

## NGHIÊN CỨU XÂY DỰNG MÔ HÌNH NHÀ THÔNG MINH SỬ DỤNG RASPBERRY PI 3

Hoàng Đại Long\*, Trương Quốc Khánh,  
Nguyễn Hoàng Thanh Tùng, Nguyễn Văn Ân

Khoa Điện, Điện tử và Công nghệ Vật liệu, Trường Đại học Khoa học, Đại học Huế

\*Email: longhoang@husc.edu.vn

Ngày nhận bài: 14/6/2025; ngày hoàn thành phản biện: 22/6/2025; ngày duyệt đăng: 01/10/2025

### TÓM TẮT

Nghiên cứu này trình bày việc thiết kế và chế tạo mô hình nhà thông minh (smarthome) dựa trên nền tảng Internet vạn vật (IoT). Hệ thống sử dụng Raspberry Pi 3 làm bộ xử lý trung tâm, kết nối với các khối cảm biến (DHT11, MQ-2) và cơ cấu chấp hành (Relay, Servo) để giám sát môi trường và điều khiển thiết bị. Nền tảng Blynk được sử dụng để xây dựng ứng dụng di động, cho phép người dùng điều khiển từ xa các thiết bị như đèn, quạt, cửa và nhận thông số môi trường theo thời gian thực. Hệ thống cũng tích hợp thành công chức năng cảnh báo rò rỉ khí gas qua ứng dụng và còi báo động. Kết quả thực nghiệm cho thấy mô hình hoạt động ổn định, đáp ứng nhanh, cung cấp một giải pháp hiệu quả và chi phí thấp cho việc tự động hóa nhà ở.

**Từ khóa:** Smarthouse, IoT, Raspberry Pi.

### 1. MỞ ĐẦU

Ngày nay, Internet vạn vật - Internet of Things (IoT) là một xu hướng công nghệ cho phép việc kết nối và thu thập thông tin giữa các thiết bị được thực hiện mọi lúc, mọi nơi. IoT đang được nghiên cứu, triển khai ở rất nhiều lĩnh vực khác nhau như nhà thông minh, thành phố thông minh, nông nghiệp thông minh, chăm sóc sức khỏe thông minh.

Đặc biệt, vài năm trở lại đây, khi thế giới đang dần tiến vào kỷ nguyên Internet of Things, nhà thông minh trở thành một xu hướng công nghệ tất yếu [1] [2] [3]. Trong căn hộ thông minh, các thiết bị từ rèm cửa, điều hòa, hệ thống ánh sáng, hệ thống an ninh... được kết nối với nhau và hoạt động tự động theo kịch bản lập trình sẵn.

Các nghiên cứu về nhà thông minh ở Việt Nam cũng đã được thực hiện với nhiều giải pháp nền tảng khác nhau [4] [5] [6]. Tuy nhiên, việc xây dựng một hệ thống mở, dễ

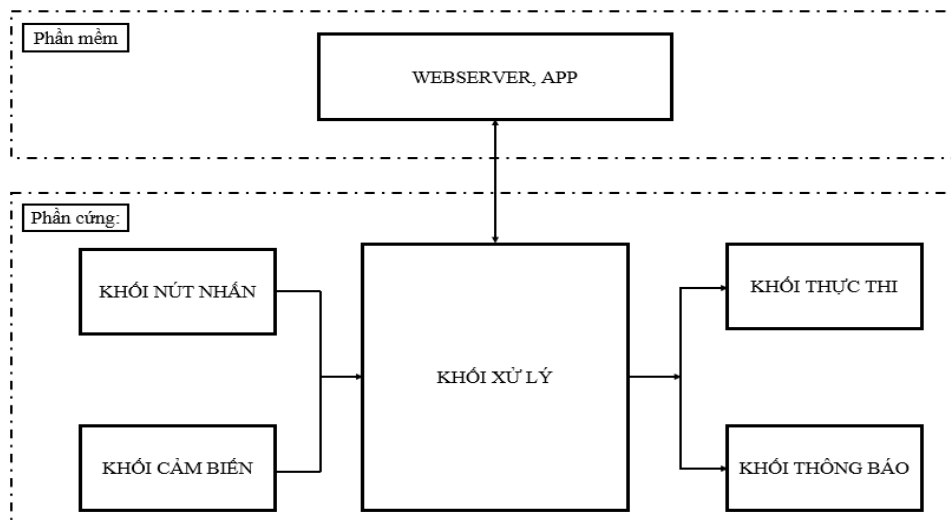
dạng tùy biến, tích hợp nhiều chức năng và có chi phí hợp lý vẫn là một bài toán cần được quan tâm. Raspberry Pi, với khả năng xử lý mạnh mẽ của một máy tính nhúng và hệ điều hành Linux, cung cấp một nền tảng lý tưởng để phát triển các ứng dụng IoT phức tạp. Trong nghiên cứu này, chúng tôi sử dụng Raspberry Pi 3 làm bộ xử lý trung tâm để xây dựng mô hình nhà thông minh.

