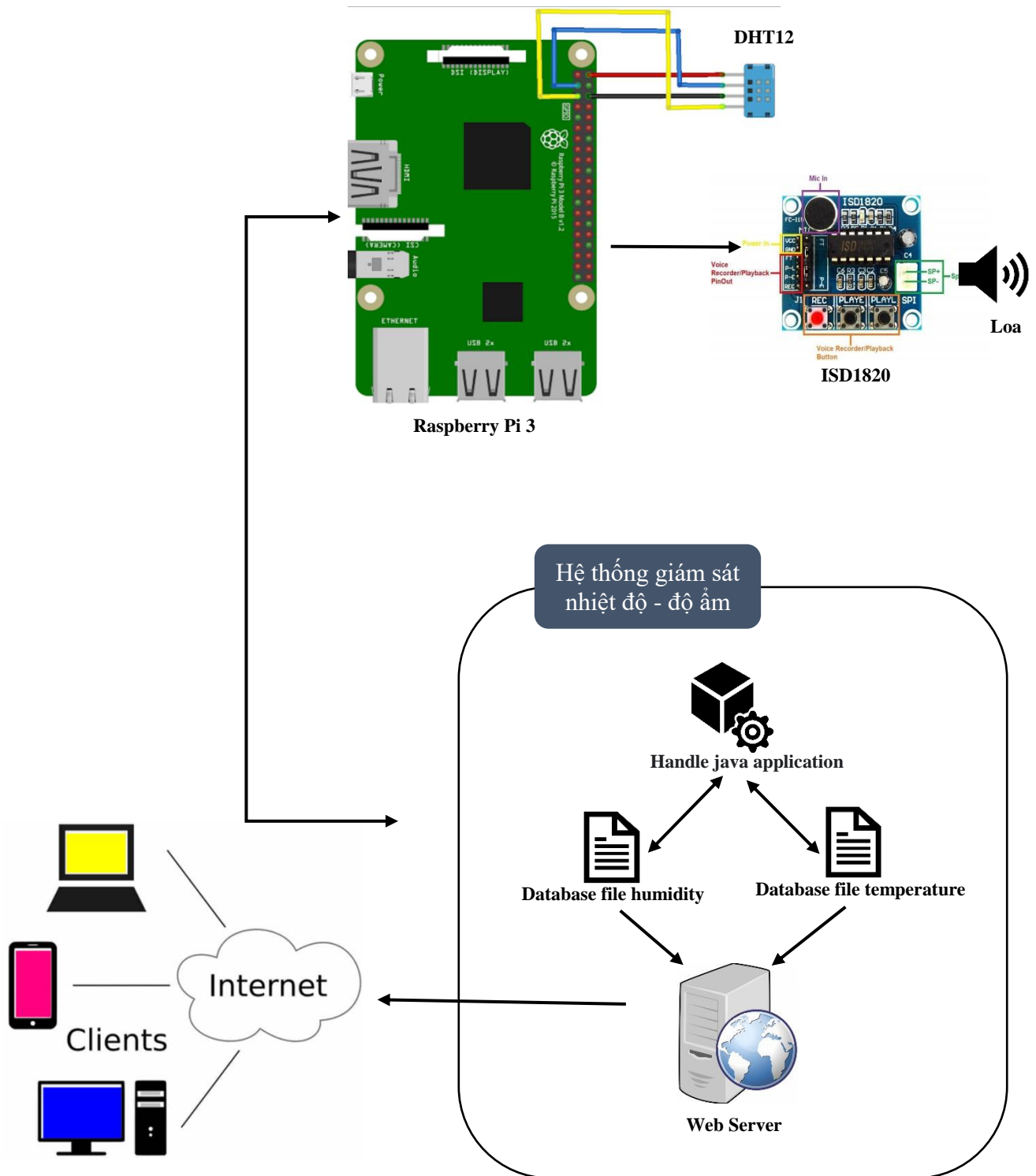
setState: Kiểm soát thiết lập trạng thái GPIO.

(tham khảo thêm: <https://www.pi4j.com/1.2/usage.html>)

Chương 2: Thiết Kế Giải Pháp

1. Kiến trúc hệ thống:

Mô hình tổng thể các thành phần trong hệ thống.



2. Thiết kế giải thuật xử lý

- Bắt đầu xây dựng phần mềm giám sát nhiệt độ - độ ẩm ta tạo một project với tên class là **NLCS_DHT12**.

- Khai báo thư viện gồm:

```
+ import com.pi4j.io.*;  
+ import java.util.Date;  
+ import java.io.FileWriter;  
+ import java.io.*;  
+ import java.text.SimpleDateFormat;  
+ import java.util.ArrayList;
```

- Định nghĩa và khai báo cổng GPIO:

```
+ I2CBus bus = I2CFactory.getInstance(I2CBus.BUS_1);  
+ final GpioController gpio = GpioFactory.getInstance();  
+ final GpioPinDigitalOutput pin=gpio.provisionDigitalOutputPin  
(RaspiPin.GPIO_07, "MyLED", PinState.HIGH) ;
```

- Cho vòng lặp **while(true)** bao hàm lại các đoạn code ở dưới để cho chương trình chạy liên tục.

- Khai báo hàm `SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");` để lấy được thông tin ngày và giờ hiện tại.

- Khai báo hàm đọc file khi để đưa dữ liệu từ file txt lên chương trình:

```
FileReader fr2 = new FileReader("/var/www/html/filedoam.txt");  
BufferedReader br2 = new BufferedReader(fr2);
```

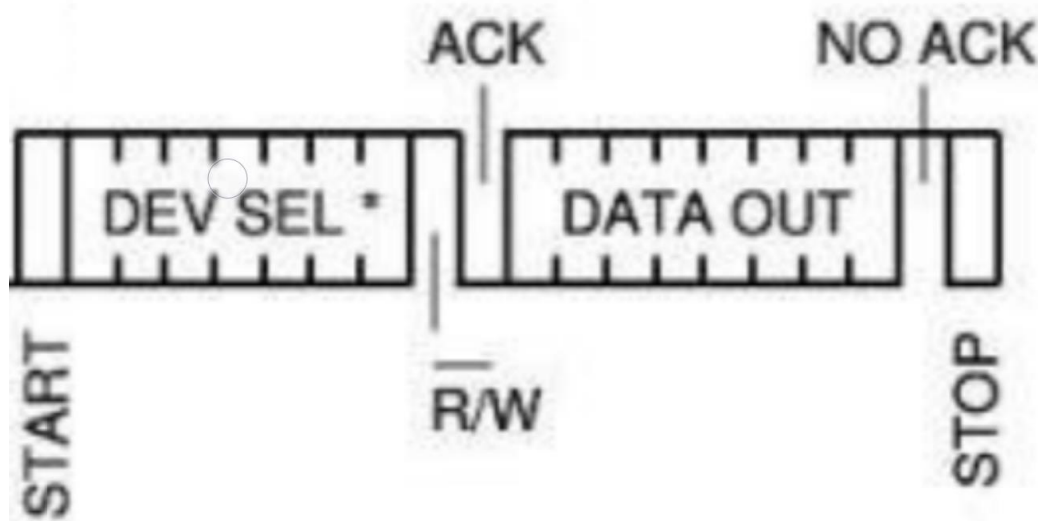
- Khai báo hàm ghi file để lưu dữ liệu thành một file:

```
+FileWriter fw = new FileWriter("/var/www/html/filenhietdo.txt",true);  
- Đặt điều kiện if() lưu dữ liệu nhiệt độ và độ ẩm theo thứ tự giờ vào file.  
- Đặt điều kiện if(nhietdoint > 40 | nhietdoint<=0) nếu thỏa điều kiện thì tính hiệu trên cổng GPIO sẽ mở và xuất tính hiệu.
```

