**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC KINH TẾ - KỸ THUẬT CÔNG NGHIỆP**

**KHOA ĐIỆN TỬ & KỸ THUẬT MÁY TÍNH**

**BỘ MÔN LẬP TRÌNH DI ĐỘNG**

**BÁO CÁO ĐỒ ÁN**

**NGÀNH: KỸ THUẬT MÁY TÍNH**

**THIẾT KẾ, XÂY DỰNG HỆ THỐNG GIÁM SÁT THỜI TIẾT SỬ DỤNG IOT**

**Sinh viên thực hiện : Nguyễn Hữu Việt Anh**

**Kim Văn Hòa**

**Đỗ Minh Hiệp**

**Lớp : DHKM15A2HN**

**Nhóm : 8**

**Giảng viên hướng dẫn : THS. Vũ Trung Dũng**

**Hà Nội – 2024**

# 

# LỜI CAM ĐOAN

Chúng em xin cam đoan đồ án : “Thiết kế , xây dựng hệ thống giám sát thời tiết sự dụng iot”do chúng em tự thiết kế dưới sự hướng dẫn của giảng viên. Các số liệu và kết quả chưa từng được công bố.

Để hoàn thành báo cáo này chúng em chỉ sử dụng những tài liệu được thì trong danh mục tài liệu tham khảo và không sao chép hay sử dụng bất kỳ tài liệu nào khác. Nếu phát hiện có sự sao chép chúng em xin chịu hoàn toàn trách nhiệm.

Hà Nội, ngày tháng năm 2024

Sinh viên

(*ký, ghi rõ họ tên*)

Hòa

Kim Văn Hòa

# 

# LỜI CẢM ƠN

Trong những tháng ngày căng thẳng và tận hiến vào việc học tập, chúng em không thể không bày tỏ lòng biết ơn sâu sắc đến những người đã đồng hành và ủng hộ chúng em trong quá trình hoàn thành báo cáo đồ án này.

Đầu tiên, chúng em muốn gửi lời cảm ơn đặc biệt đến gia đình và người thân của mình. Họ đã luôn ở bên cạnh, không chỉ bằng tình yêu thương mà còn bằng những lời khích lệ và sự động viên tinh thần. Những khoảnh khắc khó khăn và đắng cay trở nên dễ dàng hơn nhờ tình thương vô bờ bến của họ.

Tiếp theo và quan trọng nhất, chúng em xin gửi lời cảm ơn đến giảng viên hướng dẫn của mình. Sự kiên nhẫn, kiến thức sâu rộng và sự hướng dẫn tận tâm của thầy đã giúp chúng em định hình và hoàn thiện dự án này. cô đã là nguồn động viên và kiến thức quý báu cho chúng em trong suốt quá trình này.

Bạn bè cũng đã đóng vai trò quan trọng trong hành trình này. Họ luôn ở đó để chia sẻ niềm vui và nỗi buồn, để động viên và hỗ trợ khi chúng em cảm thấy mệt mỏi và bất ổn. Họ là nguồn động viên tinh thần không thể thiếu.

Cuối cùng, chúng em cảm ơn bản thân mỗi thành viên trong nhóm vì sự kiên nhẫn, sự cống hiến và sự nỗ lực không ngừng nghỉ trong suốt thời gian dự án. Đây là một bước quan trọng trong hành trình của chúng em, và chúng em sẽ luôn nhớ đến sự hỗ trợ và động viên từ mọi người. Đồ án chuyên ngành đã mang đến cho chúng em nhiều bài học quý báu và kỷ niệm đáng trân trọng.

Xin chân thành cảm ơn !

**MỤC LỤC**

[**LỜI CAM ĐOAN................................................................................2**](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.30j0zll)

[**LỜI CẢM ƠN................**](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1fob9te) **.....................................................................3**

[**MỤC LỤC...........**](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3znysh7)**................................................................................4**

[**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT..................................5**](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2et92p0)

[**DANH MỤC HÌNH VẼ......................................................................5**](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.tyjcwt)

[**TÓM TẮT ĐỒ ÁN..............................................................................7**](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3dy6vkm)

CHƯƠNG I: TỔNG QUAN ĐỀ TÀI...................................................8

[1.1 Đặt vấn đề: .................................................................................8](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2s8eyo1)

[1.2 Mục đích đề tài...........................................................................9](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3rdcrjn)

1.3 Kết luận chương 1.......................................................................9

[CHƯƠNG II: CƠ SỞ Lý THUYẾT...................................................10](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.26in1rg)

[2.1 Hệ thống tự động sử dụng iot....................................................10](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.lnxbz9)

[2.2 Ưu điểm và nhược điểm của hệ thống ......................................12](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2jxsxqh)

[2.3 Ứng dụng của hệ thống thời tiết trong nông nghiệp……..........13](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.z337ya)

2.4 Kết luận chương 2 .....................................................................19

[Chương III: THIẾT KẾ HỆ THỐNG](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1pxezwc) ................................................20

[3.1 Giới thiệu về mô hình hệ thống.................................................](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.49x2ik5) 20

[3.2 Thiết kế sơ đồ khối cho hệ thống..............................................20](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2p2csry)

[3.3Chức năng của từng khối............................................................20](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3o7alnk)

[3.3.1 Khối vi điều khiển..............................................................](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.23ckvvd) 21

[3.3.2 Khối thu thập dữ liệu](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1hmsyys) ..........................................................25

[3.3.3 Khối hiển thị](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2u6wntf) ....................................................................28

[3.3.4 Khối thiết bị........................................................................33](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.28h4qwu)

