

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



BÁO CÁO ĐỒ ÁN MÔN HỌC
“TÌM HIỂU VÀ XÂY DỰNG ỨNG DỤNG TRÊN
ARDUINO/ESP8266”

Môn học: Hệ thống nhúng mạng không dây

GVHD: Lê Anh Tuấn

Lớp: NT131.012

Nhóm: 7

STT	Họ và tên	MSSV
1	Võ Thùy Duyên	21520023
2	Cao Thị Kim Cẩm	21520641
3	Huỳnh Ngọc Bảo Châu	21520643
4	Lê Minh Phúc	21521285

MỤC LỤC

MỨC ĐỘ ĐÓNG GÓP CỦA CÁC THÀNH VIÊN.....	3
Chương 1. TỔNG QUAN VỀ ĐỒ ÁN.	3
1.1. Đặt vấn đề.	3
1.2. Mục tiêu đồ án.	3
1.3. Nội dung đồ án.	3
1.4. Phạm vi ứng dụng.....	4
Chương 2. CƠ SỞ LÝ THUYẾT.....	4
2.1. Giới thiệu về internet of things.	4
2.1.1. Giới thiệu về IoT.....	4
2.1.2. Lịch sử hình thành.	5
2.1.3. Ứng dụng của IoT.	5
2.2. Công nghệ Wifi.	6
2.2.1. Giới thiệu.	6
2.2.2. Công nghệ truyền nhận dữ liệu.	7
2.2.3. Giới thiệu về ESP8266 NodeMCU.....	8
2.2.4. Module DHT11.....	10
2.2.5. Các chuẩn giao tiếp được sử dụng.	11
2.2.6. Phần mềm sử dụng.....	12
Chương 3. PHÂN TÍCH HỆ THỐNG VÀ THIẾT KẾ.	12
3.1. Cách thức hoạt động của hệ thống.....	12
3.2. Mô tả hoạt động của hệ thống.	13
3.3. Thiết kế sơ đồ hệ thống.	13
Chương 4. THI CÔNG HỆ THỐNG.	15
4.1. Thực hiện lắp ráp và ghép nối các mạch và module.....	15
4.2. Lập trình hệ thống.....	16
Chương 5. KẾT QUẢ, NHẬN XÉT, ĐÁNH GIÁ.....	25
5.1. Kết quả đạt được.	26
5.2. Kết quả thực nghiệm.....	26
5.3. Nhận xét và đánh giá.	27
Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	28
6.1. Ưu điểm và nhược điểm.....	28
6.2. Hướng phát triển.	28

MỨC ĐỘ ĐÓNG GÓP CỦA CÁC THÀNH VIÊN.

Họ và tên	MSSV	Mức độ đóng góp
Võ Thuỳ Duyên	21520023	code web (100%), báo cáo fileword (10%), slide (100%)
Cao Thị Kim Cẩm	21520641	lắp ráp mạch (100%), code arduino (50%), báo cáo file word (20%)
Huỳnh Ngọc Bảo Châu	21520643	code app (100%), báo cáo file word (50%)
Lê Minh Phúc	21521285	code arduino (50%), Blynk app (100%), báo cáo file word (20%)

Chương 1. TỔNG QUAN VỀ ĐỒ ÁN.

1.1. Đặt vấn đề.

Ngày nay, Việt Nam đứng thứ hai Đông Nam Á về kim ngạch xuất khẩu hàng nông, lâm, thủy sản, kèm theo đó là sản lượng ngày một nhiều, việc bảo quản sản phẩm được lâu ngày mà vẫn giữ được chất lượng cũng là một vấn đề đáng quan tâm. Giải pháp phổ biến nhất có thể kể đến là bảo quản trong kho lạnh, cụ thể ở đây là việc bảo quản trái cây, yêu cầu nhiệt độ khoảng 0 đến 15 độ C, độ ẩm 85 đến 95%. Tuy nhiên, cho dù được đầu tư lắp đặt các hệ thống máy làm lạnh, nhưng vẫn không thể tránh được trường hợp các loại máy này bị hư hỏng do phải làm việc liên tục trong thời gian dài, và tệ nhất là nó hỏng đột ngột, người dùng không thể phát hiện kịp thời. Để có thể chủ động giám sát và ngăn chặn các mối nguy hại liên quan đến nhiệt độ, độ ẩm có thể xảy ra trong kho lạnh bảo quản trái cây, nhóm chúng em quyết định triển khai một đồ án về “Giám sát nhiệt độ, độ ẩm và điều khiển thiết bị từ xa”. Từ đồ án này, chúng em hi vọng rằng thiết bị sẽ được áp dụng rộng rãi và giúp cho các doanh nghiệp ngày càng phát triển.

1.2. Mục tiêu đồ án.

- Chức năng giám sát, theo dõi nhiệt độ, độ ẩm trên ứng dụng, web.
- Nhận thông báo nếu trong kho hàng gặp vấn đề về nhiệt độ, độ ẩm.
- Điều khiển thiết bị điện (đèn) từ xa.
- Thi công đồ án trên mô hình.

1.3. Nội dung đồ án.

Để xây dựng đồ án “Giám sát nhiệt độ, độ ẩm và điều khiển thiết bị từ xa dùng Arduino/ESP8266” cần phải thực hiện các nội dung sau:

- Tìm hiểu về Internet of Things.
- Tìm hiểu về công nghệ Wifi: giới thiệu về ESP8266, module DHT11.
- Phân tích và thiết kế hệ thống.
- Thi công hệ thống phần cứng, thử nghiệm, hiệu chỉnh.
- Thử nghiệm và điều chỉnh hệ thống và chương trình để hệ thống được tối ưu.
- Nhận xét, đánh giá.

1.4. Phạm vi ứng dụng.

- Đề tài tập trung vào việc ứng dụng giám sát nhiệt độ, độ ẩm trong việc bảo quản trái cây trong kho lạnh. Ngoài ra có thể phát triển ứng dụng trong các kho bảo quản thực phẩm, trong y tế...

Chương 2. CƠ SỞ LÝ THUYẾT.

2.1. Giới thiệu về internet of things.

2.1.1. Giới thiệu về IoT.

- IoT là viết tắt của Internet of Things, có thể dịch cụ thể là “Mạng lưới vạn vật kết nối internet”, dùng để chỉ các thiết bị được kết nối internet có khả năng thu thập dữ liệu, chia sẻ thông tin với độ bao phủ toàn cầu, nghĩa là bất cứ thiết bị nào có khả năng kết nối internet, thu thập, lưu giữ và chia sẻ thông tin thì đều là IoT. Có thể thấy xung quanh chúng ta có rất nhiều sự xuất hiện của các thiết bị IoT: máy tính, điện thoại di động, ô tô cảm biến nhiệt, các thiết bị gia dụng cảm ứng nhiệt, các hệ thống tự động hóa... Sự xuất hiện của các thiết bị IoT giúp bổ sung một mức độ thông minh kỹ thuật số tới các thiết bị tự động khác, cho phép chúng tự động thu thập, trao đổi thông tin tự động mà không cần sự can thiệp của con người, giúp tối ưu hóa giữa hai thế giới vật lý và kỹ thuật số.

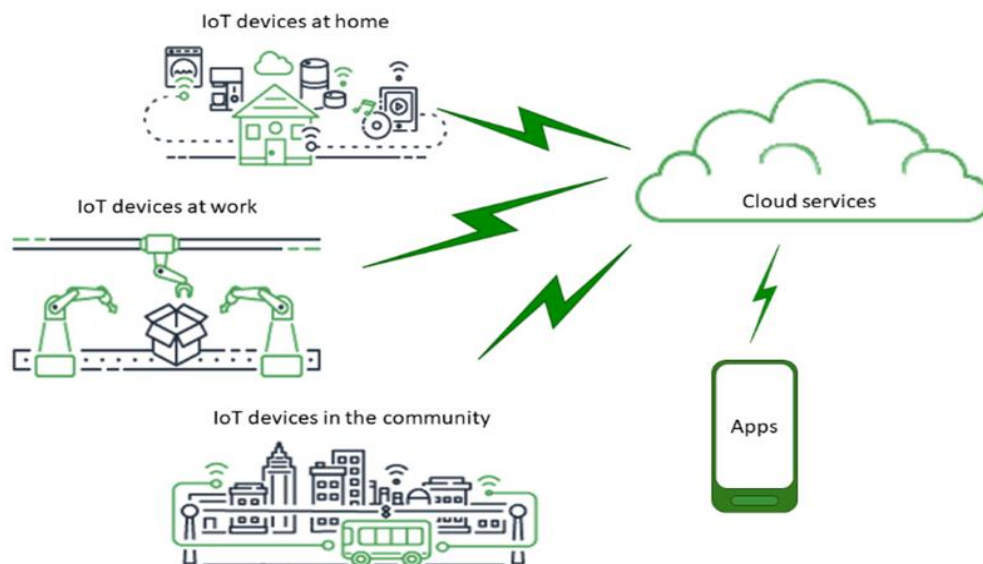
- Một hệ thống IoT gồm ba thành phần:

+ Thiết bị thông minh: thiết bị này thu thập dữ liệu từ môi trường xung quanh, thao tác nhập liệu của người dùng hoặc mô hình sử dụng và truyền/nhận dữ liệu qua Internet từ ứng dụng IoT của nó.