- Đặt ngưỡng thời gian nghỉ của chương trình: **Thread.sleep();**

3. Thiết kế

3.1. Giao thức I2C



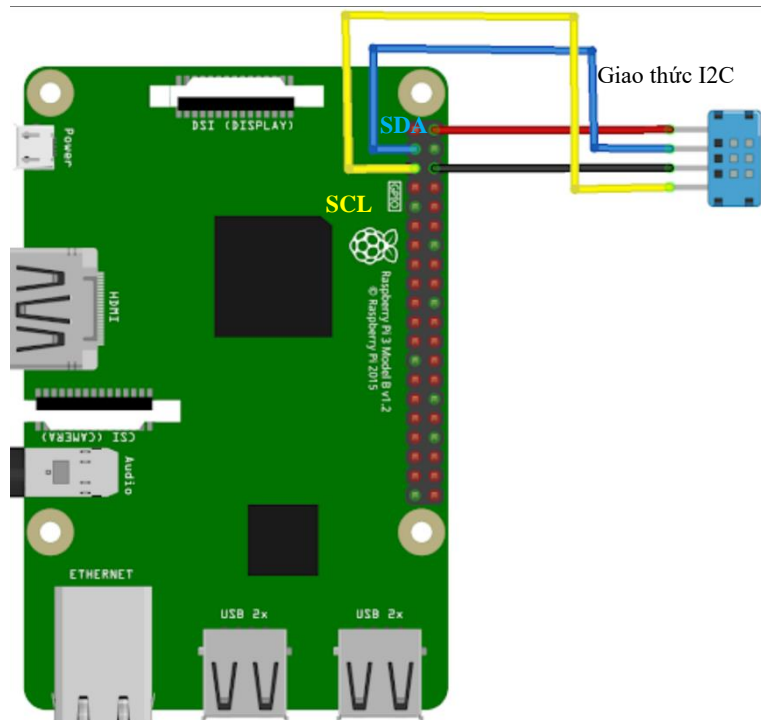
Cách bước Raspberry Pi thông qua giao thức I2C để đọc hoặc ghi dữ liệu từ các Slave:

Bước 1: Raspberry Pi sẽ phát ra một khung bắt đầu (START) để cho các Slave biết mình chuẩn bị truyền dữ liệu.

Bước 2: Raspberry Pi sẽ phát ra một khung gồm 7 bit trong đó là địa chỉ của Slave mà Raspberry Pi muốn kết nối tới và kèm theo 1 bit theo sau nếu bit đó có giá trị là 0 thì Raspberry Pi sẽ gửi dữ liệu đến các Slave còn ngược lại là bit 1 thì Raspberry sẽ đọc dữ liệu của Slave. Ở đây khung dữ liệu địa chỉ gồm 7 bit cho nên có thể có 2^7 thiết bị Slave kết nối tới Raspberry Pi, mỗi Slave sẽ có một địa chỉ riêng do nhà sản xuất mặc định trên đó.

Bước 3: Nếu trong đường truyền không có thiết bị Slave nào có trùng địa chỉ mà Raspberry phát ra thì các thiết bị sẽ có khung bit 1 (NO ACK) sẽ trả về cho Raspberry, ngay sau đó Raspberry phát ra khung STOP để dừng việc nối kết. Ngược lại nếu có thiết bị Slave trùng với địa chỉ mà Raspberry vừa truyền thì ngay sau khi truyền địa chỉ 7 bit và 1 bit R/W (đọc hoặc ghi) thì các thiết bị có địa chỉ trùng là phát ra bit 0 (ACK) để xác nhận và sau đó Slave sẽ truyền dữ liệu, sau khi truyền thành công dữ liệu Slave sẽ phát ra bit 0 (ACK) để cho Raspberry biết đã truyền xong và cuối cùng Raspberry phát khung STOP để dừng việc kết nối.

3.2. Raspberry Pi đọc dữ liệu từ DHT12



Raspberry Pi có nhiệm vụ đọc dữ liệu bằng giao thức I2C từ chân SDA của sensor DHT12 vào chân GPIO_08 của Raspberry Pi. Các bước tiến hành đọc dữ liệu:

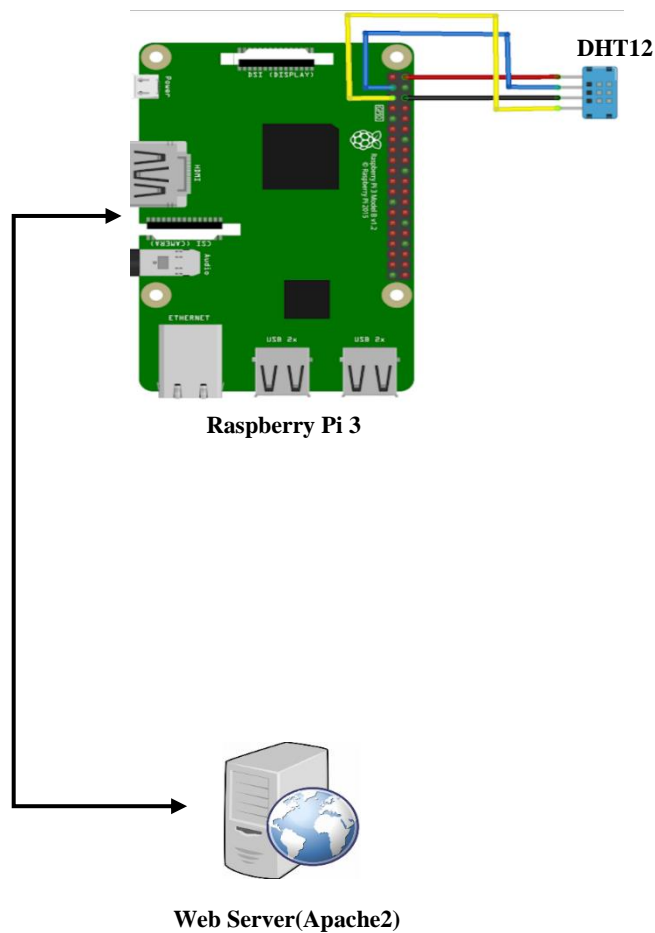
Bước 1: Raspberry Pi sẽ phát ra một khung bắt đầu(START) để cho sensor DHT12 biết mình chuẩn bị truyền dữ liệu.

Bước 2: Raspberry Pi sẽ phát ra một khung gồm **7 bit** trong là địa chỉ của sensor DHT12, trong trường hợp này Raspberry Pi đã phát ra địa chỉ cơ số thập lục phân(**0x5c**) chuyển thành nhị phân(**1011100**). Do địa chỉ “**0x5c**” là do nhà sản xuất mặc định trên sensor DHT12 nên Raspberry muốn giao tiếp được với DHT12 thông qua giao tiếp I2C thì chỉ có một địa chỉ (**0x5c**). Sau khi gửi địa chỉ thì Raspberry gửi tiếp **bit 1** để cho biết Raspberry sẽ đọc(R) dữ liệu.