Cấu trúc bài báo cáo được trình bày gồm các phần sau. Phần mở đầu giới thiệu về xây dựng hệ thống IoT. Sơ đồ khối của hệ thống được giới thiệu ở phần 2. Phần 3 giới thiệu về thiết kế phần cứng của hệ thống. Phần 4 trình bày xây dựng phần mềm điều khiển hệ thống. Kết quả xây dựng hệ thống được trình bày trong phần 5. Cuối cùng là kết luận.

## 2. GIỚI THIỆU VỀ HỆ THỐNG

Mục tiêu của nghiên cứu là xây dựng mô hình nhà thông minh có khả năng điều khiển bật/tắt các thiết bị (đèn, quạt), đóng/mở cửa, đồng thời giám sát thông tin về nhiệt độ, độ ẩm, và khí gas. Hệ thống cho phép theo dõi các thông số này từ xa và đưa ra cảnh báo khi phát hiện rò rỉ khí gas.

Dựa trên các mục tiêu đó, hệ thống được thiết kế với sơ đồ khối mô tả tại Hình 1 bao gồm các thành phần chính: Khối xử lý trung tâm, Khối cảm biến, Khối thông báo (LCD, còi), Khối nút nhấn (điều khiển tại chỗ), và Khối thực thi (Relay, Servo). Mô hình đảm bảo các thông báo và trạng thái được hiển thị đồng bộ trên màn hình LCD và ứng dụng di động.



Hình 1. Sơ đồ khối của hệ thống nhà thông minh

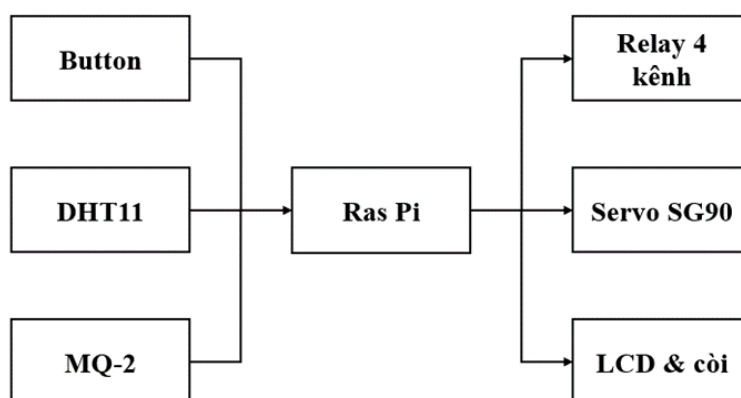
**Chức năng các khối của hệ thống:**

- ✓ Khối cảm biến : Nhận tín hiệu của cảm biến nhiệt độ và khí gas. Gửi tín hiệu cho khối xử lý.
- ✓ Khối xử lý: Là khối quan trọng nhất có chức năng phân tích, xử lý dữ liệu vào ra của hệ thống nhà thông minh. Xử lý các tín hiệu đầu vào của khối nút nhấn, khối cảm biến, webservice, app và gửi tín hiệu ra điều khiển khối thực thi và khối thông báo.
- ✓ Khối Web Server, App: Sử dụng ứng dụng Blynk để trao đổi dữ liệu với khối xử lý, nhận thông báo và điều khiển bật tắt từ xa.
- ✓ Khối thông báo : Nhận tín hiệu từ khối xử lý sau đó hiển thị thông báo qua màn hình và ứng dụng, đồng thời cảnh báo bằng âm thanh.
- ✓ Khối thực thi : Nhận tín hiệu từ khối xử lý dùng để bật tắt các thiết bị trong hệ thống nhà thông minh.
- ✓ Khối nút nhấn: Sử dụng Button để truyền tín hiệu bật/tắt các chức năng của hệ thống. Gửi tín hiệu cho khối xử lý.