+ Ứng dụng IoT: là một tập hợp các dịch vụ và phần mềm có chức năng tích hợp dữ liệu nhận được từ các thiết bị IoT khác nhau. Ứng dụng này sử dụng AI để phân tích dữ liệu và đưa ra các quyết định sáng suốt. Những quyết định này được

truyền trở lại thiết bị IoT và sau đó, thiết bị IoT đó sẽ phản hồi lại dữ liệu đầu vào một cách thông minh.

+ Giao diện đồ họa người dùng: một hoặc một nhóm các thiết bị IoT có thể được quản lý thông qua giao diện đồ họa người dùng. Được sử dụng phổ biến nhất là ứng dụng di động hoặc trang web có thể được sử dụng để đăng ký và kiểm soát các thiết bị thông minh.



2.1.2. Lịch sử hình thành.

- Năm 1982, đã có những ý tưởng thảo luận về việc thêm cảm biến và trí thông minh vào các vật bình thường được đưa ra.
- Năm 1999, tại buổi thuyết trình của công ty Procter & Gamble, nhà khoa học Kevin Ashton là người đầu tiên đề cập đến thuật ngữ Internet of Things (IoT).
- Giai đoạn từ năm 2000 đến 2013, IoT được nghiên cứu và cho vào sử dụng rộng rãi trong các lĩnh vực khác nhau như thiết bị chăm sóc sức khỏe hay đồ gia dụng.
- Đến năm 2014, số lượng các thiết bị máy móc và di động được kết nối với mạng Internet đã vượt qua cả dân số của thế giới lúc bấy giờ.
- Năm 2015, một số loại mô hình trang trại IoT, robot IoT đã được công bố và đưa vào ứng dụng cũng như được phát triển cho đến ngày nay.

2.1.3. Ứng dụng của IoT.

- Ô tô thông minh: những phương tiện như ô tô có thể kết nối internet thông qua camera hành trình hoặc qua cổng kết nối của phương tiện. Chúng thu thập dữ

liệu từ chân ga, phanh, đồng hồ đo tốc độ, đồng hồ đo quãng đường, bánh xe và bình xăng để giám sát cả hiệu suất của người lái và tình trạng phương tiện.

- Nhà thông minh: các thiết bị gia đình thông minh tập trung chủ yếu vào hoạt động cải thiện hiệu quả và độ an toàn của ngôi nhà. Các thiết bị như ổ điện thông minh có thể giám sát mức sử dụng điện và bộ điều hòa thông minh có thể cung cấp khả năng kiểm soát nhiệt độ tốt hơn. Các hệ thống thủy canh có thể sử dụng cảm biến IoT để quản lý khu vườn, trong khi đó, máy báo khói IoT có thể phát hiện khói thuốc lá. Các hệ thống an ninh gia đình như khóa cửa, camera an ninh và máy phát hiện rò nước có thể phát hiện và ngăn chặn các mối nguy hiểm, đồng thời gửi cảnh báo tới chủ nhà.

- Thành phố thông minh: các ứng dụng IoT đã giúp quá trình quy hoạch đô thị và bảo trì cơ sở hạ tầng hiệu quả hơn. Chính phủ cũng đang sử dụng ứng dụng IoT để giải quyết những vấn đề về cơ sở hạ tầng, y tế và môi trường.

- Công trình thông minh: các công trình như khuôn viên trường đại học và công trình thương mại sử dụng ứng dụng IoT để thúc đẩy hoạt động hiệu quả hơn. Những công trình thông minh có thể sử dụng các thiết bị IoT cho mục đích: giảm mức tiêu thụ năng lượng, giảm chi phí bảo trì, tận dụng không gian làm việc hiệu quả hơn.

2.2. Công nghệ Wifi.

2.2.1. Giới thiệu.

- Wi-Fi là viết tắt của Wireless Fidelity, là một công nghệ mạng không dây cho phép các thiết bị điện tử kết nối với nhau và truy cập Internet mà không cần cáp. Wi-Fi sử dụng sóng vô tuyến để truyền dữ liệu, tương tự như sóng điện thoại, truyền hình và radio.

- Nguyên lý hoạt động của Wifi: Wi-Fi hoạt động dựa trên nguyên lý giao tiếp điểm-điểm (point-to-point). Các thiết bị Wi-Fi được kết nối với nhau thông qua một điểm phát sóng (access point). Điểm phát sóng sẽ phát sóng tín hiệu Wi-Fi ra môi trường xung quanh. Các thiết bị Wi-Fi khác có thể nhận được tín hiệu này và kết nối với điểm phát sóng để truy cập Internet hoặc giao tiếp với các thiết bị khác.

- Một hệ thống Wi-Fi bao gồm các thành phần sau:

- + Thiết bị Wi-Fi: Là các thiết bị có khả năng kết nối Wi-Fi, chẳng hạn như máy tính, điện thoại, máy tính bảng, v.v.

- + Điểm phát sóng: Là thiết bị phát sóng tín hiệu Wi-Fi ra môi trường xung quanh.

- + Modem: Là thiết bị kết nối mạng Wi-Fi với Internet.

- Ưu điểm của Wi-Fi:

+ Tiện lợi: Wi-Fi cho phép người dùng kết nối với mạng mà không cần phải sử dụng cáp.

+ Linh hoạt: Wi-Fi có thể được sử dụng ở nhiều vị trí khác nhau, chẳng hạn như trong nhà, ngoài trời, v.v.

+ Chi phí thấp: Wi-Fi có chi phí triển khai và vận hành thấp hơn so với các phương thức kết nối mạng có dây.

- Nhược điểm của Wi-Fi:

+ Tốc độ truy cập: Tốc độ truy cập Wi-Fi phụ thuộc vào nhiều yếu tố, chẳng hạn như chuẩn Wi-Fi, khoảng cách giữa thiết bị và điểm phát sóng, v.v.

+ Bảo mật: Wi-Fi có thể bị tấn công bởi các hacker nếu không được bảo mật đúng cách.

- Wi-Fi được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

+ Truyền thông: Wi-Fi được sử dụng để kết nối các thiết bị điện tử với nhau, chẳng hạn như máy tính, điện thoại, máy tính bảng, v.v.

+ Truy cập Internet: Wi-Fi được sử dụng để truy cập Internet từ các thiết bị di động.

+ Giải trí: Wi-Fi được sử dụng để phát trực tuyến video, âm nhạc, v.v.

+ Doanh nghiệp: Wi-Fi được sử dụng trong các doanh nghiệp để kết nối các thiết bị mạng, chẳng hạn như máy in, máy quét, v.v.

2.2.2. Công nghệ truyền nhận dữ liệu.

- Wi-Fi được phát triển dựa trên các chuẩn của họ IEEE 802.11. Hiện nay, có 9 chuẩn Wi-Fi đã được phát triển, bao gồm:

+ 802.11a: Phát hành năm 1999, sử dụng băng tần 5 GHz, tốc độ tối đa 54 Mbps.

+ 802.11b: Phát hành năm 1999, sử dụng băng tần 2,4 GHz, tốc độ tối đa 11 Mbps.

+ 802.11g: Phát hành năm 2003, sử dụng băng tần 2,4 GHz, tốc độ tối đa 54 Mbps.

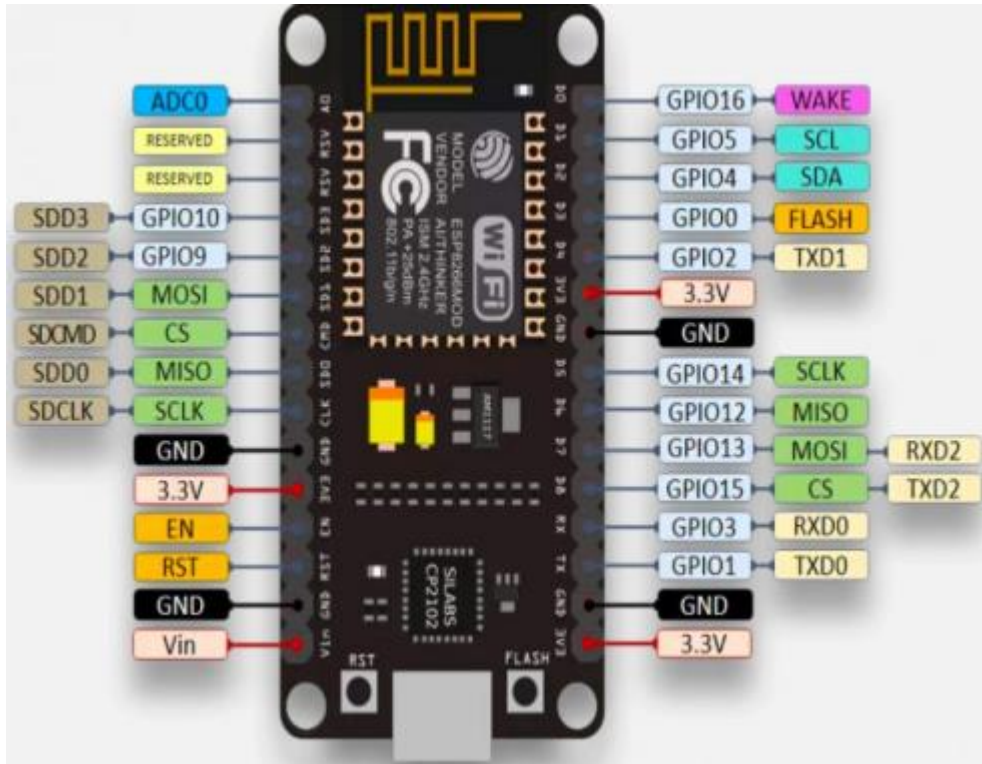
+ 802.11n: Phát hành năm 2009, sử dụng cả băng tần 2,4 GHz và 5 GHz, tốc độ tối đa 600 Mbps.