Bước 3: Sensor sẽ phát ra **bit 0(ACK)** để xác nhận, ngay sau đó sensor DHT12 sẽ truyền liên tục **40 bit** dữ liệu theo thứ tự: **8 bit** biểu thị phần nguyên của độ ẩm + **8 bit** biểu thị phần thập phân của độ ẩm + **8 bit** biểu thị phần nguyên của nhiệt độ + **8 bit** biểu thị phần thập phân của nhiệt độ + **8 bit** check sum.

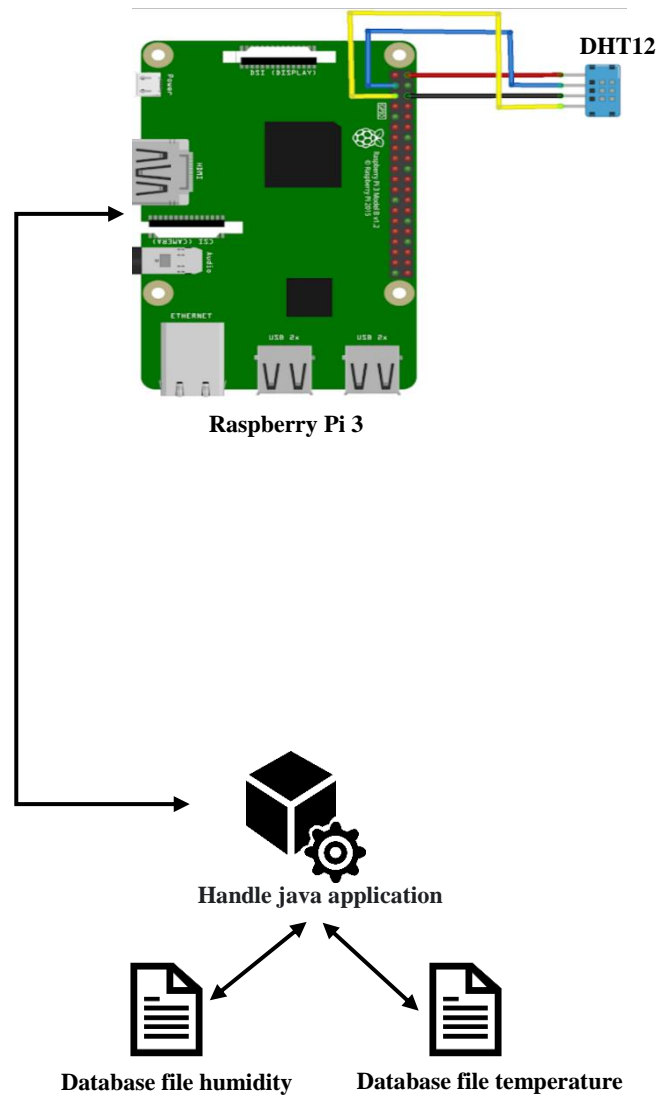
Bước 4: Sau khi truyền xong 40 bit dữ liệu, DHT12 sẽ phát ra một **bit 0(ACK)** để cho Raspberry biết đã truyền xong dữ liệu và cuối cùng Raspberry lưu **40 bit** dữ liệu và phát ra khung STOP để dừng việc kết nối.

3.3. Apache2 chạy trên Raspberry Pi



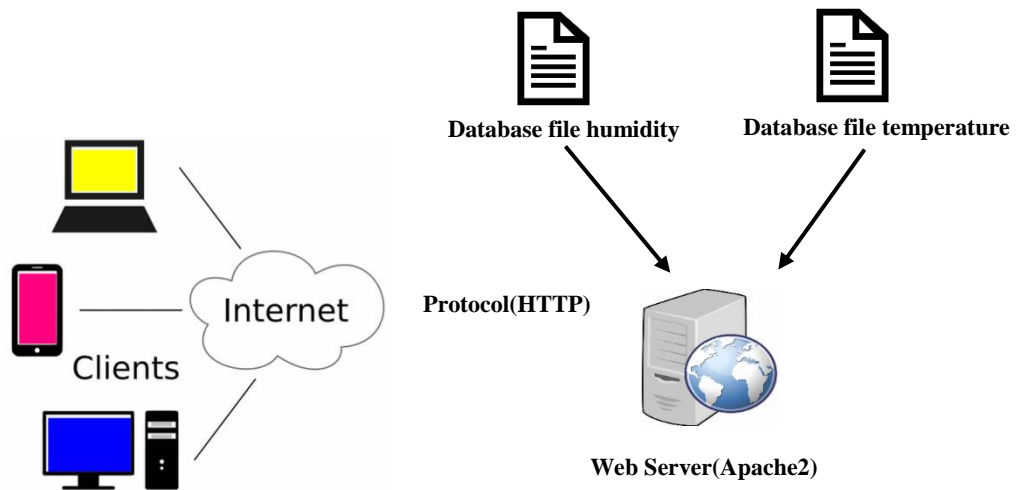
Triển khai Apache2 trên Raspberry Pi để xây dựng Web Server để thuận tiện giúp ta truyền tải dữ liệu nhiệt độ và độ ẩm lên biểu đồ trên website.

3.4. Xử lý và lưu trữ dữ liệu



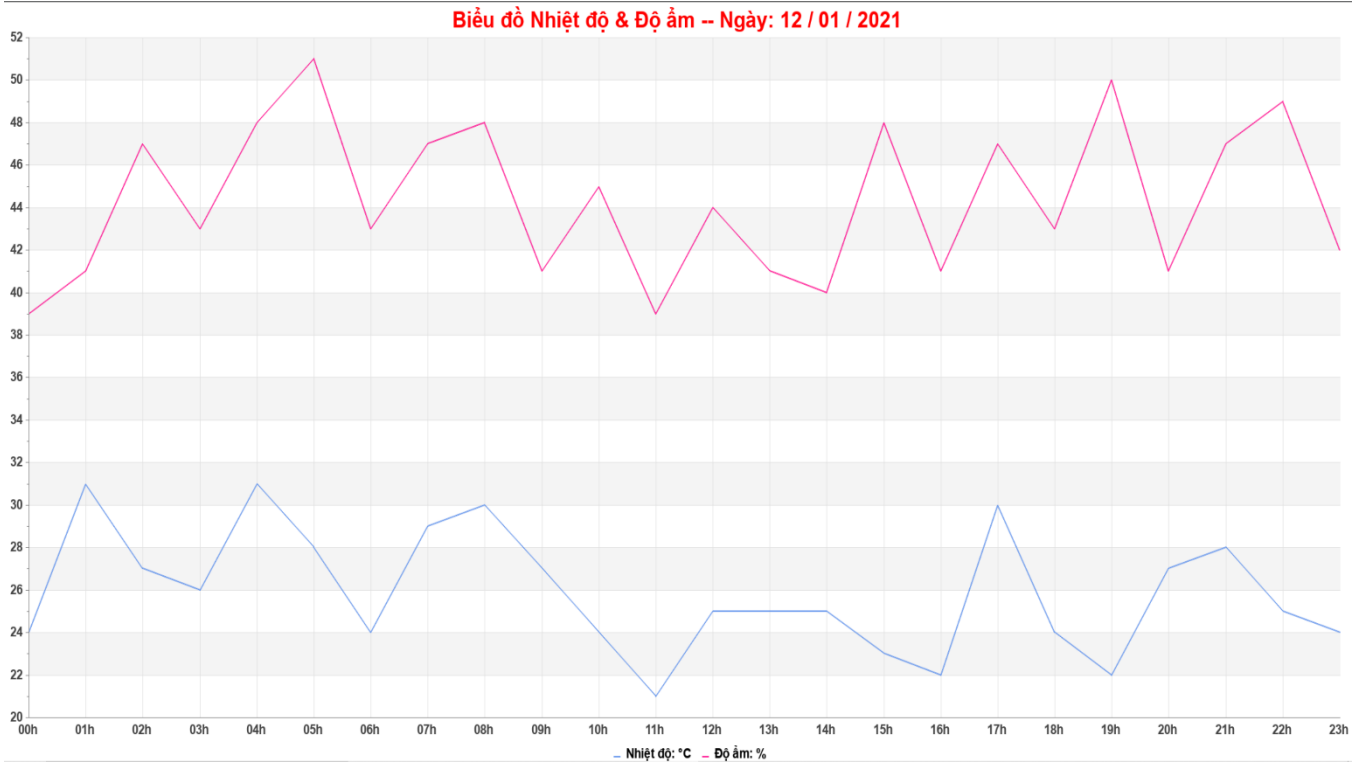
Phần mềm Java cứ sau mỗi 30 giây sẽ nhận thông tin nhiệt độ và độ ẩm từ DHT12 trả về cho Raspberry Pi, sau đó lưu giá trị nhiệt độ và độ ẩm vào biến và lưu giá trị vào trong file txt. Nếu nhiệt độ vượt quá ngưỡng cho phép thì loa sẽ báo động.