### 3. THIẾT KẾ PHẦN CỨNG CỦA HỆ THỐNG

#### 3.1. Thiết bị của hệ thống

Từ những phân tích chi tiết về nguyên lý hoạt động trong sơ đồ khối, chúng tôi tiến hành thiết kế hệ thống nhằm hiện thực hóa các chức năng đã đề ra. Các khối chức năng như khối nguồn, khối xử lý trung tâm, và khối chấp hành đều được tính toán kỹ để đảm bảo sự đồng bộ khi kết nối với nhau. Dựa trên các yêu cầu đó, các linh kiện phần cứng chính được lựa chọn không chỉ cần đáp ứng tốt về mặt thông số kỹ thuật mà còn phải có chi phí tiết kiệm.



Hình 2. Các thiết bị của hệ thống nhà thông minh

**Khối xử lý trung tâm:** Sử dụng vi máy tính **Raspberry Pi 3 Model B**. Đây là khối chức năng đóng vai trò cốt lõi, thực hiện nhiệm vụ thu thập dữ liệu từ các cảm biến đầu vào, thực thi thuật toán xử lý và xuất tín hiệu điều khiển tới các cơ cấu chấp hành. Bên cạnh đó, nhờ module Wi-Fi tích hợp sẵn, khối này còn đảm nhận việc truyền nhận dữ liệu hai chiều với máy chủ Blynk theo thời gian thực (Real-time).

**Khối cảm biến bao gồm:**

- ✓ Cảm biến nhiệt độ, độ ẩm DHT11: Dùng để đo lường các thông số môi trường trong phòng.
- ✓ Cảm biến khí gas MQ-2: Dùng để phát hiện nồng độ khí gas (LPG, CO...) nhằm đưa ra cảnh báo rò rỉ.

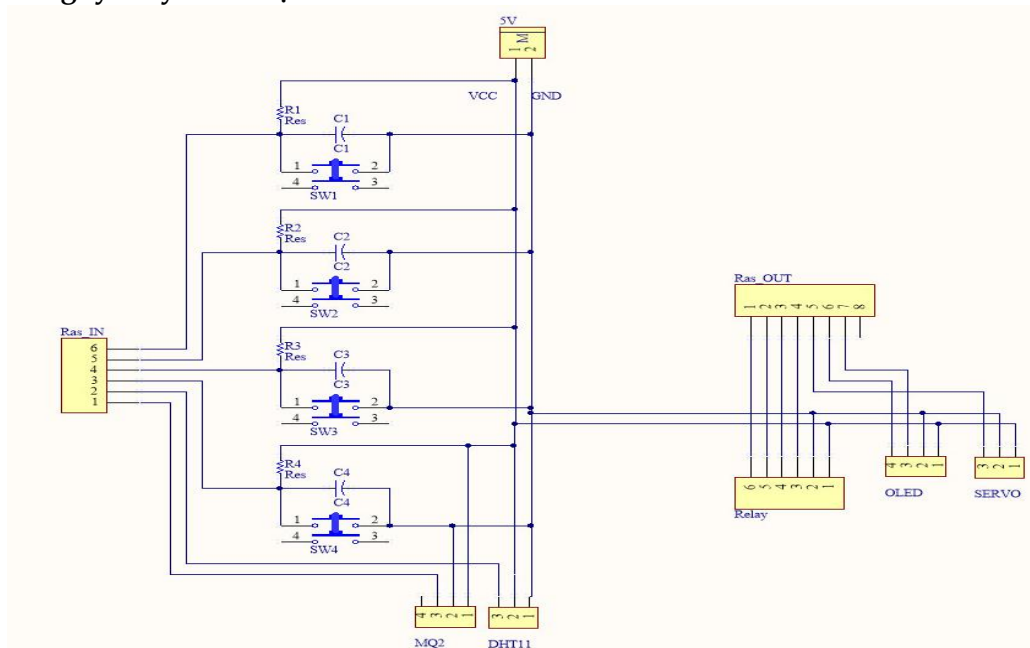
**Khối thực thi bao gồm:**

- ✓ Module Relay 4 kênh: Dùng làm công tắc điện tử, nhận lệnh từ Raspberry Pi để đóng/ngắt các thiết bị điện gia dụng như đèn và quạt.
- ✓ Động cơ Servo: Dùng để mô phỏng cơ cấu đóng/mở cửa tự động.

**Khối thông báo bao gồm:**

- ✓ Màn hình LCD 16x2 (kèm module I2C): Hiển thị tại chỗ các thông số nhiệt độ, độ ẩm và trạng thái hệ thống.
- ✓ Còi (Buzzer): Phát âm thanh cảnh báo khi cảm biến MQ-2 phát hiện nồng độ gas vượt ngưỡng an toàn.