3.4 Kết luận chương 3......................................................................33

[Chương IV: THI CÔNG HỆ THỐNG](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.37m2jsg) ...............................................34

[4.1 Mô Hình Thực Nghiệm](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1mrcu09)..............................................................34

[4.1.1 Phần mềm mô phỏng Altium..............................................34](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.46r0co2)

[4.1.2 Sơ đồ nguyên lí...................................................................](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.111kx3o) 35

[4.1.3 Nối dây cho mạch...............................................................](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1egqt2p) 38

[4.2 Mạch in thủ công.......................................................................41](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1rvwp1q)

[4.3 Gắn linh kiện.............................................................................42](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2r0uhxc)

[4.4 Lập trình hệ thống.....................................................................43](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.25b2l0r)

[4.4.1 Giới thiệu phần mềm Arduino...........................................](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.kgcv8k) 43

[4.4.2 Lưu đồ giải thuật................................................................48](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1x0gk37)

4.4.2.1. Lưu đồ thuật toán chương trình chính.........................48

4.4.2.3 Lưu đồ gửi dữ liệu........................................................50

[4.4.3 Android studio code...........................................................50](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1baon6m)

4.5 Kết luận chương 4......................................................................52

[Chương V: KẾT QUẢ........................................................................53](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.pkwqa1)

[5.1 Kết quả thu được.......................................................................53](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.39kk8xu)

[5.1.1 Sử dụng module ESP32 Node MCU..................................53](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1opuj5n)

[5.1.2 Sử dụng cảm biến...............................................................53](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.48pi1tg)

[5.1.3 Lập trình Android studio code...........................................53](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2nusc19)

[5.2 Kết quả thực nghiệm................................................................53](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1302m92)

[Chương VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....................55](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.319y80a)

6.Giới thiệu chung..........................................................................55

[6.2 Kết luận.....................................................................................55](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1gf8i83)

[6.3 Hướng phát triển đề tài.............................................................56](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.40ew0vw)

[TÀI LIỆU THAM KHẢO..................................................................56](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2fk6b3p)

[PHỤ LỤC...........................................................................................57](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.upglbi)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

LCD: Liquid-Crystal Display

MIT: Massachusetts Institute of Technology

PCB: Printed Circuit Board

IDE: Integrated Development Environment

LED: Light Emitting Diode

OS: Operating System

**DANH MỤC HÌNH VẼ**

[Hình 1.1 Ảnh hưởng của thời tiết đến nông nghiệp.............................8](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.17dp8vu)

[Hình 2.1 Robot cảnh sát giao thông....................................................10](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.35nkun2)

[Hình 2.2 Mô hình trạm giám sát thời tiết……....................................11](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1ksv4uv)

[Hình 2.3 Hệ thống tưới nước tự động.................................................12](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.44sinio)

[Hình 2.4 Hệ thống giám sát thời tiết IOT...........................................13](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3j2qqm3)

[Hình 3.1 Sơ đồ khối của hệ thống......................................................15](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2xcytpi)

[Hình 3.2 ESP32…………………………..........................................17](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3whwml4)

[Hình 3.3 Sơ đồ chân của ESP32 NodeMCU......................................18](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.qsh70q)

[Hình 3.4 Cảm biến nhiệt độ độ ẩm DHT 11.......................................20](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.147n2zr)

[Hình 3.5 Cảm biến DHT 11 loại 3 chân.............................................22](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.ihv636)

[Hình 3.6 Cảm biến DHT 11 loại 4 chân.............................................23](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.32hioqz)

[Hình 3.7 Module Cảm biến mưa........................................................25](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.vx1227)

[Hình 3.8 Module Cảm biến ánh sáng.................................................26](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3fwokq0)

[Hình 3.9 Sơ đồ cảm biến ánh sáng.....................................................26](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1v1yuxt)

[Hình 3.10 cảm biến áp suất................................................................27](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.4f1mdlm)

[Hình 3.11 Máy bơm nước 1 chiều R385 5-12V.................................29](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.19c6y18)

[Hình 4.1 Phần mềm Altium Designer................................................34](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2lwamvv)

[Hình 4.2 Sơ đồ nguyên lí mạch giám sát thời tiết..............................36](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3l18frh)

[Hình 4.3 Giao diện layout linh kiện trên phần mềm altium...............36](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.206ipza)

[Hình 4.4 Hình ảnh 3D của mạch điện................................................37](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.4k668n3)

[Hình 4.5 Mạch in thủ công................................................................38](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2zbgiuw)

[Hình 4.6](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3ygebqi)  Mạch điện hoàn chỉnh sau khi hiệu chỉnh………………..39

[Hình 4.7 Hướng dẫn cài đặt Arduino IDE bằng hình ảnh..................39](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2dlolyb)

[Hình 4.8 Giao diện Arduino IDE...............](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.sqyw64)....................................... 40

[Hình 4.9 Tải package cho NodeMCU................................................](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3cqmetx) 40

[Hình 4.10 Hộp thoại Board Manager..................................................41](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.4bvk7pj)

[Bảng 4.11 Chọn board........................................................................42](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1664s55)

[Hình 4.12 Chọn cổng kết nối..............................................................44](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.34g0dwd)

[Hình 4.13 Lưu đồ thuật toán chương trình chính................................44](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1jlao46)

[Hình 4.14 Lưu đồ gửi dữ liệu..............................................................46](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.43ky6rz)

[Hình 4.15 Giao diện Android Studio..................................................46](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2iq8gzs)

[Hình 4.16 Giao diện chính của phần mềm..........................................47](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.xvir7l)

[Hình 4.17 Chọn cổng kết nối..............................................................47](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3hv69ve)