3.5. Đọc dữ liệu lưu trữ lên Web Server



Đọc dữ liệu từ file nhiệt độ và file độ ẩm lên PHP xử lý. Sau đó hiển thị biểu đồ nhiệt độ và độ ẩm bằng thư viện Jpgraph.

3.6 Biểu diễn dữ liệu lên biểu đồ Website



Lấy dữ liệu từ file nhiệt độ và file độ ẩm sau đó biểu diễn lên trang web bằng những đường line. Biểu đồ được biểu diễn gồm 23 cột(trục tung) tương ứng với số giờ trong ngày và giá trị nhiệt độ và độ ẩm được biểu diễn bằng các hàng(trục hoành) với giá trị thay đổi theo thời gian thực, tùy thuộc vào nhiệt độ và độ ẩm có giá trị max là bao nhiêu thì trục hoành sẽ cho giá trị max bằng với giá trị max của nhiệt độ và độ ẩm.

Chương 3: Giải Pháp Thiết Kế

1. Đọc và lưu vào mảng thông qua giao thức I2C

```
dev = bus.getDevice(0x5c);  
byte data[] = new byte[5];  
dev.read(0x00, data, 0, 5);  
...
```

- Khai báo mới một mảng byte **data[]** có kích thước 5 byte.
- Hàm **bus.getDevice(0x5c)**: Lúc này Raspberry Pi sẽ kết nối với DHT12 thông qua truyền địa chỉ **0x5c** lên giao tiếp I2C, sau đó DHT12 gửi dữ liệu cho Raspberry để lưu vào **data**, bắt đầu từ bộ đệm dữ liệu **0x00** được đọc từ thiết bị I2C trong một lần và sau đó bù đắp trong bộ đệm bằng **0** và số size được đọc là **5** byte.

2. Nối mảng byte để trả kết quả nhiệt độ và 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("Nhiệt Độ : "+ nhietdo +" °C " + "--- Độ ẩm : "+ doam + " %");  
  
}  
else  
  
    System.out.println("Không nhận được Data");
```

- vì mảng byte **data[4]** là kết quả gộp lại từ mảng **data[0] + data[1] + data[2] + data[3]**, nếu kết ra bằng nhau thì sẽ cộng mảng **data[0]** với **data[1]** để có được kết quả độ ẩm và **data[2] + data[3]** sẽ có được kết quả nhiệt độ.

3. Hàm trả kết quả ngày giờ

```
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("Ngày: " + ngay + "--Gio: " + gio);
int gioint = Integer.parseInt(gio);
```

Chúng ta khai báo hàm nhận thời gian gồm: SimpleDateFormat và hàm Date. Sau đó chúng ta chỉ lấy giá trị ngày và giá trị giờ chúng ta đổi sang kiểu Integer(**int gioint = Integer.parseInt(gio)**) để sau này chúng ta sẽ lưu giá trị nhiệt độ và độ ẩm thông qua dùng giờ để làm số thứ tự của mảng nhiệt độ và độ ẩm.

4. Hàm ghi dữ liệu vào file txt

```
//Ghi file
FileWriter fw = new FileWriter("/var/www/html/filenhietdo.txt",true);
FileWriter fw2 = new FileWriter("/var/www/html/filedoam.txt",true);
```

Dùng hàm FileWrite để ghi dữ liệu nhiệt độ và độ ẩm vào hai file txt lần lượt là **filenhietdo.txt** và **filedoam.txt**.

5. Hàm điều kiện vượt ngưỡng cho phép

```
if(nhietdoint > 40 | nhietdoint<=0) {
    try{
        System.out.println("Canh bao!! Nhiệt độ vượt qua ngưỡng cho phép");
        pin.high();
        Thread.sleep(15000);
        // turn off gpio pin #01
        pin.low();
        Thread.sleep(15000);
    }//close try
    catch(Exception e){
        System.out.println("Lỗi nhập xuất!");
    }
}
```

Nếu biến **nhietdoint** lúc này trả về giá trị lớn hơn 40 °C hoặc nhỏ hơn hoặc bằng 0 °C thì tính hiệu sẽ mở lên trên chân GPIO_07 và sau đó được kích lên module ISD 1820 sẽ phát ra báo động.

6. Đọc dữ liệu từ file txt xử lý trên Web Server

```
//Doc File
$file1 = "./filenhietdo.txt";
$file2 = "./filedoam.txt";
$file1s = file_get_contents($file1);
$file2s = file_get_contents($file2);
$line1s = explode("\n", $file1s);

foreach($line1s as $newline1[]) {
}

$line2s = explode("\n", $file2s);