+ 802.11ac: Phát hành năm 2014, sử dụng băng tần 5 GHz, tốc độ tối đa 6,93 Gbps.

+ 802.11ax: Phát hành năm 2019, sử dụng cả băng tần 2,4 GHz và 5 GHz, tốc độ tối đa 9,6 Gbps.

2.2.3. Giới thiệu về ESP8266 NodeMCU.

- Cấu tạo của NODEMCU ESP8266:



- Thông số kỹ thuật:

- + IC chính: ESP8266 Wifi SoC.
- + Phiên bản firmware: NodeMCU Lua.
- + Điện áp giao tiếp: 3.3 VDC.
- + Chuẩn WiFi: 802.11 b/g/n.
- + WiFi 2.4GHz, hỗ trợ chế độ bảo mật WPA/WPA2.
- + Hỗ trợ 3 chế độ hoạt động STA/AP/STA+AP.
- + Chip nạp và giao tiếp UART: CH340.
- + Chân D0~D8, SD1~SD3: sử dụng như GPIO, PWMM, IIC... dòng điều khiển 15mA.
- + GPIO giao tiếp mức 3.3VDC.
- + A0 (AD0): 1 kênh ADC.
- + Nguồn vào: 5VDC MicroUSB hoặc Vin.

- + Flash: 4MB.
- Chức năng của các chân như sau:
 - + VCC: 3.3V.
 - + GND: chân nối đất.
 - + Tx: chân Tx của giao thức UART, kết nối đến chân Rx của vi điều khiển.
 - + Rx: chân Rx của giao thức UART, kết nối đến chân Tx của vi điều khiển.
 - + RST: chân reset, kéo xuống mass để reset.
- ESP8266 NodeMCU là một board mạch phát triển dựa trên chip ESP8266 của Espressif Systems. Nó bao gồm các thành phần chính và tính năng như sau sau:
 - + Chip ESP8266: Chip ESP8266 là một chip hệ thống trên chip (SoC) tích hợp bộ xử lý lõi đơn 32-bit, bộ nhớ RAM 512 KB, bộ nhớ flash 4 MB và bộ điều khiển Wi-Fi tích hợp.
 - + Bộ nhớ flash: Bộ nhớ flash 4MB cho phép lưu trữ firmware và dữ liệu ứng dụng.
 - + Cổng GPIO: ESP8266 NodeMCU có 12 cổng GPIO có thể được sử dụng để kết nối với các cảm biến, thiết bị ngoại vi và các thành phần khác.
 - + UART: ESP8266 NodeMCU có cổng UART có thể được sử dụng để kết nối với máy tính hoặc các thiết bị khác.
 - + SPI: ESP8266 NodeMCU có cổng SPI có thể được sử dụng để kết nối với các thiết bị ngoại vi như bộ nhớ flash hoặc màn hình LCD.
 - + I2C: ESP8266 NodeMCU có cổng I2C có thể được sử dụng để kết nối với các thiết bị ngoại vi như cảm biến hoặc màn hình LCD.
- Chân GPIO của ESP8266 NodeMCU:
 - + ESP8266 NodeMCU có 12 cổng GPIO, được đánh số từ GPIO0 đến GPIO11. Các chân GPIO này có thể được sử dụng để kết nối với các cảm biến, thiết bị ngoại vi và các thành phần khác.
 - + Mỗi chân GPIO có thể được cấu hình bên trong ở mức HIGH hoặc LOW. Các chân GPIO cũng có thể được sử dụng để xuất hoặc nhập dữ liệu.
- Chân UART của ESP8266 NodeMCU:
 - + ESP8266 NodeMCU có cổng UART, có thể được sử dụng để kết nối với máy tính hoặc các thiết bị khác.
 - + Cổng UART sử dụng giao thức truyền thông serial để truyền dữ liệu giữa các thiết bị.

- Chân SPI của ESP8266 NodeMCU:
 - + ESP8266 NodeMCU có cổng SPI, có thể được sử dụng để kết nối với các thiết bị ngoại vi như bộ nhớ flash hoặc màn hình LCD.
 - + Cổng SPI sử dụng giao thức truyền thông serial để truyền dữ liệu giữa các thiết bị.
- Chân I2C của ESP8266 NodeMCU:
 - + ESP8266 NodeMCU có cổng I2C, có thể được sử dụng để kết nối với các thiết bị ngoại vi như cảm biến hoặc màn hình LCD.
 - + Cổng I2C sử dụng giao thức truyền thông serial để truyền dữ liệu giữa các thiết bị.

2.2.4. Module DHT11.

- Module DHT11 là một cảm biến nhiệt độ và độ ẩm kỹ thuật số sử dụng giao tiếp 1 dây. Nó có độ chính xác cao, độ ổn định lâu dài và thời gian phản hồi nhanh. Module DHT11 rất dễ sử dụng và được ứng dụng rộng rãi trong các hệ thống tự động hóa, điều khiển khí hậu và dự báo thời tiết.
- Thông số kỹ thuật của module DHT11:
 - + Điện áp hoạt động: 3.3 - 5.5 VDC
 - + Dòng điện tiêu thụ: 1.5 mA
 - + Độ chính xác nhiệt độ: $\pm 2^{\circ}\text{C}$
 - + Độ chính xác độ ẩm: $\pm 5\%\text{RH}$
 - + Khoảng đo nhiệt độ: 0 - 50°C
 - + Khoảng đo độ ẩm: 20 - $80\%\text{RH}$
 - + Thời gian phản hồi: <2 giây
 - + Giao tiếp: 1 dây
- Ứng dụng của module DHT11:
 - + Hệ thống tự động hóa nhà.
 - + Hệ thống điều khiển khí hậu.
 - + Hệ thống dự báo thời tiết.
 - + Hệ thống tưới tiêu tự động.
 - + Hệ thống giám sát môi trường.
 - + Hệ thống nông nghiệp thông minh.

- Để kết nối module DHT11 với ESP8266 NodeMCU, cần sử dụng các chân GPIO sau:

+ Chân DATA của module DHT11 kết nối với chân GPIO0 (D3) của ESP8266 NodeMCU.

+ Chân VCC của module DHT11 kết nối với 3.3V nguồn dương hoặc của ESP8266 NodeMCU.

+ Chân GND của module DHT11 kết nối với chân GND.

- Sau khi kết nối, ta có thể sử dụng các thư viện có sẵn để đọc dữ liệu nhiệt độ và độ ẩm từ module DHT11.

2.2.5. Các chuẩn giao tiếp được sử dụng.

- Chuẩn giao tiếp One Wire được nghiên cứu và phát triển bởi Dallas Semiconductor (Maxim). One Wire chỉ sử dụng một dây đường truyền để thực hiện truyền và nhận dữ liệu. Chuẩn giao tiếp One Wire là chuẩn giao tiếp không đồng bộ bán song công (half-duplex: tại một thời điểm, tín hiệu chỉ có thể chạy theo một hướng).

- Vì One Wire chỉ sử dụng một dây để nối nguồn và truyền dữ liệu, nên khi ở trạng thái rảnh (không có dữ liệu trên đường truyền) thì nó cần phải ở mức cao, do đó cần kết nối dây này với nguồn thông qua một điện trở. Điện trở này thường được gọi là điện trở kéo lên (pull-up resistor). Tùy theo các thiết bị mà giá trị điện trở này có thể thay đổi. Cảm biến DHT11 sử dụng giao tiếp One Wire.

- Chuẩn giao tiếp UART: các UART sẽ truyền và nhận dữ liệu từ một bus dữ liệu (Data Bus), giao thức truyền không đồng bộ. Ngoài dữ liệu ra trong 1 lần truyền còn nhét thêm các Start bit, Stop bit, Parity bit. Các bit thêm này vào giúp Slave nhận biết, kiểm tra và nhận được đúng tín hiệu.

- Ưu điểm:

+ Chỉ sử dụng hai dây.

+ Không cần tín hiệu đồng hồ.

+ Có một bit chẵn lẻ để cho phép kiểm tra lỗi.

+ Cấu trúc của gói dữ liệu có thể được thay đổi.