Hình 4.18 Lưu đồ thuật toán chương trình chính................................48

Hình 4.19 Lưu đồ thuật toán chế độ tự động bật bơm.........................49

[Hình 4.20 Lưu đồ gửi dữ liệu.............................................................](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2w5ecyt) 50

[Hình 4.22 Giao diện chính của phần mềm..........................................52](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2afmg28)

[Hình 5.1 Sản phầm sau khi hoàn thành..............................................53](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.3mzq4wv)

[Hình 5.2 Lịch sử lưu trong App.........................................................](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2250f4o) 54

[Hình 5.3 Thông tin được gửi lên app.................................................](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.haapch) 54

**TÓM TẮT ĐỒ ÁN**

Trong bối cảnh cuộc cách mạng công nghiệp 4.0, ngành nông nghiệp đang đối diện với yêu cầu đổi mới, bỏ qua các phương thức sản xuất cũ và lạc hậu.Đồng thời, thiên tai xảy ra ngày càng nhiều gây ra vô số thiệt hại cho người nông dân Ứng dụng các kỹ thuật tiên tiến của IoT vào nông nghiệp không chỉ tăng hiệu quả sản xuất mà còn giúp con người đưa ra các phương pháp chăm sóc, nâng cao năng suất cây trồng giảm thiểu tối đa thiệt hại do thời tiết gây ra.

Đồ án này gồm 6 chương chính, tập trung vào việc thiết kế và xây dựng hệ thống giám sát thời tiết tự động sử dụng IoT nhằm cải thiện hiệu suất giám sát điều kiện thời tiết, giúp người nông dân quản lý nông nghiệp thông minh và hiệu quả hơn. Nội dung các chương bao gồm:

Chương 1: Tổng quan đề tài “ *Thiết kế , xây dựng hệ thống giám sát thời tiết sự dụng iot”*

Chương 2: Cơ sở lí thuyết

Chương 3: Thiết kế hệ thống

Chương 4: Thi công hệ thống

Chương 5: Kết quả

Chương 6: Kết luận và hướng phát triển

Sau khi hoàn thành , kết quả phù hợp với tiêu chí đề ra giúp tiết kiệm thời gian, cảnh báo kịp thời về sự thay đổi về các điều kiện thời tiết và giảm sức người trong việc chăm sóc cây. Sản phẩm phù hợp đặt trong các vườn cây,thảm cỏ.....

**CHƯƠNG I: TỔNG QUAN ĐỀ TÀI**

* 1. **Đặt vấn đề**

Hiện nay khí hậu toàn cầu đã và đang biến đổi mạnh mẽ với tần suất và cường độ ngày càng gia tăng trên toàn thế giới đã gây ra nhiều thiệt hại nặng nề cho nhiều khu vực trên thế giới, đặc biệt tại Việt Nam đã xảy ra nhiều hiện tượng thời tiết phức tạp như: mưa, rét kéo dài, bão lốc và áp thấp nhiệt đới hoạt động bất thường không theo quy luật, mùa mưa ít mưa, hạn hán nghiêm trọng thiếu nước sinh hoạt và sản xuất trên diện rộng, nguy cơ cháy rừng rất cao,…nên rất cần có các giải pháp hoặc hệ thống, thiết bị có thể đưa ra các cảnh báo sớm, hiệu quả để hạn chế các thiệt hại do thiên tai, môi trường gây ra. Xuất phát từ nhu cầu thực tế đó, nghiên cứu này đề xuất xây dựng một hệ thống quan trắc với các tính năng cải tiến và hiệu quả hơn nhằm đáp ứng tốt hơn các yêu cầu dự báo thời tiết hiện nay.

Các quy luật khí hậu bị phá vỡ khiến ngành khí tượng thủy văn ngày càng gặp nhiều khó khăn trong việc dự báo, cảnh báo. Với sự phát triển của xã hội ngày nay, việc dự báo thời tiết đón đầu những vận động của thiên nhiên phục vụ cho đời sống cộng đồng khiến công tác khí tượng thủy văn ngày càng đóng vai trò quan trọng trong việc bảo vệ con người và tài sản. Trong khi đó Việt Nam là một quốc gia mà ngành nông nghiệp đóng vai trò chủ đạo thì nhu cầu về thông tin thời tiết, về sự thay đổi mùa, càng cần thiết hơn bao giờ hết. Để hạn chế những tác động này, các quốc gia trên thế giới và Việt Nam đã không ngừng đầu tư khoa học công nghệ, nghiên cứu các lĩnh vực khí tượng thủy văn, thời tiết. Sự phát triển của khoa học kỹ thuật trong nghiên cứu, ứng dụng và giám sát, theo dõi những biến động của thời tiết khí hậu, giới nghiên cứu khoa học khí tượng trên thế giới đã tạo ra những sản phẩm thiết yếu cho đời sống dân sinh góp phần giảm nhẹ những thiệt hại do thiên tai gây ra nhằm đảm bảo sự an toàn về tính mạng, tài sản cho toàn xã hội.

Quá trình hội nhập quốc tế đòi hỏi chất lượng nông sản càng cao; cùng với diện tích đất bị thu hẹp do đô thị hóa, do biến đổi khí hậu trong khi dân số tăng nên nhu cầu lương thực không ngừng tăng lên… là những thách thức rất lớn đối với sản xuất nông nghiệp ở Việt Nam hiện nay.

Những thách thức do thời tiết mang lại không chỉ ảnh hưởng đến năng suất và chất lượng sản phẩm, mà còn đe dọa đến sinh kế của hàng triệu nông dân Việt Nam. Điều này làm tăng nhu cầu cấp bách về việc áp dụng các giải pháp công nghệ trong giám sát, dự báo và ứng phó kịp thời với những thay đổi thời tiết, từ đó giảm thiểu thiệt hại cho ngành nông nghiệp.