foreach($line2s as $newline2[]) {
}
```

Lúc này file php sẽ đọc dữ liệu từ hai file nhiệt độ và độ ẩm thông qua đường dẫn **./filenhietdo.txt** và **./filedoam.txt**, sau đó được lưu vào hai mảng **newline1** và **newline2** trên php sau đó biểu diễn biểu đồ trên website thông qua thư viện Jpgraph.

Chương 4: Cài Đặt Giải Pháp

4.1. Cài đặt và quản trị hệ điều hành Raspbian trên Raspberry Pi

1. Chuẩn bị thẻ nhớ microSD có dung lượng tối thiểu 8GB và đã được định dạng.
2. Download hệ điều hành Raspbian về máy tính bằng đường link: https://downloads.raspberrypi.org/imager/imager_1.4.exe .
3. Download phần mềm balenaEtcher giải nén hệ điều hành Raspbian vào thẻ microSD.
4. Mở terminal gõ lệnh: `sudo raspi -config`, lúc này màn hình hiển thị hộp thoại Raspberry Software Configuration Tool -> Interfacing Options -> SSH để có thể mở port 22 để cho phép truy cập từ xa vào Raspberry Pi bằng dòng lệnh và mở port 5900 để truy cập từ xa bằng giao diện GUI.
5. Truy cập từ xa ssh từ máy trạm: **ssh pi@IP** của Raspberry và nhập mật khẩu là raspberry .

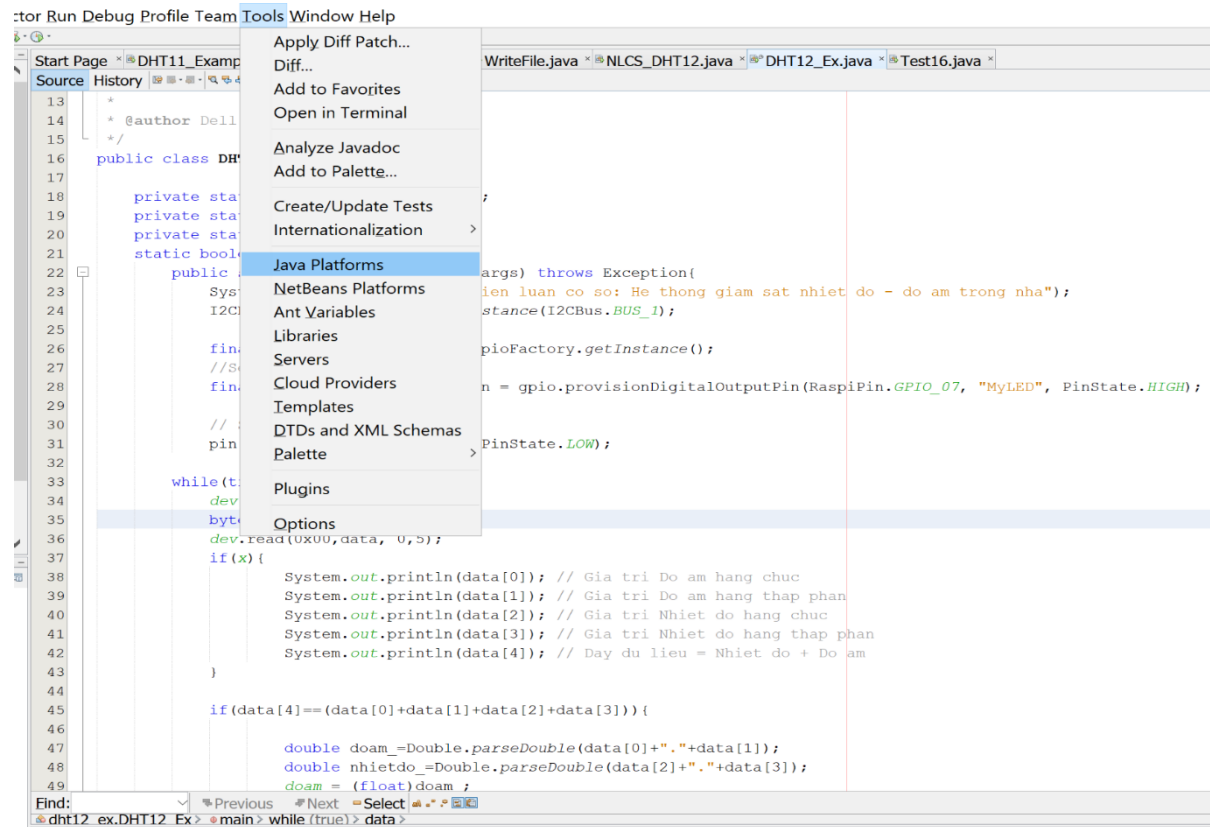
4.2. Thiết lập môi trường phát triển ứng dụng Java trên Raspberry Pi

1. Khi kết nối được Raspberry Pi bằng SSH, tiếp theo sử dụng lệnh: **sudo apt update** để cập nhật lại hệ điều hành.