+ Phương pháp được ghi chép rõ ràng và được sử dụng rộng rãi.

- Nhược điểm:

+ Kích thước của khung dữ liệu được giới hạn tối đa là 9 bit.

+ Không hỗ trợ nhiều hệ thống phụ hoặc nhiều hệ thống chính.

+ Tốc độ truyền của UART phải nằm trong khoảng 10% của nhau.

2.2.6. Phần mềm sử dụng.

- Phần mềm lập trình Arduino IDE là phần mềm mã nguồn mở chủ yếu được sử dụng để viết và biên dịch mã vào module Arduino. Đây là một phần mềm Arduino chính thức, giúp cho việc biên dịch mã trở nên dễ dàng mà ngay cả một người bình thường không có kiến thức kỹ thuật cũng có thể làm được. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300.000 board mạch được thiết kế sẵn với các cảm biến, linh kiện. Phần mềm giúp bạn có thể sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.

- Blynk là một nền tảng với các ứng dụng iOS và Android để điều khiển Arduino, Raspberry Pi và các ứng dụng tương tự qua Internet.

+ Nó là một bảng điều khiển kỹ thuật số nhờ đó có thể xây dựng giao diện đồ họa cho dự án của mình bằng cách kéo và thả các widget.

+ Việc thiết lập mọi thứ rất đơn giản và ta sẽ bắt đầu sau chưa đầy 5 phút.

+ Blynk không bị ràng buộc với một số bo hoặc shield cụ thể. Thay vào đó, nó hỗ trợ phần cứng mà ta lựa chọn. Cho dù Arduino hoặc Raspberry Pi được liên kết với Internet qua Wi-Fi, Ethernet hoặc chip ESP8266, Blynk sẽ giúp chúng ta online và sẵn sàng cho IoT.

+ Có ba thành phần chính trong nền tảng:

1. Ứng dụng Blynk - cho phép tạo giao diện cho các dự án của mình bằng cách sử dụng các widget khác nhau.

2. Blynk Server - chịu trách nhiệm về tất cả các giao tiếp giữa điện thoại thông minh và phần cứng. Có thể sử dụng Blynk Cloud hoặc chạy cục bộ máy chủ Blynk riêng của mình. Nó là mã nguồn mở, có thể dễ dàng xử lý hàng nghìn thiết bị và thậm chí có thể được khởi chạy trên Raspberry Pi.

3. Thư viện Blynk - dành cho tất cả các nền tảng phần cứng phổ biến - cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và lệnh đi.

+ Mỗi khi nhấn một nút trong ứng dụng Blynk, thông điệp sẽ truyền đến không gian của đám mây Blynk, và tìm đường đến phần cứng của chúng ta.

Chương 3. PHÂN TÍCH HỆ THỐNG VÀ THIẾT KẾ.

3.1. Cách thức hoạt động của hệ thống.

- Các thành phần quan trọng:

+ Bộ xử lý trung tâm.

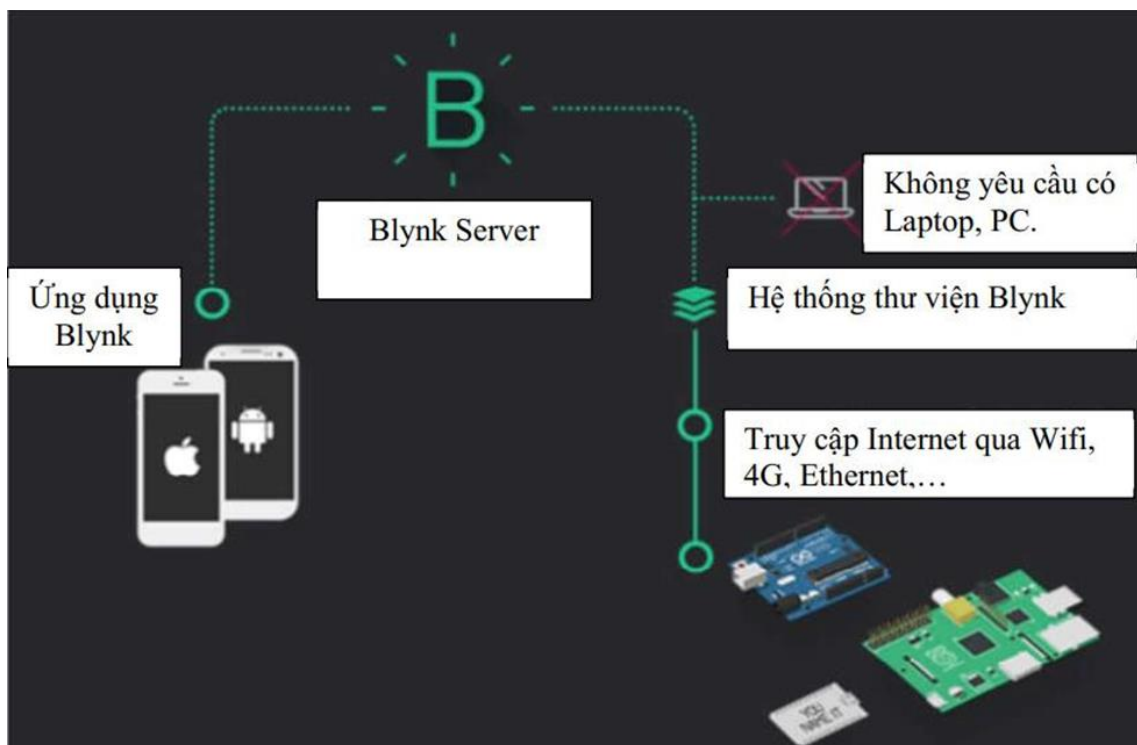
+ Ứng dụng Blynk.

Đề tài yêu cầu thiết kế một hệ thống giám sát nhiệt độ, độ ẩm và điều khiển thiết bị từ xa. Hệ thống được giám sát bằng ứng dụng Blynk đồng thời có thể điều khiển từ xa thông qua nút điều khiển trên app Blynk.

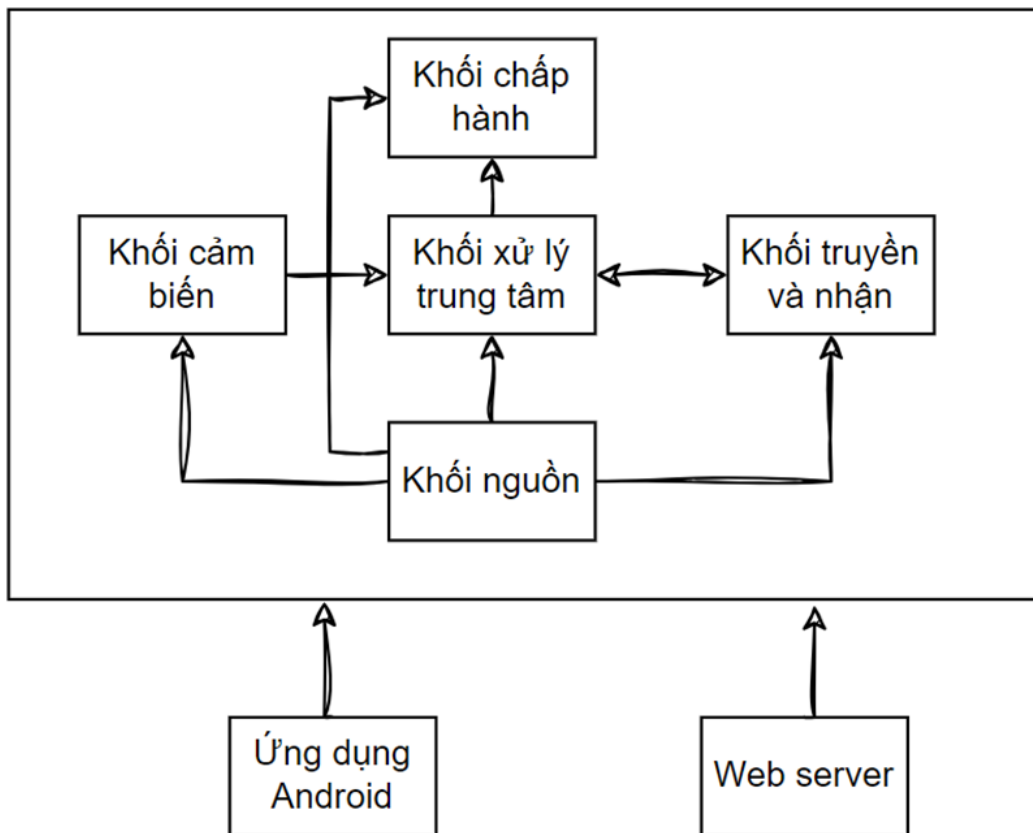
3.2. Mô tả hoạt động của hệ thống.