Việc phát triển hệ thống giám sát thời tiết mà nhóm chúng em đề xuất sẽ góp phần hỗ trợ nông dân chủ động hơn trong việc lên kế hoạch gieo trồng, tưới tiêu, và phòng ngừa thiên tai, nhằm đảm bảo an toàn và ổn định trong sản xuất nông nghiệp, đặc biệt trong bối cảnh biến đổi khí hậu ngày càng phức tạp.



**Hình 1.1 Ảnh hưởng thời tiết đến nông nghiệp**

**1.2 Mục đích đề tài**

Đẩy mạnh việc áp dụng khoa học công nghệ cao vào quá trình thiết kế xây dựng thiết bị theo dõi các thông số thời tiết và môi trường trong việc đánh giá điều kiện thời thiết hiện tại và dự báo thời tiết tương lai. Thiết bị đảm bảo tính linh hoạt, thuận tiện trong quá trình sử dụng như: Thiết kế, chế tạo hệ thống bao gồm các cảm biến có thể đo đạc được những chỉ số quan trọng như: nhiệt độ, độ ẩm, áp suất khí quyển, tốc độ gió, lưu lượng mưa,... Thiết kế và xây dựng chương trình xử lý và gửi thông tin thống kê số liệu qua Internet từ các đầu đo cảm biến tới điện thoại di động để có thể theo dõi liên tục các chỉ số từ môi trường và thời tiết, đưa ra giải pháp xử lý kịp thời.

Mục tiêu của đề tài là nghiên cứu, phân tích và thiết kế được mạch có khả năng điều khiển và giám sát hệ thống thời tiết thông qua các thông tin thu thập về. Thông tin độ ẩm môi trường, các cảm biến đo chuyển tới khối xử lý dữ liệu. Dựa vào các thông tin trên để điểu khiển máy bơm tưới nước tự động.

**1.3 Kết luận chương 1**

Tổng kết lại chương 1 :Việt Nam là một nước nông nghiệp vẫn đóng vai trò quan trọng mà áp dụng khoa học công nghệ vào sản xuất còn ít dẫn đến năng suất và chất lượng nông sản thấp do thiên tai tàn phá .Qua đó dẫn đến sự ra đời của đề tài là một giải pháp phát triển nền nông nghiệp, mang đến nhiều lợi ích cho người nông dân.

**CHƯƠNG II: CƠ SỞ Lý THUYẾT**

**2.1 Hệ thống tự động sử dụng IoT**

Hệ thống điều khiển tự động là hệ thống, bao gồm các phần tử tự động nhằm điều khiển các quy trình xảy ra trong thiên nhiên, cuộc sống mà không có sự tham gia trực tiếp của con người.

IoT (Internet of Things) nghĩa là Internet vạn vật. Một hệ thống các thiết bị tính toán, máy móc cơ khí và kỹ thuật số hoặc con người có liên quan với nhau và khả năng truyền dữ liệu qua mạng mà không yêu cầu sự tương tác giữa con người với máy tính.

Với một hệ thống IoT chúng sẽ bao gồm 4 thành phần chính đó là thiết bị (Things), trạm kết nối (Gateways), hạ tầng mạng (Network and Cloud) và bộ phân tích và xử lý dữ liệu (Services-creation and Solution Layers).

Các cảm biến sẽ có nhiệm vụ cảm nhận các tín hiệu từ môi trường như nhiệt độ, áp suất, ánh sáng,… và chuyển chúng thành các dạng dữ liệu trong môi trường Internet. Sau đó các tín hiệu sẽ được xử lý và đưa ra các thay đổi theo ý của người tiêu dùng. Hiện nay chúng thường được ứng dụng thông qua các ứng dụng trên điện thoại hay trên máy tính,…

Hiện nay, hệ thống tự động sử dụng IoT xuật hiện ngày càng phổ biến,gần như mọi lĩnh vực trong đời sống đều gắn với hệ thống tự động

* Trong y tế:

Tự động hóa trong các quy trình khám chữa bệnh từ xa, chăm sóc sức khỏe. Nhiều  bệnh viện đã triển khai các robot trong các hoạt động khám bệnh, chẩn đoán bệnh, quản lý… Điều này giúp cho các bệnh viện, cơ sở y tế không bị quá tải và giúp bù đắp thiếu hụt nhân lực.

* Trong giao thông:



**Hình 2.1 Robot cảnh sát giao thông**

Nhiều hoạt động như tìm kiếm xe cứu hộ, phòng tránh ùn tắc giao thông đều được ứng dụng IoT. Các hệ thống sẽ giám sát, cung cấp thông tin cho người dân để nắm bắt thông tin tốt nhất. Nó đã góp phần hỗ trợ kiểm soát giao thông, giúp đỡ hoạt động giao thông của người dân.

* Trong sinh hoạt hàng ngày:

Các cảm biến được kết nối với nhau và với các thiết bị để xây dựng nhà ở thông minh giúp cuộc sống tiện nghi ,dễ dàng hơn.

* Tự động hóa dự báo thời tiết:

Tự động hóa trong dự báo thời tiết là một bước tiến quan trọng trong việc nâng cao độ chính xác và hiệu quả của các hệ thống dự báo. Bằng cách sử dụng mạng lưới các cảm biến IoT đặt tại nhiều khu vực khác nhau, hệ thống có thể thu thập dữ liệu thời tiết theo thời gian thực như nhiệt độ, độ ẩm, áp suất khí quyển, ánh sáng và mưa. Những dữ liệu này được truyền về trung tâm xử lý, nơi các thuật toán và trí tuệ nhân tạo phân tích và đưa ra dự báo chi tiết.