**3.2. Sơ đồ nguyên lý toàn mạch**



Hình 3. Sơ đồ nguyên lý toàn mạch

Dựa trên sơ đồ khối của hệ thống được trình bày như ở Hình 1 và các thiết bị được lựa chọn như ở Hình 2, chúng tôi đã xây dựng sơ đồ nguyên lý như ở Hình 3. Sơ đồ mô tả mạch giao tiếp ngoại vi cho bộ xử lý trung tâm (ký hiệu là các port **Ras\_IN** và **Ras\_OUT** - tương ứng với các chân GPIO của Raspberry Pi). Hệ thống sử dụng nguồn cấp 5V.

#### **Khối Đầu vào (Kết nối với Ras\_IN):**

- ✓ Nút nhấn (SW1 - SW4): Gồm 4 nút nhấn được thiết kế theo cấu hình điện trở kéo lên (Pull-up Resistor) (R1-R4). Khi nhấn nút, tín hiệu gửi về Raspberry Pi sẽ chuyển từ mức cao (1) xuống mức thấp (0). Đặc biệt, mỗi nút nhấn đều đi kèm một tụ điện (C1-C4) mắc song song để chống rung phím (Hardware Debouncing), đảm bảo tín hiệu nhận được ổn định.
- ✓ Cảm biến: MQ2: Cảm biến khí gas, kết nối chân tín hiệu vào Ras\_IN; DHT11: Cảm biến nhiệt độ/độ ẩm, kết nối chân dữ liệu vào Ras\_IN.

#### **Khối Đầu ra (Kết nối với Ras\_OUT):**

- ✓ Relay: Cụm 6 chân điều khiển relay để đóng/ngắt các thiết bị điện áp cao (đèn, quạt...).
- ✓ OLED: Màn hình hiển thị, kết nối qua giao tiếp (thường là I2C) để hiển thị thông số.
- ✓ SERVO: Động cơ Servo dùng để điều khiển cơ cấu chấp hành (ví dụ: mở cửa, xoay van...).

### **3.3. Các khối chức năng của hệ thống**

Trong phần này trình bày chi tiết về hoạt động của các khối trong hệ thống với các thiết bị được lựa chọn bao gồm khối xử lý trung tâm, khối thông báo, khối cảm biến và khối thực thi.

#### **3.3.1. Khối xử lý trung tâm**

Khối xử lý trung tâm có nhiệm vụ thu thập dữ liệu từ cảm biến và nút nhấn, sau đó xử lý để hiển thị lên màn hình LCD và gửi về Webserver. Raspberry Pi 3 được lựa chọn làm bộ điều khiển chính (Hình 2) nhờ ưu điểm nhỏ gọn, khả năng giao tiếp ngoại vi linh hoạt và kết nối Internet ổn định, đáp ứng tốt các yêu cầu của hệ thống. Thông số kỹ thuật cụ thể như ở Bảng 1 [7].

**Bảng 1.** Thông số kỹ thuật của Raspberry Pi 3

<b>Thông số kỹ thuật</b>	<b>Nội dung</b>
Nguồn cung cấp	5V DC, dòng cấp tối thiểu 800mA
SoC	Broadcom BCM2836

CPU	ARM cortex – A7 lõi tứ, xung nhịp 900MHZ
RAM	1 Gb SDRAM (được chia sẻ giữa GPU và CPU)
Cổng USB	4 cổng USB 2.0
GPU	Broadcom VideoCore IV xung nhịp 250 MHz OpenGL ES 2.0 (tốc độ 24 GFLOPS) MPEG – 2 và VC – 1, 1080p30 H.264/MPEG – 4 AVC
Ngõ vào video	Cổng CSI (15 chân) sử dụng chung với module camera
Ngõ ra video	HDMI (rev 1.3 và 1.4)
Ngõ ra audio	Cổng audio 3.5mm hoặc cổng HDMI
Khe cắm thẻ nhớ	Hỗ trợ thẻ Micro SD
Kết nối mạng	Cổng Ethernet 10/100 Mbit/s, chuẩn RJ45
Ngõ ra khác	17 chân GPIO

### 3.3.2. Khối thông báo

Để đáp ứng yêu cầu về kích thước nhỏ gọn và tiết kiệm năng lượng cho mô hình nhà thông minh, chúng tôi sử dụng màn hình LCD 16x2 tích hợp module chuyển đổi I2C. Việc sử dụng giao tiếp I2C giúp mạch kết nối đơn giản (chỉ tốn 2 chân GPIO) nhưng vẫn đảm bảo đường truyền ổn định.