2. Cài đặt biến môi trường Java trên Raspberry Pi: **sudo apt-get install default-jdk** .
3. Kiểm tra xem có phải phiên bản Java 1.8 không bằng lệnh: **sudo update-alternatives --display java** .

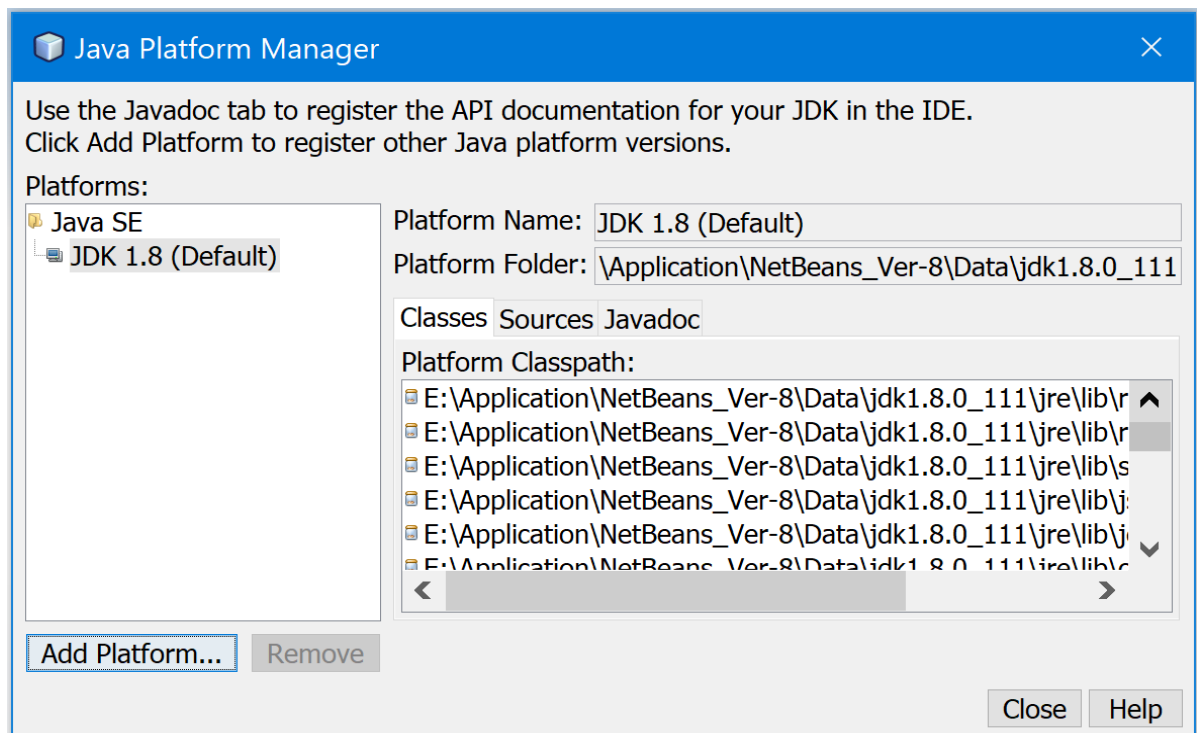
4.3. Cài đặt NetBeans trên máy trạm

1. Cài đặt **netbeans 8.2** , sau đó tải thư viện **pi4j-1.2** và **JDK 1.8** để có thể thiết lập được môi trường chung giữa NetBeans 8.2 và **Raspberry Pi**.
2. Tạo 1 Project mới trên **NetBeans 8.2** và lập trình 1 đoạn code để điều khiển (Đèn Led) sau đó mở thanh công cụ **tools**.

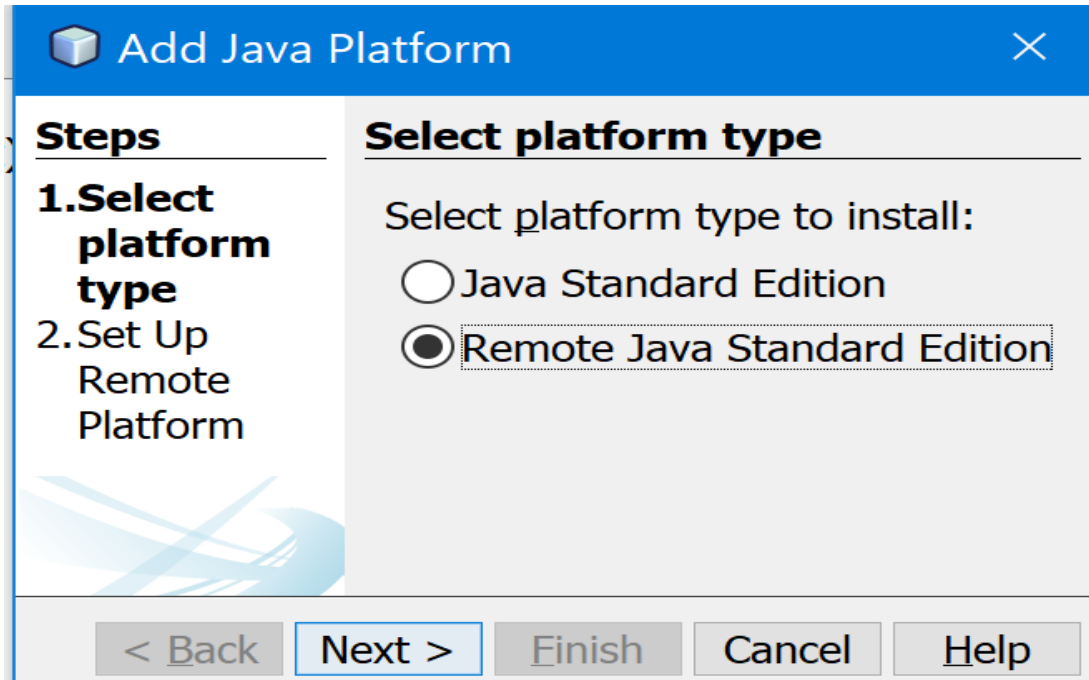
3. Chọn Java Platform



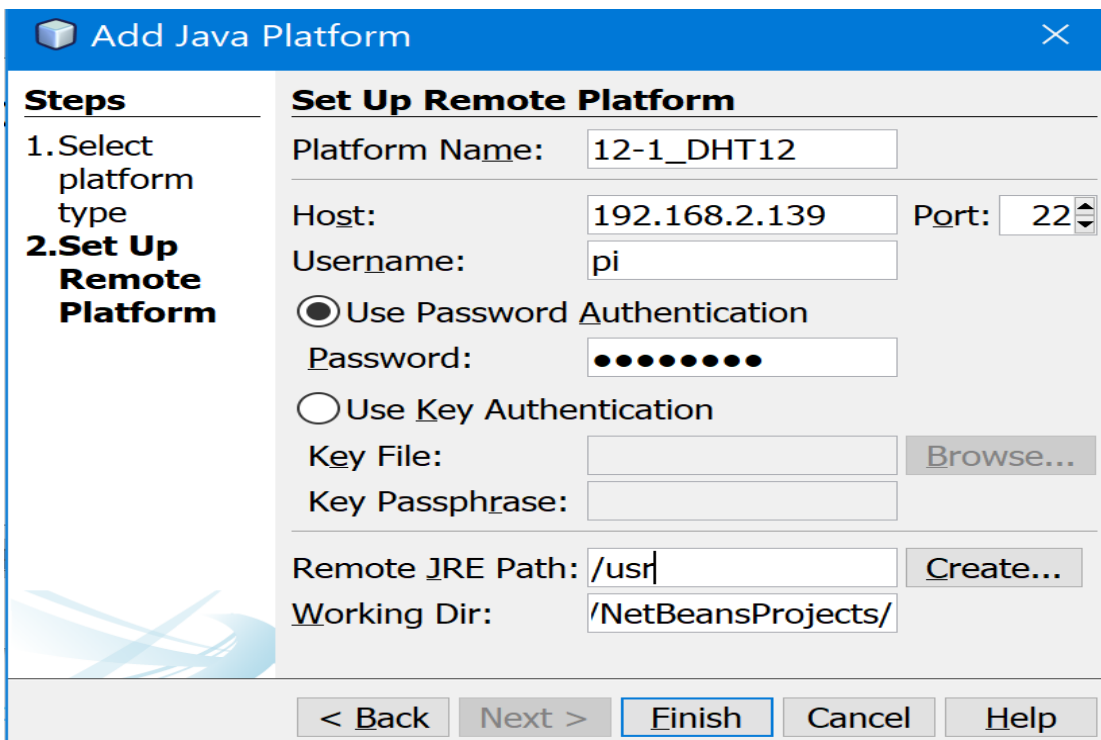
4. Chọn Add Platform .



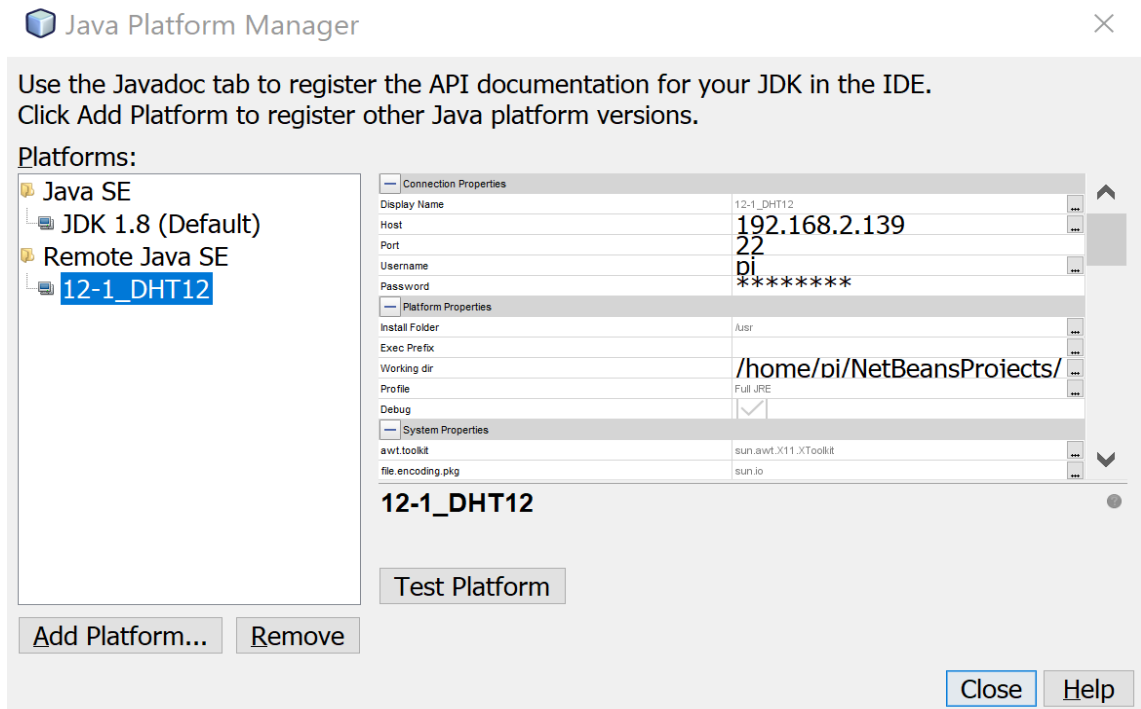
5. Chọn **Remote Java Standard Edition**, ấn **Next** .



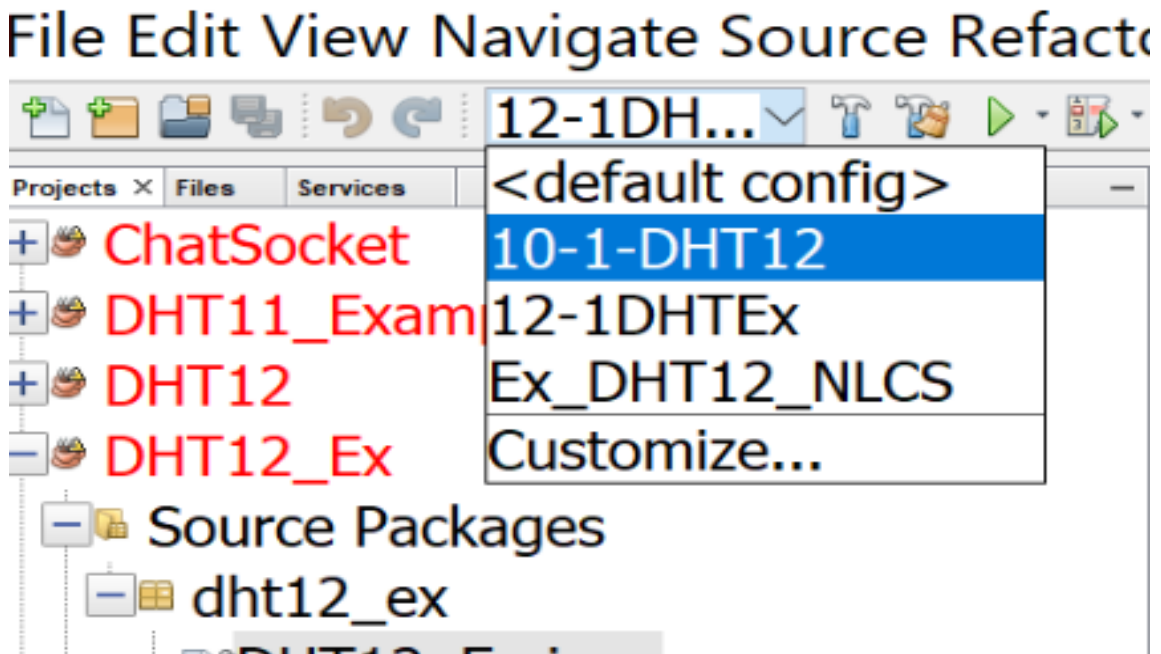
6. Điền thông tin của **Raspberry Pi** sau đó **Finish** .



7. Kiểm tra xem NetBeans đã kết nối thành công với Raspberry Pi chưa bằng cách nhấn **Test Platform**.

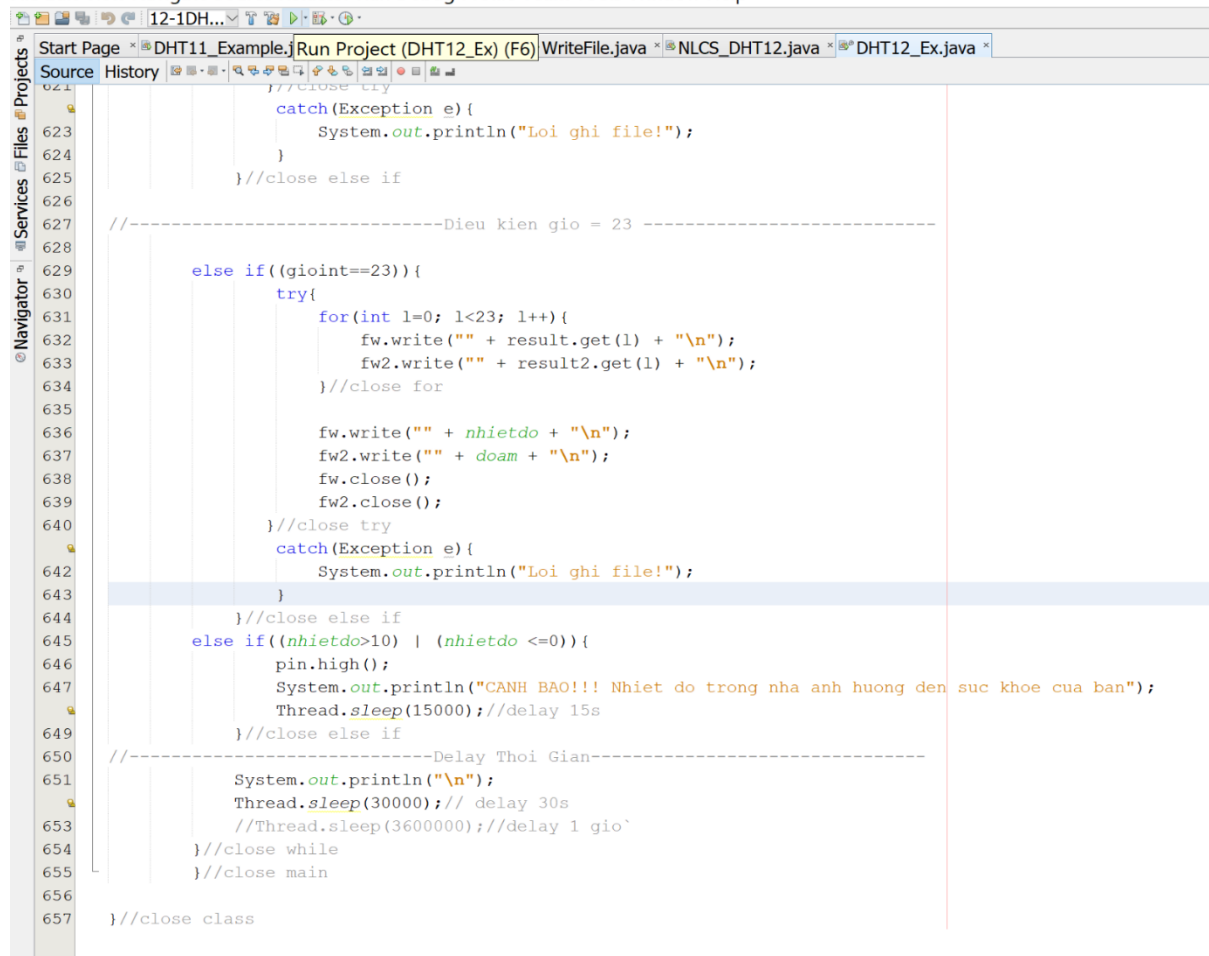


8. Vào thanh **default config** để chọn **Platform** đã tạo.



9. Sau đó Run.

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help