Nhờ IoT, việc dự báo trở nên nhanh chóng và chính xác hơn, cung cấp thông tin kịp thời để phục vụ các hoạt động kinh tế, xã hội, và bảo vệ nông nghiệp trước thiên tai. Hệ thống tự động hóa này giúp giảm thiểu sự can thiệp của con người, tối ưu hóa nguồn lực và tăng tính liên tục trong việc theo dõi thời tiết.



**Hình 2.2 Mô hình trạm giám sát thời tiết**

* Tự động hóa tưới cây trồng:



**Hình 2.3 Hệ thống tưới cây tự động**

Tự động hóa IoT trong tưới cây trồng mang lại giải pháp thông minh và hiệu quả cho việc quản lý nguồn nước trong nông nghiệp. Bằng cách sử dụng các cảm biến IoT để đo độ ẩm đất, nhiệt độ môi trường và lượng mưa, hệ thống có thể thu thập dữ liệu theo thời gian thực và tự động điều chỉnh lượng nước tưới phù hợp cho cây trồng. Kết hợp với các bộ điều khiển tưới, hệ thống có thể lập lịch tưới tự động, đảm bảo cung cấp đủ nước khi cây cần mà không gây lãng phí.

Điều này không chỉ giúp tiết kiệm tài nguyên nước mà còn tăng năng suất cây trồng và giảm công sức của người nông dân. Nhờ IoT, việc chăm sóc cây trồng trở nên chính xác, tối ưu và dễ dàng hơn trong mọi điều kiện thời tiết.

**2.2 Ưu điểm và nhược điểm của hệ thống IoT**

**Ưu điểm**

* Tăng thông lượng hoặc năng suất.
* Truy cập thông tin từ mọi lúc, mọi nơi trên mọi thiết bị.
* Cải thiện việc giao tiếp giữa các thiết bị điện tử được kết nối.
* Chuyển dữ liệu qua mạng Internet giúp tiết kiệm thời gian và tiền bạc.
* Tự động hóa các nhiệm vụ giúp cải thiện chất lượng dịch vụ của doanh nghiệp.
* Cải thiện mạnh mẽ (thống nhất), quy trình hay sản phẩm.
* Tăng tính nhất quán của đầu ra.
* Giảm chi phí nhân công trực tiếp và chi phí nhân lực
* Thực hiện nhiệm vụ đó là vượt quá khả năng của con người về kích thước, trọng lượng, tốc độ, sức chịu đựng, vv

**Khó khăn**

* Khi nhiều thiết bị được kết nối và nhiều thông tin được chia sẻ giữa các thiết bị, thì hacker có thể lấy cắp thông tin bí mật cũng tăng lên.
* Các doanh nghiệp có thể phải đối phó với số lượng lớn thiết bị IoT và việc thu thập và quản lý dữ liệu từ các thiết bị đó sẽ là một thách thức.
* Nếu có lỗi trong hệ thống, có khả năng mọi thiết bị được kết nối sẽ bị hỏng.
* Vì không có tiêu chuẩn quốc tế về khả năng tương thích cho IoT, rất khó để các thiết bị từ các nhà sản xuất khác nhau giao tiếp với nhau.

**2.3 Ứng dụng của hệ thống giám sát thời tiết, tưới cây theo thời gian thực.**

Hệ thống giám sát thời tiết và tưới cây tự động đang trở thành một giải pháp tiên tiến và thiết yếu trong nông nghiệp hiện đại. Ứng dụng công nghệ IoT vào các hệ thống này giúp người nông dân có thể theo dõi các yếu tố môi trường như nhiệt độ, độ ẩm, lượng mưa, và ánh sáng một cách chính xác và theo thời gian thực. Dựa trên dữ liệu thu thập, hệ thống tưới cây tự động sẽ điều chỉnh lượng nước cung cấp cho cây trồng phù hợp với nhu cầu thực tế, giúp cây phát triển khỏe mạnh và tăng năng suất.

Việc tưới cây theo thời gian thực không chỉ giảm thiểu sự lãng phí nước mà còn giảm công sức lao động, tiết kiệm chi phí, và tối ưu hóa quy trình sản xuất nông nghiệp. Đặc biệt, hệ thống này còn giúp người nông dân ứng phó hiệu quả hơn với biến đổi khí hậu và thời tiết thất thường, góp phần nâng cao hiệu quả kinh tế trong nông nghiệp.



**Hình 2.4 Hệ thống giám sát thời tiết IOT**

**2.4 Kết luận chương 2**

Trong chương 2 này chúng em đã giới thiệu về hệ thống IoT , ứng dụng trong thực tiễn của Iot, ưu nhược điểm của hệ thống. Ngoài ra em còn giới thiệu về ứng dụng của hệ thống giám sát thời tiết, tưới cây theo thời gian thực.

Qua việc nghiên cứu và triển khai hệ thống giám sát thời tiết có thể thấy rõ rằng công nghệ IoT đã mang lại những cải tiến đột phá cho ngành nông nghiệp. Hệ thống không chỉ giúp người nông dân kiểm soát nhiệt độ , độ ẩm , áp suất của môi trường, giúp cho con người quản lý các thay đổi của môi trường phát hiện những cơn mưa bất thường để chủ động cảnh báo đến người dùng giúp giảm thiểu sức lao động của con người.