**Nguyên lý giao tiếp I2C:** Hoạt động dựa trên hai đường tín hiệu chính (cần treo trở kéo lên):

- ✓ **SDA (Serial Data):** Đường truyền dữ liệu hai chiều giữa thiết bị chủ (Master) và tớ (Slave).
- ✓ **SCL (Serial Clock):** Đường xung nhịp dùng để đồng bộ hóa quá trình truyền dữ liệu.

### 3.3.3. Khối cảm biến

#### a) Cảm biến nhiệt độ, độ ẩm DHT11

Module DHT11 được lựa chọn để đo thông số môi trường nhờ ưu điểm giá rẻ và tích hợp sẵn bộ chuyển đổi tín hiệu số đã qua hiệu chỉnh. Cảm biến hoạt động ở điện áp 3-5V, sử dụng giao tiếp 1-wire với khả năng đo nhiệt độ từ 0-50°C và độ ẩm từ 20-90% RH. Về kết nối, DHT11 sử dụng 3 chân đơn giản: 2 chân cấp nguồn và 1 chân truyền dữ liệu.

#### b) Cảm biến khí gas MQ-2

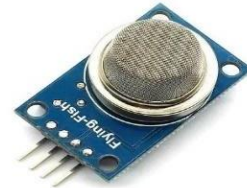
Cảm biến khí Gas MQ-2: Là loại cảm biến bán dẫn sử dụng vật liệu SnO<sub>2</sub>, có độ nhạy cao với các loại khí gây cháy và khói (LPG, Propane, Hydrogen, CO, Methane...). Khi phát hiện khí gas, độ dẫn điện của cảm biến thay đổi và được mạch xử lý chuyển đổi thành tín hiệu điện áp. Cảm biến MQ-2 có 4 chân (VCC, GND, DO, AO). Trong hệ thống này chúng tôi sử dụng chân Digital- DO cho phép xác định ngưỡng trực tiếp bằng phần cứng thông qua biến trở ở trên cảm biến. Khi có khí gas đèn sẽ sáng lên và chân DO sẽ chuyển từ mức cao (1) xuống mức thấp (0).



Hình 4. Màn hình LCD



Hình 5. Cảm biến DHT11



Hình 6. Cảm biến khí gas MQ-2

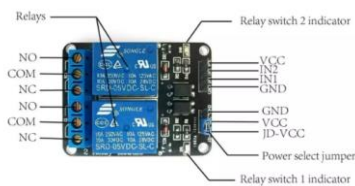
### 3.3.4. Khối thực thi

#### a) Module relay 2 kênh 5V

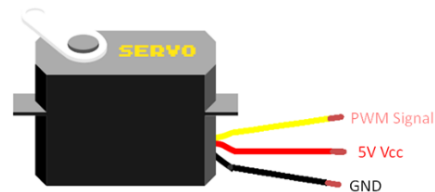
Module Relay 2 kênh 5V được sử dụng để điều khiển các thiết bị điện áp cao với dòng tải lên tới 10A. Module hỗ trợ tín hiệu kích 3.3V/5V (tích cực mức thấp) và sử dụng công nghệ Optocoupler để cách ly tín hiệu điều khiển với mạch động lực. Ngoài 4 chân tín hiệu cơ bản (VCC, GND, IN1, IN2), module còn tích hợp Jumper cho phép người dùng tùy chọn cấp nguồn riêng (RY\_VCC) cho Relay, giúp tách biệt hoàn toàn nguồn nuôi cuộn hút và nguồn tín hiệu để chống nhiễu hệ thống.

#### b) Động cơ servo SG90

Động cơ Servo SG90: Là dòng động cơ nhỏ gọn, phản hồi nhanh với góc quay 180 độ, chuyên dùng cho các mô hình tải trọng nhẹ. Động cơ tích hợp sẵn mạch Driver, cho phép điều khiển góc quay chính xác thông qua tín hiệu độ rộng xung (PWM). Do sử dụng hệ thống bánh răng nhựa, cần lưu ý giới hạn tải trọng để tránh làm hỏng cơ cấu.



Hình 7. Module Relay 2 Channel 5V



Hình 8. Động cơ Servo

## 4. XÂY DỰNG PHẦN MỀM ĐIỀU KHIỂN

Sau khi thực thi xong phần thiết kế và thi công phần cứng của hệ thống, tiến hành lập trình điều khiển cho Raspberry Pi 3 và sử dụng ứng dụng trên điện thoại để

kết nối với thiết bị. Trong hệ thống này sử dụng ứng dụng Blynk trên hệ điều hành iOS và Android để hiển thị, cảnh báo thông tin được gửi từ hệ thống.