- Bộ xử lý trung tâm:
 - + Điện áp vào khoảng 5 vdc.
 - + Thu thập các chỉ số từ thiết bị cảm biến.
 - + Giao tiếp với Web server bằng Wifi (truyền nhận dữ liệu).
 - + Giao tiếp với ứng dụng Android.
- Web server Blynk:
 - + Truyền và nhận dữ liệu.
 - + Đồng bộ điều khiển giữa các thiết bị, ứng dụng Android và nút điều khiển.
- Ứng dụng Android (Blynk):
 - + Giúp tương tác với bộ xử lý trung tâm thông qua wifi.
 - + Giám sát nhiệt độ, độ ẩm, độ sáng đèn và có thể điều chỉnh từ xa.

3.3. Thiết kế sơ đồ hệ thống.



Sơ đồ mô hình



Sơ đồ khối

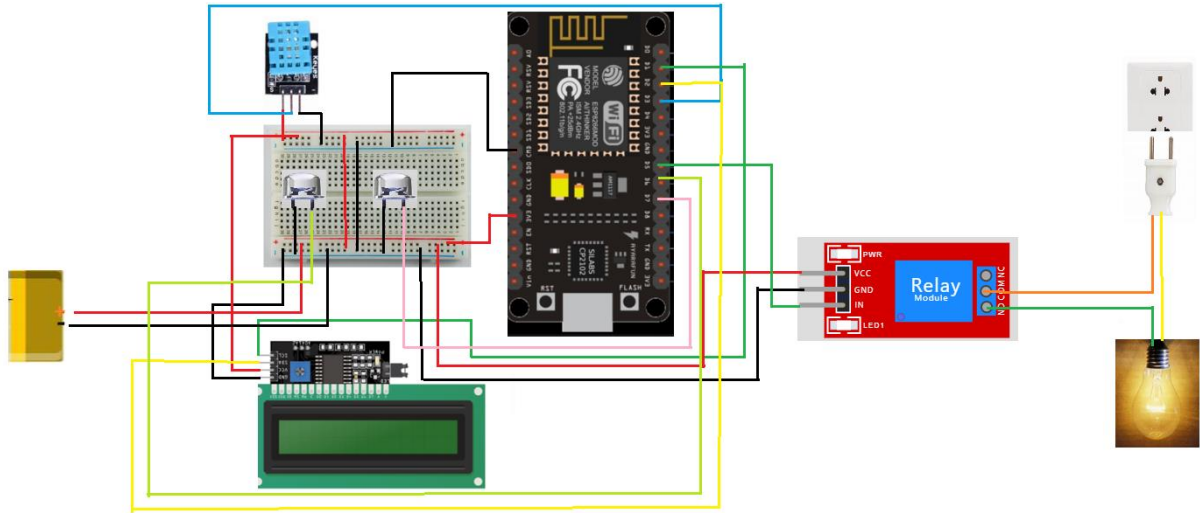
- Chức năng từng khối:

- + Khối nguồn: cung cấp nguồn cho toàn bộ hệ thống.
- + Khối xử lý trung tâm: điều khiển mọi sự hoạt động của hệ thống theo chương trình đã nạp sẵn. Nhận dữ liệu từ khối cảm biến, xử lý rồi gửi tín hiệu đến khối truyền và nhận nếu có yêu cầu.
- + Khối truyền và nhận: nhận dữ liệu từ tin nhắn của điện thoại để đưa đến khối xử lý trung tâm sau đó khối xử lý trung tâm sẽ xử lý và gửi dữ liệu đến điện thoại thông qua khối truyền và nhận.
- + Khối cảm biến: thu thập dữ liệu từ các cảm biến nhiệt độ, độ ẩm, cường độ sáng và đưa đến khối xử lý trung tâm.
- + Khối chấp hành: Điều chỉnh cường độ sáng theo lệnh từ khối xử lý.
- + Web server: Nhận dữ liệu điều khiển từ khối xử lý trung tâm, tiến hành lưu trạng thái của thiết bị vào server, để biết được lịch sử điều khiển của thiết bị. Người dùng có thể tác động vào phần giao diện theo ý muốn. Đồng thời dữ liệu điều khiển được đồng bộ với ứng dụng Android và hệ thống.

+ Ứng dụng Android: Xử lý và gửi tín hiệu điều khiển đến vi điều khiển, điều khiển trực tiếp trên thiết bị Android.

Chương 4. THI CÔNG HỆ THỐNG.

4.1. Thực hiện lắp ráp và ghép nối các mạch và module.



- Đấu nối DHT11:
 - + VCC (+): nối với nguồn dương 3,3V
 - + GND (-): nối nguồn âm (GND)
 - + Signal: nối với chân D3
- Đấu nối Relay với bóng đèn: cơ chế dùng dòng điện nhỏ để điều khiển nguồn điện lớn
 - + VCC: nối nguồn dương 3,3V
 - + GND: nối nguồn âm
 - + Chân IN: nối với chân D5
 - + Chân NC: không nối
 - + Chân COM: nối với 1 chân của nguồn điện lớn
 - + Chân NO: nối với 1 chân bóng đèn
 - + Còn chân còn lại của nguồn điện lớn nối với chân còn lại của bóng đèn
- Đấu nối LCD với module điều khiển I2C:
 - + Nối các chân oled với I2C như trong hình
- Đấu nối I2C điều khiển với esp8266:

- + GND: nối với nguồn âm
- + VCC: nối với nguồn dương
- + SDA: nối với chân D2
- + SCL: nối với chân D1
- Đối với LED:
 - + Đèn 1: một chân nối nguồn âm, một chân nối với chân D6
 - + Đèn 2: một chân nối nguồn âm, một chân nối với chân D7

4.2. Lập trình hệ thống.

- **Phần mềm lập trình cho vi điều khiển:** dùng Arduino IDE
- + Khai báo các thư viện sử dụng:

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <BlynkSimpleEsp8266.h>
```

- + Khai báo Blynk Auth Token:

```
char auth[] = "84NFK0f7n6y24Je00XyseSXVyvmQuJKY";
```

- + Khai báo SSID, mật khẩu Wifi đang sử dụng để kết nối:

```
char ssid[] = "firebase";
char password[] = "12345678";
```

- + Khai báo các chân LED, relay kết nối với Arduino và khai báo DHT (chân DHTPIN là D3, DHTTYPE là DHT11):

```
#define DHTPIN D3
#define DHTTYPE DHT11
int LED[2] = {D6,D7};
int relay=D5;
unsigned long currentTime;
unsigned long lastTime;

DHT dht(DHTPIN, DHTTYPE);
```

- + Ở hàm **setup()**:

Kết nối Wifi cho ESP8266, nếu thành công sẽ in IP ra serial:

```
WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
}

Serial.println("\nWiFi connected");
Serial.println("IP Address: " + WiFi.localIP().toString());
```

Khởi tạo kết nối Wire giữa chân D1 (SCL) và chân D2 (SDA) cho màn hình LED LCD OLED 16x2:

```
Wire.begin(D2,D1); //D2: SDA, D1: SCL
lcd.init(); //khởi động màn hình
delay(100);
lcd.backlight();
lcd.clear();
```

Kết nối Blynk và DHT11:

```
Blynk.begin(auth, ssid, password);
dht.begin();
```

+ Ở hàm **loop()**: Code sẽ được lặp đi lặp lại:

Kết nối Blynk và khai báo các biến temperature và humidity để đo nhiệt độ, độ ẩm:

```
Blynk.run();

float humidity = dht.readHumidity();
float temperature = dht.readTemperature();
readAndSendSensorData(temperature, humidity);
```

Cập nhật nhiệt độ, độ ẩm lên màn hình LCD sau mỗi 1s:

```
currentTime=millis();
lcd.display();
if ((currentTime-lastTime)>=1000)
{
  lcdDisplay(temperature, humidity);
  lastTime=currentTime;
  Serial.println("Ham");
}
```

+ Các hàm con:

Hàm đọc và ghi nhiệt độ, độ ẩm, hiển thị lên Blynk. Đồng thời xử lý điều kiện, nếu nhiệt độ vượt khoảng (0,15) hoặc độ ẩm vượt khoảng (85,95) thì đèn cảnh báo (đèn màu đỏ) sẽ được bật lên.