Ngoài ra hệ thống còn tự động hóa quy trình tưới tiêu dựa trên các dữ liệu thời tiết theo thời gian thực, đặt lịch tưới cây giám sát lưu lượng nước hệ thống đảm bảo cây trồng được cung cấp đúng lượng nước cần thiết, cải thiện năng suất và giảm thiểu tác động tiêu cực từ thời tiết khắc nghiệt.

Kết quả đạt được đã đáp ứng tốt các mục tiêu đề ra, đồng thời mở ra tiềm năng ứng dụng rộng rãi trong các mô hình canh tác quy mô lớn và những vùng có điều kiện khí hậu khắc nghiệt, cảnh báo cho người dùng về sự thay đổi của thời tiết. Hệ thống này là một giải pháp toàn diện, góp phần đưa nông nghiệp truyền thống lên một tầm cao mới trong thời đại công nghiệp 4.0.

**Chương III: THIẾT KẾ HỆ THỐNG**

**3.1 Giới thiệu về mô hình hệ thống**

Hệ thống gồm 4 phần chính :

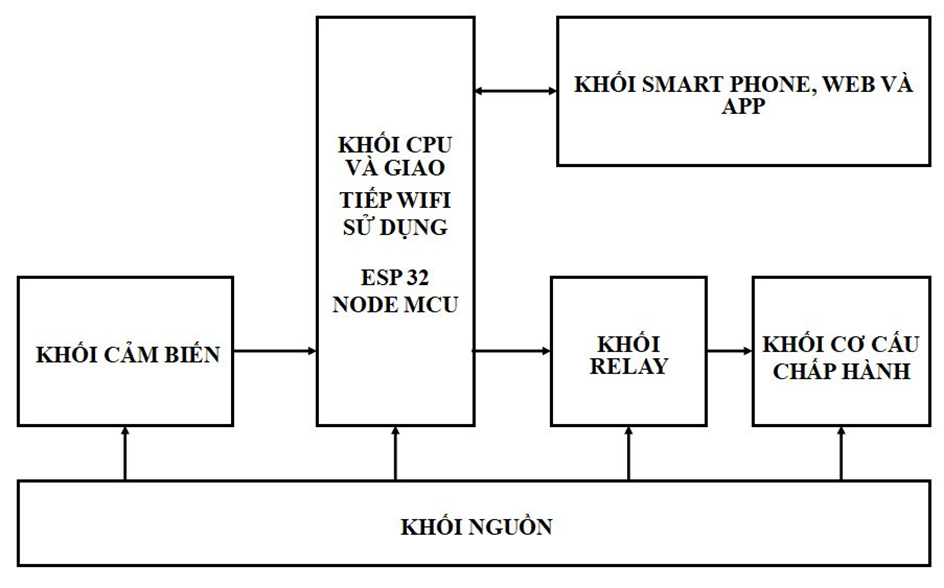
Phần thứ nhất là khối thu thập dữ liệu gồm các cảm biến nhiệt độ-độ ẩm DHT-11, cảm biến mưa, cảm biến ánh sáng, cảm biến áp suất . Các cảm biến này sẽ thu thập thông tin trong môi trường hiện tại sau đó sẽ gửi thông tin đến trung tâm xử lí bằng công nghệ truyền dẫn không dây.

Phần thứ hai là bộ điều khiển trung tâm có nhiệm vụ nhận tín hiệu gửi về từ khối thu thập dữ liệu xử lí . Thông tin sẽ được gửi đến người dùng qua wifi.

Phần thứ ba là khối thiết bị gồm có máy bơm nước. Khi đạt đủ điều kiện hệ thống điều khiển sẽ bật ( tắt ) máy bơm cung cấp nước cho cây trồng.

Phần thứ 4 là khối hiển thị và điều khiển gồm có hiển thị lên smart phone giúp người dùng theo dõi giám sát các thông số thời tiết, theo dõi lại lịch sử hằng ngày ngoài ra hệ thống còn có thể theo dõi thông qua web một cách dễ dàng và thuận tiện thông qua kết nối Internet.

**3.2 Thiết kế sơ đồ khối cho hệ thống**



**Hình 3.1 Sơ đồ khối của hệ thống**

**3.3 Chức năng của từng khối**

***3.3.1 Khối vi điều khiển***

* Đây là bộ xử lý chính của ESP32, quản lý và điều phối tất cả hoạt động của hệ thống, cung cấp khả năng kết nối không dây WiFi cho ESP32 cho phép nó giao tiếp với các thiết bị khác hoặc mạng Internet.

***3.3.2 Khối cảm biến***

* Kết nối với các cảm biến để thu thập thông tin từ môi trường xung quanh, như nhiệt độ, độ ẩm, mưa, ánh sáng, áp suất, cảm biến thời gian thực DS3231 và gửi về ESP32 để xử lý và điểu khiển các thiết bị trong nhà thông minh.

***3.3.3 Khối cơ cấu chấp hành***

* Điều khiển các cơ cấu chấp hành như động cơ, relay, hoặc các thiết bị đầu cuối khác để thực hiện các hành động cụ thể.

***3.3.4 Khối nguồn***

* Cung cấp nguồn điện cho toàn bộ hệ thống ESP32, đảm bảo hoạt động ổn định và liên tục.

***3.3.5 Khối smart phone, web***

* Theo dõi và giám sát từ xa các thông số của cảm biến, giúp người dùng có thể đưa ra dự báo và phát hiện sự thay đổi của thời tiết ảnh hưởng đến nông nghiệp làm giảm thiệt hại do thiên tai gây ra đồng thời làm giảm sức lao động cho người nông dân. Ngoài ra, còn hiển thị lịch sử từng khung giờ từng ngày trên App/Web giúp con người theo dõi liên tục thông số, giám sát thời tiết 24/24.Thiết lập hẹn giờ tưới cây theo sự điều khiển thông qua app giúp tiết kiệm lưu lượng nước, tăng sức sinh trưởng của cây trồng.