```
621 //close try
622     catch(Exception e){
623         System.out.println("Loi ghi file!");
624     }
625 } //close else if
626
627 //-----Dieu kien gio = 23 -----
628
629     else if((gioint==23)){
630         try{
631             for(int l=0; l<23; l++){
632                 fw.write("" + result.get(l) + "\n");
633                 fw2.write("" + result2.get(l) + "\n");
634             } //close for
635
636             fw.write("" + nhietdo + "\n");
637             fw2.write("" + doam + "\n");
638             fw.close();
639             fw2.close();
640         } //close try
641         catch(Exception e){
642             System.out.println("Loi ghi file!");
643         }
644     } //close else if
645     else if((nhietdo>10) | (nhietdo <=0)){
646         pin.high();
647         System.out.println("CANH BAO!!! Nhiet do trong nha anh huong den suc khoe cua ban");
648         Thread.sleep(15000); //delay 15s
649     } //close else if
650
651 //-----Delay Thoi Gian-----
652     System.out.println("\n");
653     Thread.sleep(30000); // delay 30s
654     //Thread.sleep(3600000); //delay 1 gio`
655 } //close while
656 } //close main
657
658 } //close class
```

10. Khi kết nối vào **Raspberry Pi** dùng lệnh **ls** xem có thư mục **NetBeansProjects** hay chưa.