```
void readAndSendSensorData(float temperature, float humidity) {

    if (!isnan(humidity) && !isnan(temperature)) {
        Serial.print("Humidity: ");
        Serial.print(humidity);
        Serial.print("% Temperature: ");
        Serial.println(temperature);

        Blynk.virtualWrite(V5, temperature);
        Blynk.virtualWrite(V6, humidity);
        if (humidity>95||humidity<85)
            digitalWrite(LED[0], HIGH);
        else if (temperature<0 || temperature>15)
            digitalWrite(LED[0], HIGH);
        else digitalWrite (LED[0], LOW);
    }
    else {
        Serial.println("Failed to read from DHT sensor!");
    }
}
```

+ Hàm lcdDisplay được gọi khi cần hiển thị nhiệt độ, độ ẩm lên LCD:

```
void lcdDisplay(float t, float h)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Nhiệt độ: ");
    lcd.setCursor(10,0);
    lcd.print(round(t));
    lcd.setCursor(0,1);
    lcd.print("Độ ẩm: ");
    lcd.setCursor(10,1);
    lcd.print(round(h));
}
```

+ Lấy trạng thái Blynk điều khiển đèn thông qua hàm **BLINK_WRITE()** được hỗ trợ của thư viện Blynk:

```
BLYNK_WRITE(V1) {
  control(LED[0], param.asInt());
}
BLYNK_WRITE(V2) {
  control(LED[1], param.asInt());
}
BLYNK_WRITE(V3) {
  control(relay, param.asInt());
} //relay
```

- **Về web của nhóm: sử dụng Python web app và Streamlit**

Để sử dụng web app, chạy setup_run.bat để vào được web với Local URL:

<http://localhost:8501> hoặc **Network URL: <http://192.168.103.141:8501>**

+ Khi vừa truy cập vào web app, ta sẽ thấy giao diện đăng nhập/đăng ký:

Gmail Login

Enter your email:

Enter your password:

Enter your token:

Register

Login

+ Ngay bên dưới phần đăng nhập/đăng ký là Light Controller dùng để điều khiển các đèn:



Arduino Web-Application

Light controller

Light 1

Light 1: On ●

Light 2

Light 2: Off ●

Light 3

Light 3: Off ●

+ Tiếp theo là phần cài đặt cho việc gửi thông báo, có thể đặt lịch để gửi email định kỳ:

Send me a noti

Send me an instant email

Select time interval to send email:

5 minutes

Send me scheduled emails

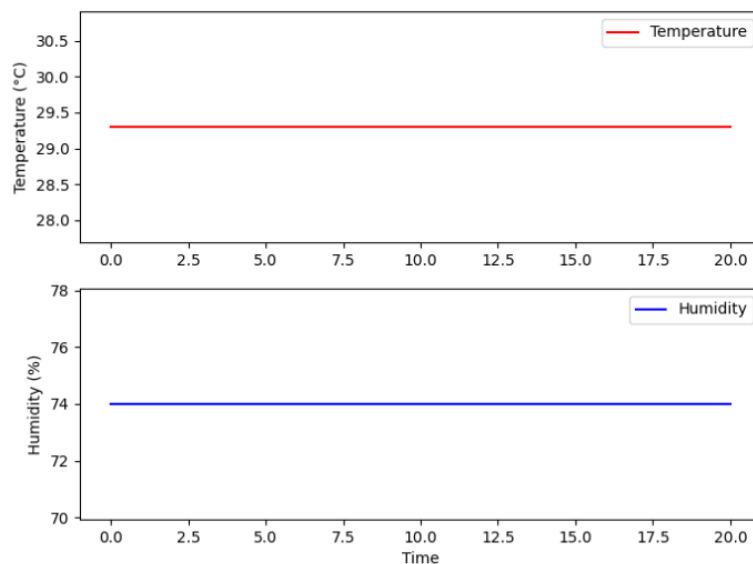
+ Cuối cùng là phần hiển thị nhiệt độ, độ ẩm, và hai biểu đồ biểu thị sự thay đổi của nhiệt độ, độ ẩm:

TnH reporter

Temperature: 29.3°C

Humidity: 74.0%

Time: 23:29:09



- Về app của nhóm: sử dụng ngôn ngữ Java trên IDE Android Studio và Realtime Database của Firebase.

+ Sơ lược về code của app:

--Lắng nghe sự thay đổi của đèn 1 và xử lý bật tắt đèn 1, các đèn còn lại tương tự:


```
light1Ref.addValueEventListener(new ValueEventListener() {
    2 usages
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        Boolean isLight1On = dataSnapshot.getValue(Boolean.class);
        // Xử lý trạng thái đèn 1
        lightSwitch1.setChecked(isLight1On != null && isLight1On); // Đèn 1 bật
    }
});
```

```
lightSwitch1.setOnCheckedChangeListener((buttonView, isChecked) -> {
    // Xử lý logic bật/tắt đèn 1
    if (isChecked) {
        turnOnLight1();
    } else {
        turnOffLight1();
    }
});
```

--Xử lý nút đăng xuất:

```
// Xử lý đăng xuất
firebaseAuth = FirebaseAuth.getInstance();
logoutButton.setOnClickListener(v -> {
    firebaseAuth.signOut();
    Toast.makeText(context: MainActivity.this, text: "Đăng xuất thành công", Toast.LENGTH_SHORT).show();
});
```

--Hàm cập nhật nhiệt độ, độ ẩm từ realtime database và kiểm tra xem các thông số có nằm trong khoảng quy định hay không:

```
private void updateTemperatureAndHumidity() {
    DatabaseReference databaseRef = FirebaseDatabase.getInstance().getReference( path: "sensor_data");
    databaseRef.addValueEventListener(new ValueEventListener() {
        2 usages
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (dataSnapshot.exists()) {
                // Lấy dữ liệu nhiệt độ và độ ẩm từ dataSnapshot
                double temperature = dataSnapshot.child( path: "temperature").getValue(Double.class);
                double humidity = dataSnapshot.child( path: "humidity").getValue(Double.class);

                // Kiểm tra nhiệt độ và độ ẩm
                if (temperature >= 0 && temperature <= 10 && humidity >= 65 && humidity <= 75) {
                    // Hiển thị nhiệt độ và độ ẩm trong giao diện người dùng
                    temperatureTextView.setText(String.valueOf(temperature));
                    humidityTextView.setText(String.valueOf(humidity));
                } else {
                    // Gửi cảnh báo qua email
                    temperatureTextView.setText(String.valueOf(temperature));
                    humidityTextView.setText(String.valueOf(humidity));
                    Toast.makeText(getApplicationContext(), text: "Cảnh báo: Nhiệt độ, độ ẩm không nằm trong khoảng quy định!", Toast.LENGTH_SHORT).show();
                    //sendNotificationEmail();
                }
            }
        }
    });
}
```

--Xử lý sự kiện cho nút đăng nhập:

```
buttonLogin.setOnClickListener(view -> {
    String email = editTextEmail.getText().toString();
    String password = editTextPassword.getText().toString();

    // Xác thực người dùng
    FirebaseAuth.getInstance().signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(task -> {
            if (task.isSuccessful()) {
                Toast.makeText(context: LoginActivity.this, text: "Đăng nhập thành công", Toast.LENGTH_SHORT).show();

                /*
                // Hiển thị tài khoản đang đăng nhập
                FirebaseAuth mAuth = FirebaseAuth.getInstance();
                FirebaseUser currentUser = mAuth.getCurrentUser();

                if (currentUser != null) {
                    String email1 = currentUser.getEmail();
                    setContentView(R.layout.activity_main);
                    TextView userTextView = findViewById(R.id.userTextView);
                    userTextView.setText(email1);
                }*/

                // Quay lại màn hình chính
                Intent intent = new Intent(context: LoginActivity.this, MainActivity.class);
                startActivity(intent);
            } else {
                Toast.makeText(context: LoginActivity.this, text: "Đăng nhập thất bại", Toast.LENGTH_SHORT).show();
            }
        })
    }
```

--Xử lý sự kiện nút đăng ký:

```
buttonRegister.setOnClickListener(v -> {
    String email = editTextEmail.getText().toString().trim();
    String password = editTextPassword.getText().toString().trim();

    // Kiểm tra hợp lệ
    if (TextUtils.isEmpty(email)) {
        Toast.makeText(getApplicationContext(), text: "Vui lòng nhập email!", Toast.LENGTH_SHORT).show();
        return;
    }

    if (TextUtils.isEmpty(password)) {
        Toast.makeText(getApplicationContext(), text: "Vui lòng nhập mật khẩu!", Toast.LENGTH_SHORT).show();
        return;
    }