***3.3.6 Khối relay***

* Khối relay được sử dụng để điều khiển máy bơm đảm bảo hoạt động ổn định bật tắt củ bơm theo ý định của người dùng.

**3.4 Giới thiệu phần cứng**

***3.4.1 KIT điều khiển ESP32 NodeMCU***

* ESP32 là một hệ thống vi điều khiển trên chip (SoC) giá rẻ của Espressif Systems, nhà phát triển của ESP8266 SoC. Nó là sự kế thừa của SoC ESP8266 và có cả hai biến thể lõi đơn và lõi kép của bộ vi xử lý 32-bit Xtensa LX6 của Tensilica với Wi-Fi và Bluetooth tích hợp.
* Điểm tốt về ESP32, giống như ESP8266 là các thành phần RF tích hợp của nó như bộ khuếch đại công suất, bộ khuếch đại nhận tiếng ồn thấp, công tắc ăng-ten, bộ lọc và Balun RF. Điều này làm cho việc thiết kế phần cứng xung quanh ESP32 rất dễ dàng vì bạn cần rất ít thành phần bên ngoài.
* Một điều quan trọng khác cần biết về ESP32 là nó được sản xuất bằng công nghệ 40 nm công suất cực thấp của TSMC. Vì vậy, việc thiết kế các ứng dụng hoạt động bằng pin như thiết bị đeo, thiết bị âm thanh, đồng hồ thông minh, ..., sử dụng ESP32 sẽ rất dễ dàng.
* ESP32 hỗ trợ nhiều loại thiết bị ngoại vi đầu vào ( đọc dữ liệu từ bên ngoài) và đầu ra (gửi lện/tín hiệu ra bên ngoài ) như cảm ứng điện dung, 12C, DAC, PWM, UART,..... để bạn tự do làm các dự án điện tử mà mình thích.

Ảnh có chứa đồ điện tử, Kỹ thuật điện, Linh kiện điện, Thành phần mạch điện

Mô tả được tạo tự động

**Hình 3.2 ESP32**

* **Các tính năng của ESP 32 bao gồm:**
  + **Bộ xử lý:**

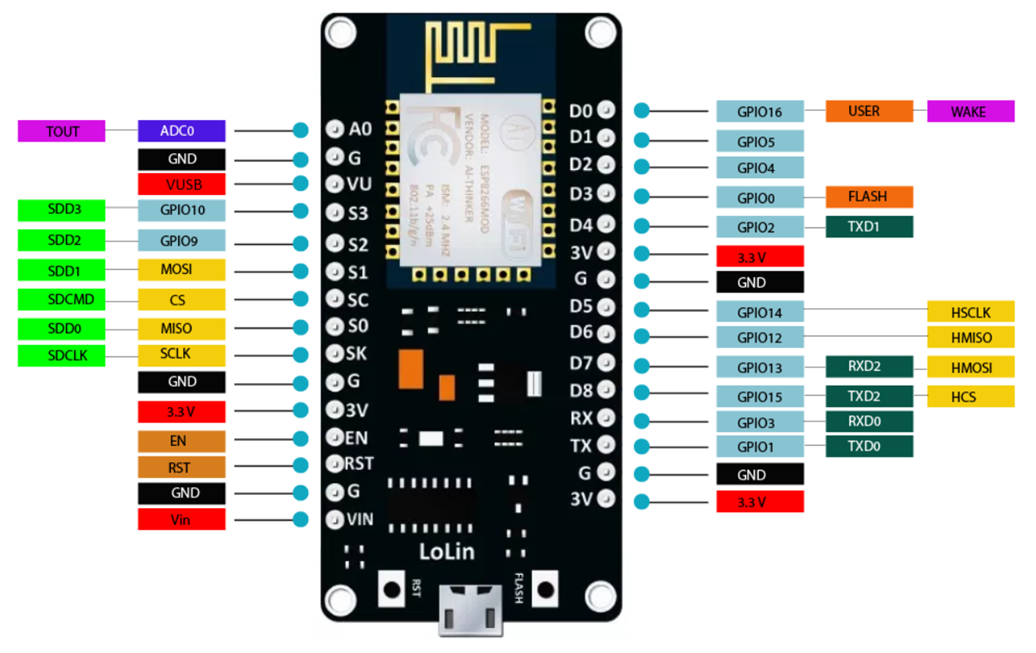
CPU: Bộ vi xử lý Xtensa lõi kép (hoặc lõi đơn) 32-bit LX6, hoạt động ở tần số 240 MHz (160 MHz cho ESP32-S0WD và ESP32-U4WDH)[]](https://vi.wikipedia.org/wiki/ESP32#cite_note-FOOTNOTEESP32_Series_Datasheet202122-4) và hoạt động ở tối đa 600 MIPS (200 MIPS với ESP32-S0WD/ESP32-U4WDH).

Bộ đồng xử lý (co-processor) công suất cực thấp (Ultra low power, viết tắt:

ULP) hỗ trợ việc đọc ADC và các ngoại vi khi bộ xử lý chính ( main processor )

vào chế độ deep sleep.