```
C:\WINDOWS\system32>ssh pi@192.168.1.128
ssh: connect to host 192.168.1.128 port 22: Connection timed out

C:\WINDOWS\system32>ssh pi@192.168.1.137
The authenticity of host '192.168.1.137 (192.168.1.137)' can't be established.
ECDSA key fingerprint is SHA256:BGzAQ2sHm3oQuA9UeYYL10zW5l+qizSea0No7oA7xgE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.137' (ECDSA) to the list of known hosts.
pi@192.168.1.137's password:
Linux xbee1 5.4.51-v7+ #1333 SMP Mon Aug 10 16:45:19 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 10 16:22:22 2020

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@xbee1:~$ ls
Bookshelf  Documents  Java       Music      NetBeansProjects  pi4led  Public  Templates  testing  Truong_code
Desktop    Downloads  mkdir.save nano.save  PHONG       Pictures Send.java test      test.java Videos
pi@xbee1:~$
```

11. Nếu có, dùng lệnh **cd /NetBeansProjects/NLCS_DHT12/dist** để mở thư mục đã chuyển từ NetBean8.2 vào **Raspberry Pi**.

```
pi@HuyGao:~/NetBeansProjects/NLCS_DHT12/dist$ ls
filedoam.txt  filenhietdo.txt  lib  NLCS_DHT12.jar  README.TXT
pi@HuyGao:~/NetBeansProjects/NLCS_DHT12/dist$
```

12. Tiếp theo dùng lệnh **java -jar NLCS_DHT12.jar** để chạy code đã chuyển từ NetBean 8.2 vào **Raspberry Pi**.

```
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
Nhiệt Độ : 25.8 °C --- Độ ẩm : 40.1 %
Ngày: 12/01/2021--Gio: 13
```

4.4. Cài đặt Apache2 và PHP trên Raspberry Pi 3

1. Mở Terminal lên và gõ lệnh: **sudo apt install apache2 -y** để cài đặt Apache2.
2. Gõ tiếp lệnh: **sudo apt install -y php7.4** để cài đặt phiên bản php chạy trên Raspbian.
3. Truy cập vào đường dẫn **/var/www/html** xem có tồn tại hay không.

```
pi@HuyGao:/var/www/html $ ls
doam.txt          filenhietdo.txt      mywebpage.php      testphptojs.php
filedoam_old.txt  index.html           nhietdo.txt
filedoam.txt      jpgraph              NLCS_Line.php
filenhietdo_old.txt jpgraph-4.3.4.tar.gz testline.php
pi@HuyGao:/var/www/html $
```


4.5. Cài đặt Jpgraph Library

1. Truy cập vào link: <https://jpgraph.net/> để download thư viện Jpgraph.
2. Vào thư mục download copy file jpgraph vào đường dẫn `/var/www/html` rồi ta dùng lệnh: **sudo cp jpgraph-4.3.4.tar.gz /var/www/html.**

```
login as: pi
pi@192.168.43.56's password:
Linux HuyGao 4.19.75-v7+ #1270 SMP Tue Sep 24 18:45:11 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 12 15:17:35 2021
pi@HuyGao:~$ whoami
pi
pi@HuyGao:~$ cd Downloads/
pi@HuyGao:~/Downloads$ ls
arialbd.ttf          NLCS_Line.php      pi4j-1.2.zip
jpgraph-4.3.4.tar.gz pi4j-1.2           testphptojs.php
pi@HuyGao:~/Downloads$ sudo cp jpgraph-4.3.4.tar.gz /var/www/html
```

3. Vào đường dẫn vừa copy xong rồi giải nén file jpgraph bằng lệnh: **sudo tar -zxvf jpgraph-4.3.4.tar.gz**

```
login as: pi
pi@192.168.43.56's password:
Linux HuyGao 4.19.75-v7+ #1270 SMP Tue Sep 24 18:45:11 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jan 13 23:27:44 2021 from 192.168.43.235
pi@HuyGao:~$ cd /var/www/html
pi@HuyGao:/var/www/html$ ls
doam.txt          filenhietdo.txt      mywebpage.php  testphptojs.php
filedoam_old.txt  index.html           nhietdo.txt
filedoam.txt      jpgraph              NLCS_Line.php
filenhietdo_old.txt jpgraph-4.3.4.tar.gz testline.php
pi@HuyGao:/var/www/html$ sudo tar -zxvf jpgraph-4.3.4.tar.gz
```

4.6. Tạo file TXT để lưu trữ dữ liệu nhiệt độ và độ ẩm

1. Ta lần lượt dụng lệnh: **touch filenhietdo.txt** và **touch filedoam.txt** để tạo ra hai file để lưu trữ dữ liệu.

```
pi@HuyGao:/var/www/html $ touch filenhietdo.txt
```

CHƯƠNG 5: Đánh Giá Và Kiểm Thử

1. 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.

2. Kiểm thử

Mô hình giám sát nhiệt độ - độ ẩm trong nhà có kết quả chính xác tuyệt đối về nhiệt độ và độ ẩm, không xảy ra hiện tượng lỗi và độ trễ. Đạt được mục tiêu đề ra đo nhiệt độ và độ ẩm trong nhà và phát loa báo động khi nhiệt độ vượt quá ngưỡng cho phép.

PHẦN III. KẾT LUẬN

1. Kết quả đạt được

Mô hình giám sát nhiệt độ - độ ẩm trong nhà đã cho kết quả chính xác, chu trình của mô hình khép kín hoạt động liên tục, dữ liệu nhiệt độ và độ ẩm được cập nhật theo thời gian thực sau mỗi 30 giây.

Giao diện web trực quan có biểu đồ hiển thị nhiệt độ và độ ẩm theo thời gian thực, giúp người dùng dễ dàng 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.

2. Hướng phát triển

Tìm hiểu sâu hơn về lập trình IoT bằng ngôn ngữ Java trên Raspberry Pi cũng như các ngôn ngữ lập trình khác.

Mở rộng mô hình xây dựng một nhà thông minh hoàn chỉnh.

Tìm hiểu và xây dựng hệ thống quản lý các thiết bị trong nhà một cách thông minh.

Xây dựng cơ sở dữ liệu lưu trữ trên đám mây giúp dễ dàng mở rộng và giảm bớt xử lý cho server.

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>