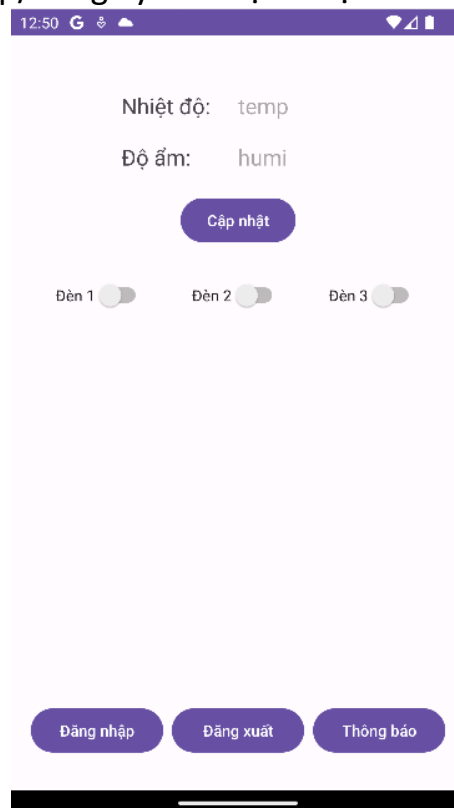
    // Đăng ký người dùng mới trên Firebase
    firebaseAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(activity: RegisterActivity.this, task -> {
            if (task.isSuccessful()) {
                Toast.makeText(getApplicationContext(), text: "Đăng ký thành công!", Toast.LENGTH_SHORT).show();
                // Chuyển về LoginActivity
                Intent intent = new Intent(context: RegisterActivity.this, LoginActivity.class);
                startActivity(intent);
                finish();
            } else {
                // Đăng ký thất bại
                Toast.makeText(getApplicationContext(), text: "Đăng ký thất bại: " + Objects.requireNonNull(task.getException()).getMessage(),
                    Toast.LENGTH_LONG).show();
            }
        })
    }
```

--Xử lý việc kiểm tra nhiệt độ, độ ẩm và hiển thị thông báo:

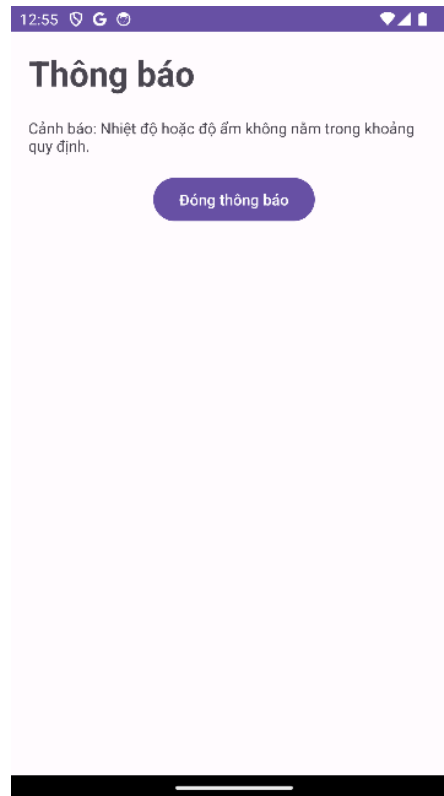
```
// Kết nối với Firebase Realtime Database và lắng nghe sự thay đổi
FirebaseDatabase.getInstance().getReference( path: "sensor_data").addValueEventListener(new ValueEventListener() {
    2 usages
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        // Lấy nhiệt độ và độ ẩm từ Firebase
        double temperature = dataSnapshot.child( path: "temperature").getValue(Double.class);
        double humidity = dataSnapshot.child( path: "humidity").getValue(Double.class);