#### 4.1. Phần mềm lập trình

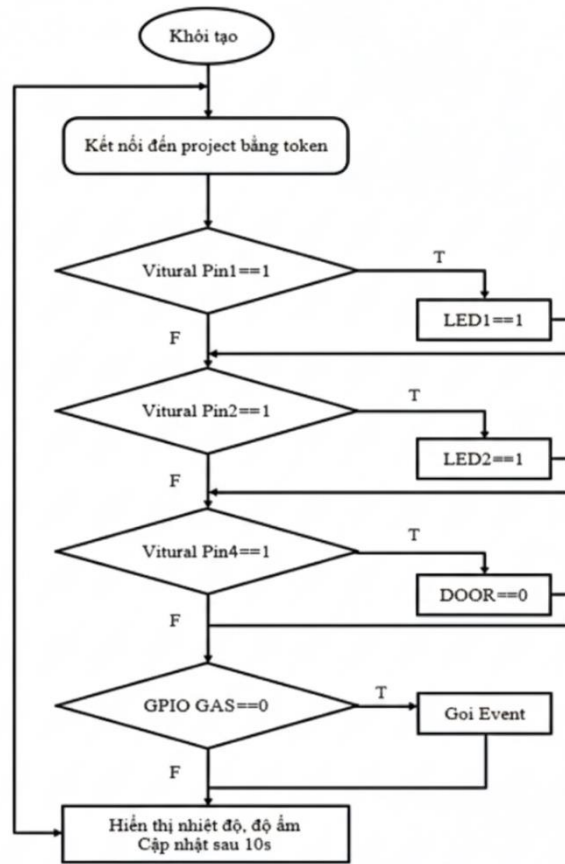
Chương trình điều khiển hệ thống được viết bằng ngôn ngữ Python. Môi trường phát triển tích hợp (IDE) Visual Studio Code (VSC) được sử dụng, kết hợp với giao thức Remote SSH để biên soạn, gỡ lỗi và nạp mã nguồn trực tiếp trên Raspberry Pi. Các công cụ này được lựa chọn dựa trên ưu điểm là mã nguồn mở, có cộng đồng hỗ trợ lớn, giúp đơn giản hóa và đẩy nhanh quá trình tiếp cận, nghiên cứu và triển khai hệ thống [8].

#### 4.2. Ứng dụng App Blynk

Blynk là một ứng dụng iOS và Android để kiểm soát thiết bị Esp8266, Arduino, Raspberry Pi và thiết bị khác thông qua Internet.

Lưu đồ thuật toán tổng thể của hệ thống được trình bày trong Hình 9. Lưu đồ giải thuật được giải thích như sau:

- ✓ **Bước 1: Khởi tạo:** Thiết lập thông số ban đầu và kiểm tra kết nối Internet/Server của Raspberry Pi.
- ✓ **Bước 2: Kiểm tra lệnh điều khiển (Virtual Pins):**
  - Virtual Pin 1 = 1: Kích hoạt Relay bật Đèn.
  - Virtual Pin 2 = 1: Kích hoạt LED báo hiệu.
  - Virtual Pin 3 = 1: Điều khiển Servo mở cửa.
  - Virtual Pin 4 = 1: Kích hoạt Quạt thông gió.
- ✓ **Bước 3: Cảnh báo an toàn:** Kiểm tra cảm biến Gas. Nếu chân tín hiệu ở mức thấp (0 - phát hiện khí gas), kích hoạt còi hú và hiển thị cảnh báo.
- ✓ **Bước 4: Cập nhật môi trường:** Đọc và hiển thị dữ liệu nhiệt độ, độ ẩm (định kỳ 10 giây/lần).
- ✓ **Bước 5:** Quay lại Bước 2 (Vòng lặp vô hạn).



Hình 9. Lưu đồ giải thuật

### 4.3. Xây dựng chương trình điều khiển cho Raspberry

Nhằm tăng tính cơ động, việc lập trình Python cho Raspberry Pi được thực hiện qua tiện ích **Remote – SSH** như ở Hình 10. Các bước kết nối bao gồm:

- ✓ **Bước 1:** Mở bảng lệnh bằng tổ hợp Ctrl + Shift + P và chọn tính năng **Add New SSH Host**.
- ✓ **Bước 2:** Nhập địa chỉ host theo cú pháp: ssh pi@<Địa\_chi\_IP> (VD: ssh pi@172.20.10.3).
- ✓ **Bước 3:** Nhập mật khẩu xác thực. Sau khi kết nối thành công, giao diện Visual Studio Code sẽ chuyển sang môi trường làm việc của Raspberry Pi.

```
Desktop > code > Raspberry Pi > ...
1 import sys
2 import RPi.GPIO as GPIO
3 import Adafruit_DHT
4 import time
5 import random
6 from gpiozero import Button
7 import drivers
8 from threading import Thread
9 led1 = LED(GPIO.get_mcp23017_pin(0), 0)
10 button_states, button_states2, button_states3, button_states4 = 0, 0, 0, 0
11
12 BLE_KEY_TOKEN = "b2c0c9f5e7b0b3c0b0e0c0c0c0c0c0c0"
13
14 display = drivers.LCD()
15
16 # Chèn nút nhấn GPIO
17 button_led1_pin = 1
18 button_led2_pin = 7
19 button_led3_pin = 8
20 button_led4_pin = 25
21 # pin cho led 1
22 led1 = 5
23 led2 = 6
24 led3 = 11
25 door = 4
26 fan = 10
27 dht11 = 15
28 gas = 14
```

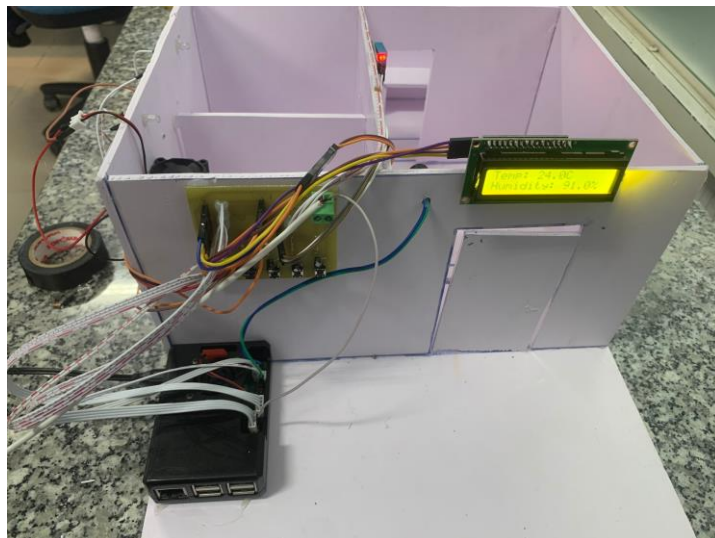
Hình 10. Giao diện lập trình cho Raspberry Pi

Qua phần trình bày trên, chúng tôi đã xây dựng hệ thống IoT thành công ứng dụng trên thiết bị và app Blynk trên điện thoại di động sử dụng Android và iOS.

## 5. KẾT QUẢ THỰC THI HỆ THỐNG

Hệ thống đã được thi công trên mô hình thực tế như Hình 11. Raspberry Pi 3, các module cảm biến và cơ cấu chấp hành được kết nối thông qua một mạch điện được thiết kế.

Để theo dõi nhiệt độ, độ ẩm trên app Blynk của điện thoại, đầu tiên chúng ta cần cài đặt ứng dụng Blynk thông qua App Store hoặc Google Play. Mở ứng dụng và tạo tài khoản mới bằng cách nhập email và mật khẩu. Sau đó giám sát thông số nhiệt độ, độ ẩm trên ứng dụng app Blynk, và nếu như có rò rỉ khí gas thì sẽ được cảnh báo trên loa và email như Hình 12, 13, 14.



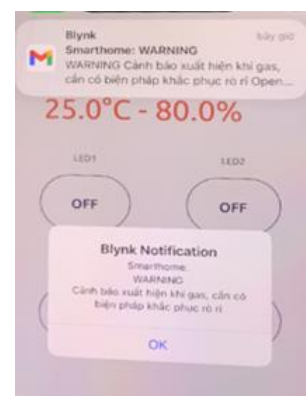
Hình 11. Mô hình nhà thông minh sau khi thực thi



Hình 12. Kết quả bật tắt các thiết bị trong nhà



Hình 13. Kết quả giám sát nhiệt độ, độ ẩm trên app Blynk



Hình 14. Kết quả giám sát khí gas trên app Blynk và cảnh báo trên mail.

Sau khi thi công, hệ thống được vận hành thử nghiệm và cho kết quả như trên Hình 12, 13, 14. Chi tiết kết quả được mô tả dưới đây.