        // Kiểm tra nhiệt độ và độ ẩm
        if (temperature < 0 || temperature > 10 || humidity < 65 || humidity > 75) {
            // Hiển thị thông báo lên TextView
            textViewContent.setText("Cảnh báo: Nhiệt độ hoặc độ ẩm không nằm trong khoảng quy định.");
            showNotification( title: "Cảnh báo", message: "Nhiệt độ hoặc độ ẩm không nằm trong khoảng quy định.");
        } else {
            // Nếu nhiệt độ và độ ẩm nằm trong khoảng yêu cầu, xóa nội dung TextView
            textViewContent.setText("");
        }
    }
}
```

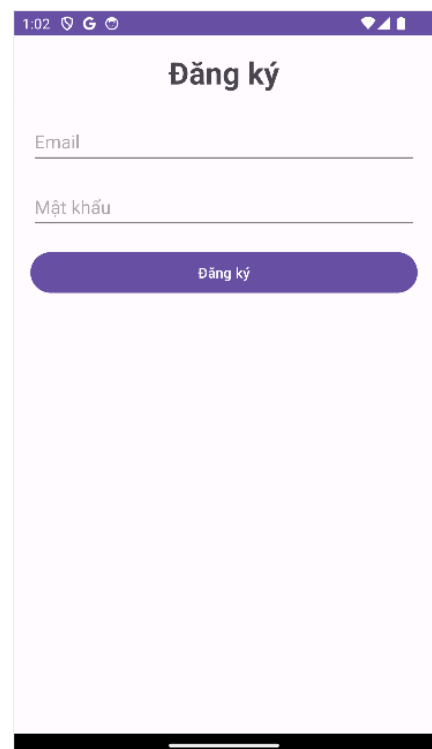
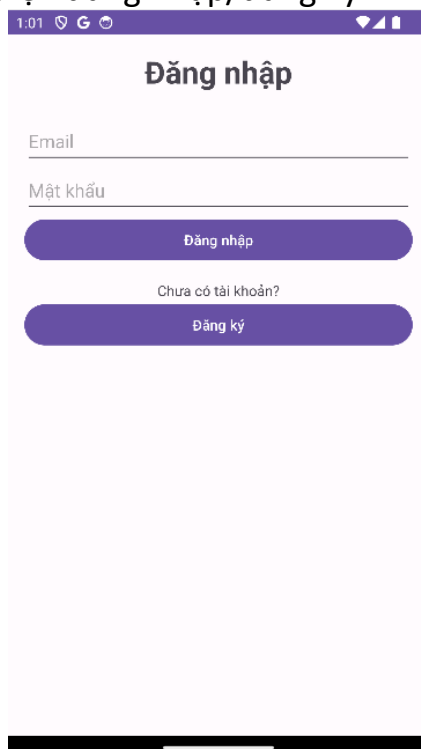
+ Giao diện chính: có các chức năng hiển thị nhiệt độ, độ ẩm, nút điều khiển các đèn, nút đăng nhập/đăng ký để nhận được email, nút xem thông báo:



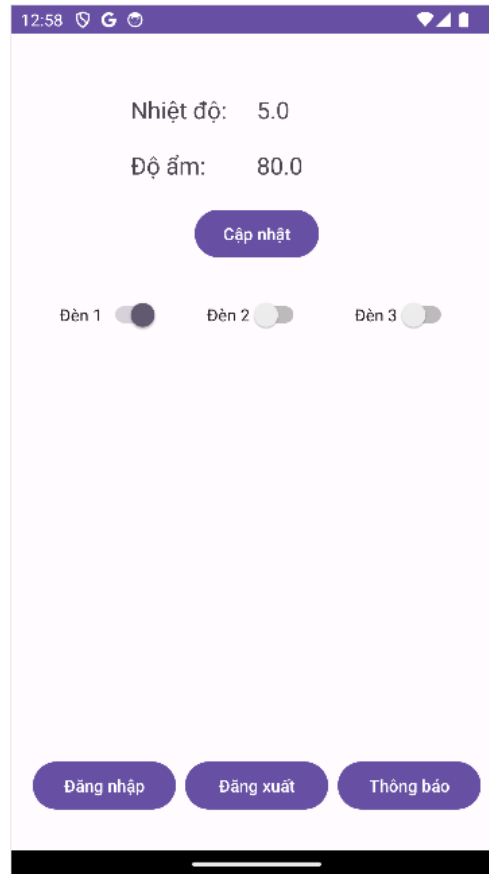
+ Giao diện xem thông báo: nếu nhiệt độ, độ ẩm không nằm trong khoảng quy định thì điện thoại sẽ hiện thông báo, đồng thời nếu đã đăng nhập thì sẽ gửi email cảnh báo:



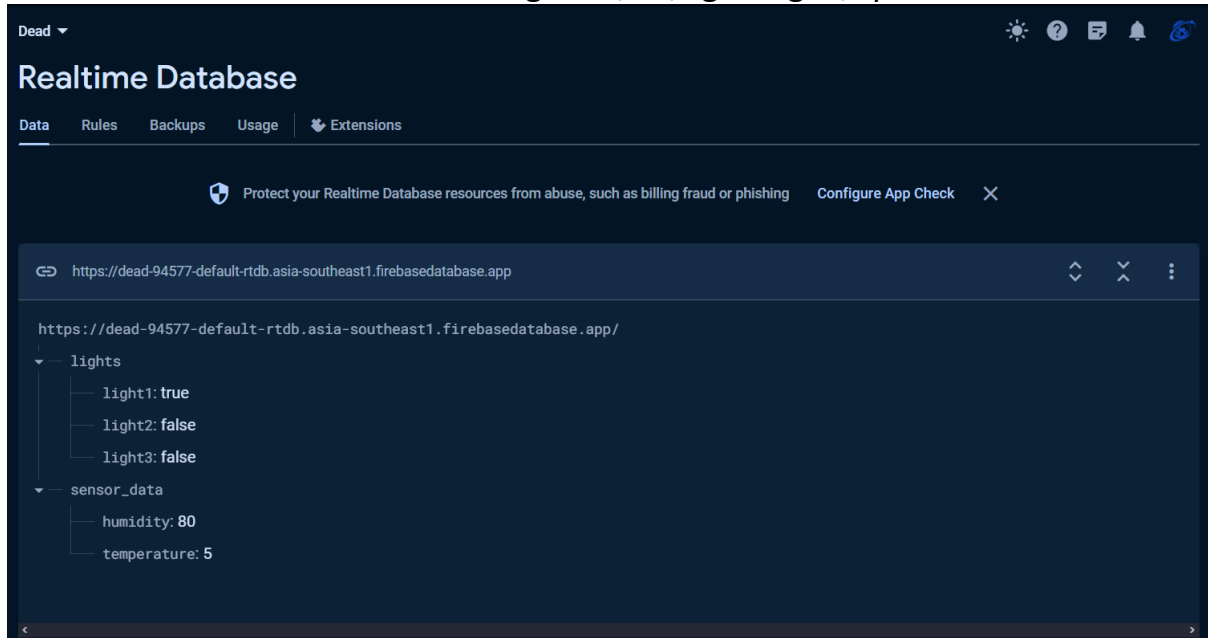
- Giao diện đăng nhập/đăng ký:



+ App đã kết nối được với realtime database:
Nhấn nút cập nhật trên app và bật đèn 1:



Realtime Database cũng đã tự động đồng bộ qua:



--> Tuy nhiên, app của nhóm em vẫn chưa kịp hoàn thành nên chưa thể thực hiện demo, một số lỗi của app vẫn chưa giải quyết được như: chưa kết nối được với mạch, chưa gửi được email cảnh báo... Cuối cùng nhóm em quyết định chỉ sử dụng web python như đã mô tả ở phần trên.

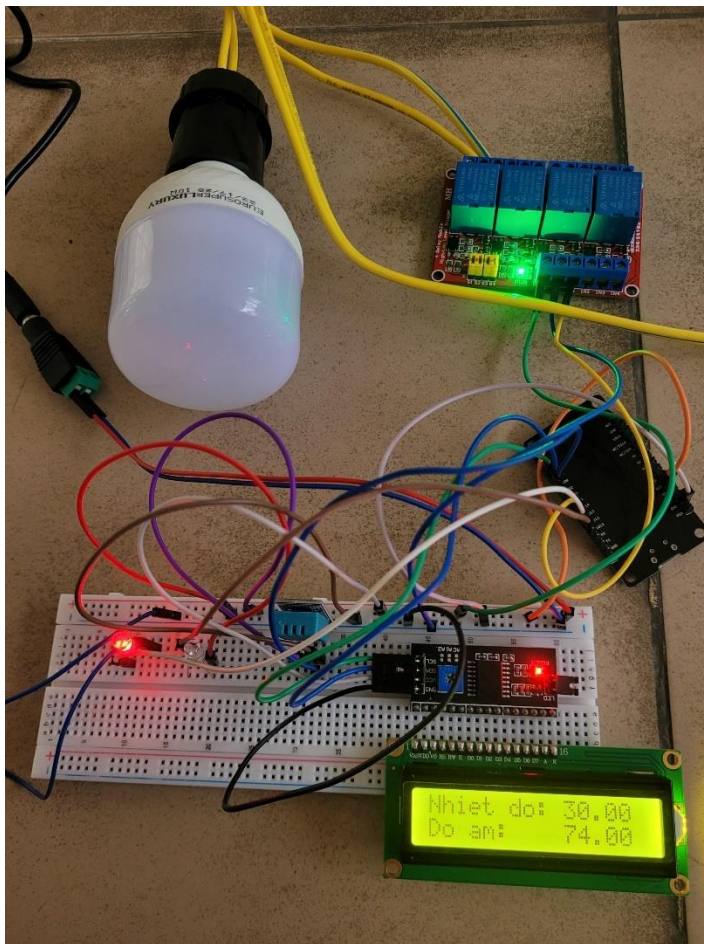
Chương 5. KẾT QUẢ, NHẬN XÉT, ĐÁNH GIÁ.

5.1. Kết quả đạt được.

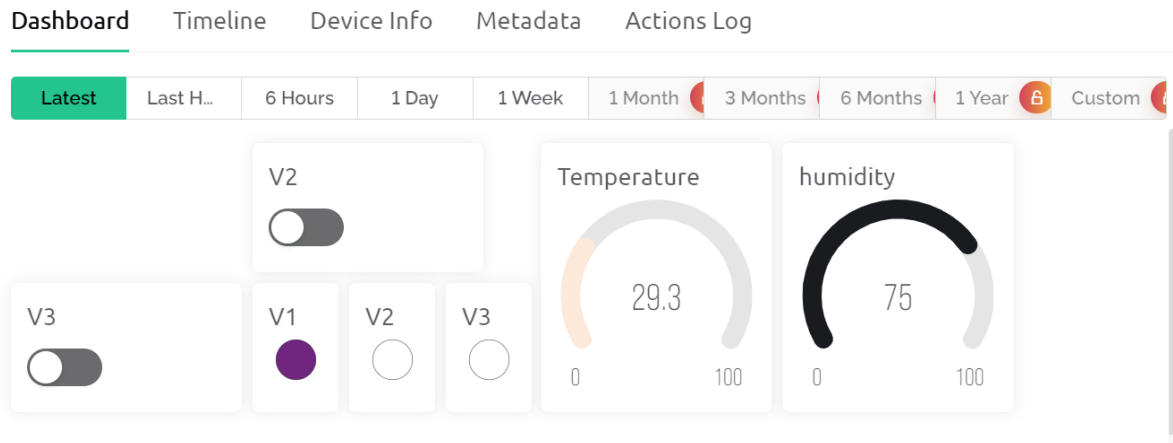
- Nhóm đã hoàn thành những mục tiêu đề án đã đặt ra là triển khai một thiết bị với các chức năng giám sát nhiệt độ, độ ẩm trên ứng dụng và web, chức năng nhận thông báo khi gặp sự cố, điều khiển thiết bị điện từ xa.
- Qua đề án này, nhóm em đã thực hiện được một thiết bị giám sát nhiệt độ, độ ẩm và điều khiển thiết bị thông qua công nghệ wifi, hiểu được cách thức hoạt động của ESP8266, cũng như những ứng dụng của thiết bị này.
- Chúng em cũng học được rất nhiều thứ về lập trình trên Arduino IDE, lập trình ứng dụng android, làm web...

5.2. Kết quả thực nghiệm.

- Ảnh mạch thực tế: Trên màn hình LCD đã hiển thị được nhiệt độ, độ ẩm.



- Trên web Blynk cũng đã đọc được thông tin nhiệt độ, độ ẩm từ cảm biến:



- Trên web app của nhóm cũng được đồng bộ:

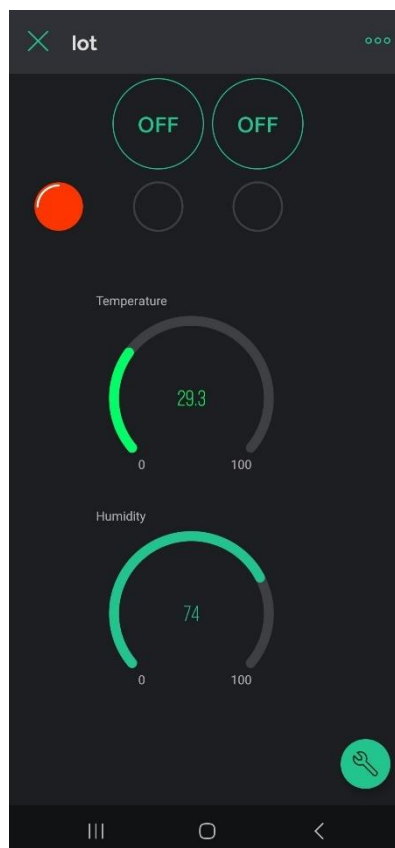
TnH reporter

Temperature: 29.3°C

Humidity: 74.0%

Time: 23:05:52

- Đồng thời, trên app Blynk cũng đã được đồng bộ:



5.3. Nhận xét và đánh giá.

- Blynk app đồng bộ còn nhiều hạn chế về thời gian.
- Nhóm gặp nhiều khó khăn khi lên kế hoạch xây dựng một ứng dụng android thay thế Blynk app, do không lường trước được độ khó nên chúng em đã không ưu tiên làm ứng dụng android cho thiết bị. Sau cùng nhóm đã không thể hoàn thành ứng dụng android một cách hoàn chỉnh để báo cáo đúng hạn.
- Tuy nhiên, thiết bị của chúng em cũng đã đồng bộ khá tốt lên Blynk và web app của nhóm.

Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.

6.1. Ưu điểm và nhược điểm.

- Ưu điểm:
 - + Việc sử dụng tạo sự thuận tiện cho người dùng khi quản lý từ xa.
 - + Có thể giám sát nhiệt độ độ ẩm từ xa.
 - + Có thể điều khiển thiết bị điện từ xa.
- Nhược điểm:
 - + Hoạt động còn phụ thuộc nhiều vào tình hình mạng.
 - + Độ chính xác nhiệt độ, độ ẩm khi sử dụng DHT11 chưa cao.
 - + Chỉ sử dụng được ở những môi trường có sử dụng mạng.

6.2. Hướng phát triển.

- Đối với việc điều khiển thiết bị điện:
 - + Có thể thêm việc điều khiển đèn bằng giọng nói.
 - + Điều khiển tự động thông qua việc sử dụng các cảm biến, như tắt đèn tự động khi trời sáng, bật đèn tự động khi trời tối qua việc sử dụng thêm cảm biến ánh sáng.
- Có thể mở rộng việc điều khiển:
 - + Điều khiển thêm đèn, quạt.
 - + Điều chỉnh độ sáng đèn.
 - + Điều chỉnh tốc độ quạt.
 - + Điều chỉnh nhiệt độ.
- Có thể ứng dụng nhiều hơn trong các mô hình thực tế hỗ trợ việc giám sát và các mô hình sử dụng công nghệ thông minh như smart home.

HẾT