- ✓ **Điều khiển (Hình 12):** Người dùng có thể điều khiển bật/tắt đèn (qua Relay) và đóng/mở cửa (qua Servo) thông qua các nút nhấn trên ứng dụng Blynk. Hệ thống phản hồi gần như tức thời.
- ✓ **Giám sát (Hình 13):** Các thông số nhiệt độ, độ ẩm từ DHT11 và trạng thái khí gas từ MQ-2 được đọc và hiển thị đồng bộ lên màn hình LCD và giao diện ứng dụng Blynk theo thời gian thực.
- ✓ **Cảnh báo (Hình 14):** Khi nồng độ khí gas vượt ngưỡng (mô phỏng bằng cách xịt gas vào cảm biến MQ-2), hệ thống lập tức kích hoạt còi báo động tại chỗ và gửi cảnh báo (notification) đến điện thoại người dùng qua Blynk.

Với các kết quả nêu trên, chúng tôi đã chứng minh được việc hoàn thiện xây dựng hệ thống nhà thông minh và kết quả hệ thống có thể vận hành được. Bên cạnh đó, hệ thống sử dụng bộ Raspberry Pi 3 với 40 chân, trong đó có chỉ mới sử dụng 13 chân trong tổng số 26 chân vào ra. Vì vậy, hệ thống có thể kết nối thêm các cảm biến, thiết bị ngoại vi khác.

## 6. KẾT LUẬN

Nghiên cứu đã xây dựng thành công một giải pháp nhà thông minh toàn diện dựa trên sức mạnh xử lý của Raspberry Pi 3 và sự linh hoạt của nền tảng Blynk. Hệ thống không chỉ dừng lại ở việc điều khiển thiết bị từ xa mà còn hoạt động như một trợ lý giám sát an toàn, cung cấp thông tin môi trường và cảnh báo rò rỉ khí gas kịp thời. Qua thử nghiệm thực tế, hệ thống vận hành ổn định, chính xác, đáp ứng tốt các yêu cầu về tốc độ

phản hồi và giao diện người dùng. Đây là minh chứng cho thấy việc ứng dụng công nghệ IoT vào đời sống là hoàn toàn khả thi với chi phí thấp và hiệu quả cao.

### TÀI LIỆU THAM KHẢO

- [1] S. K. & C. N. P. Jana, "Smart home automation system using Raspberry P," in *International Conference on Computing Methodologies and Communication (ICCMC)*, 2017.
- [2] R. e. a. Sowah, "Design of a Secure Wireless Smart Home System with Raspberry Pi and Arduino," *Journal of Sensors*, 2020.
- [3] e. a. G. J. Han, "Smart Home System Based on Raspberry Pi and IoT," *International Journal of Smart Home*, 2018.
- [4] C. V. T. & N. V. Nghĩa, "Điều khiển thiết bị qua internet dùng Raspberry Pi 2 thông qua wifi," Trường Đại học Sư phạm Kỹ thuật TP.HCM, 2016.
- [5] N. Đ. Luyện, "Xây dựng hệ thống nhà thông minh dựa trên nền tảng Raspberry Pi và trợ lý ảo tiếng Việt," Đại học Bách Khoa Hà Nội, 2019.
- [6] L. H. Nam, "Ứng dụng giao thức MQTT và Raspberry Pi trong điều khiển giám sát thiết bị điện gia đình," *Tạp chí Khoa học và Công nghệ - Đại học Thái Nguyên*, 2021.
- [7] M. R. & S. Wallace, "Getting Started with Raspberry Pi," O'Reilly Media, ISBN: 978-1449344214, 2012.
- [8] DusunIoT, "Smart Home Automation Projects Using Raspberry Pi".

## RESEARCH ON THE DESIGN AND CONSTRUCTION OF A SMART HOME MODEL USING RASPBERRY PI 3

Hoang Dai Long\*, Truong Quoc Khanh,  
Nguyen Hoang Thanh Tung, Nguyen Van An

Faculty of Electronics, Electrical Engineering and Material Technology, University of  
Sciences, Hue University

\*Email: longhoang@husc.edu.vn

### ABSTRACT

This study presents the design and implementation of a smart home model based on the Internet of Things (IoT) platform. The system employs the Raspberry Pi 3 as the central processing unit, interfaced with sensor modules (DHT11, MQ-2) and actuators (Relay, Servo) to monitor the environmental conditions and control household devices. The Blynk platform is utilized to develop a mobile application that enables users to remotely control appliances such as lights, fans, and doors, while simultaneously receiving real-time environmental data. Furthermore, the system successfully integrates a gas leak warning function through both the mobile application and an alarm buzzer. Experimental results indicate that the model operates reliably with a rapid response time, providing an efficient and cost-effective solution for home automation.

**Keywords:** Smarthouse, IoT, Raspberry P.