* + **Hệ thống xung nhịp:** CPU Clock, RTC Clock và Arduino PLL Clock.
  + **Kết nối không dây:** Wifi, Bluetooth: v4.2 BR/EDR and BLE
  + **Hỗ trợ tất cả các loại giao tiếp:** Analog(ADC) , I²C, UART, SPI, …
* **Sơ đồ chân ESP32 Node MCU:**



**Hình 3.3** **Sơ đồ chân của ESP32 NodeMCU**

Chức năng của các chân:

* VCC: Điện áp 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.
* 10 chân GPIO từ D0 – D8, có chức năng PWM, IIC, giao tiếp SPI, 1- Wire và ADC trên chân A0.

Tính năng của NODEMCU ESP32:

* WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n, hỗ trợ WPA/WPA2.
* Điện áp cung cấp : DC 5 ~ 9V.
* Bộ nhớ Flash: 16MB.
* Chuẩn giao tiếp nối tiếp UART với tốc độ Baud lên đến115200
* Tích hợp ngăn xếp giao thứcTCP / IP.
* Tích hợp chuyển đổi TR, balun, LNA, bộ khuếch đại công suất và phù hợp với mạng.
* Hỗ trợ nhiều loại anten.
* Wake up và truyền các gói dữ liệu trong <2ms.
* Hỗ trợ cả 2 giao tiếp TCP và UDP.
* Hỗ trợ các chuẩn bảo mật như: OPEN, WEP, WPA\_PSK, WPA2\_PSK, WPA\_WPA2\_PSK
* Có 3 chế độ hoạt động: Client, Access Point, Both Client and Access Point.
* Quản lý năng lượng NODEMCU ESP32
* ESP32 được thiết kế cho điện thoại di động, điện tử lắp ráp và ứng dụng InternetofThings với mục đích đạt được mức tiêu thụ điện năng thấp nhất với sự kết hợp của nhiềukỹ thuật độc quyền. Kiến trúc tiết kiệm năng lượng hoạt động trong 3 chế độ: chế độ hoạt động, chế độ ngủ và chế độ ngủ sâu.
* Bằng cách sử dụng các kỹ thuật quản lý nguồn điện và kiểm soát chuyển đổi giữa chế độ ngủ ESP32 tiêu thụ chưa đầy 12uA ở chế độ ngủ nhỏ hơn 1.0mW so với (DTIM = 3) hoặc ít hơn 0.5mW (DTIM = 10) để giữ kết nối với các điểm truy cập.
* Khi ở chế độ ngủ, chỉ có bộ phận hiệu chỉnh đồng hồ thời gian thực và cơ quan giám sát vẫn hoạt động. Đồng hồ thời gian thực có thể được lập trình để đánh thức ESP32 ở bất kỳ khoảng thời gian cần thiết nào.
* ESP32 có thể được lập trình để thức dậy khi một điều kiện chỉ định được phát hiện. Tính năng tối thiểu thời gian báo thức này của ESP32 có thể được sử dụng bởi Tính năng tối thiểu thời gian báo thức của ESP32 có thể được sử dụng bởi thiết bị di động SOC. Cho phép chúng vẫn ở chế độ chờ, điện năng thấp cho đến khi Wifi là cần thiết.
* Để đáp ứng nhu cầu điện năng của thiết bị di động và điện tử lắp giáp, ESP32 có thể được lập trình để giảm công suất đầu ra của PA phù hợp với các ứng dụng khác nhau. Bằng việc tắt khoảng tiêu thụ năng lượng.

Các chip có thể được thiết lập ở các trạng thái sau:

* **OFF**: chân CHIP\_PD ở mức thấp. Các RTC (đồng hồ thời gian) bị vô hiệu hóa và mọi thanh ghi sẽ bị xóa.
* **SLEEP DEEP:** Các RTC được kích hoạt, khi đó các phần còn lại của chip sẽ ở trạng thái off. RTC phục hồi bộ nhớ nội bộ để lưu trữ các thông tin kết nối WiFi cơ bản.
* **SLEEP**: Chỉ RTC hoạt động. Các dao động tinh thể được vô hiệu hóa. Bất kỳ sự kiện wakeup (MAC, host, RTC hẹn giờ, ngắt ngoài) sẽ đưa chip vào trạng  thái wakeup.
* **Wakeup:** Trong trạng thái này, hệ thống đitừ trạng thái ngủ sang trạng thái PWR. Các dao động tinh thể và PLLs được kích hoạt.
* **Trạng thái ON:** Xung clock tốc độ cao hoạt động và gửi đến mỗi khối được kích hoạtbằng cách đăng ký kiểm soát xung clock. Mức độ thấp hơn clock gating được thực hiện ở cấp khối, bao gồm cả CPU, có thể đạt được bằng cách sử dụng lệnh WAIT, trong khi hệ thống trên off.

***3.4.2 Cảm biến đo nhiệt độ - độ ẩm DHT11***

Khối cảm biến là thiết bị  điện tử cảm nhận những trạng thái hay quá trình vật lý, hóa học hay sinh học của môi trường cần khảo sát, và biến đổi thành tín hiệu điện để thu thập thông tin về trạng thái hay quá trình đó.

* Cảm biến nhiệt độ độ ẩm

A blue and black electronic device

Description automatically generated

**Hình 3.4 Cảm biến nhiệt độ độ ẩm DHT 11**

DHT11 Là cảm biến nhiệt độ, độ ẩm rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1-wire ( giao tiếp digital 1-wire truyền dữ liệu duy nhất). Cảm biến được tích hợp bộ tiền xử lý tín hiệu giúp dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ tính toán nào.

* **Đặc điểm:**
* Điện áp hoạt động : 3V - 5V (DC)
* Dải độ ẩm hoạt động : 20% - 90% RH, sai số ±5%RH
* Dải nhiệt độ hoạt động : 0°C ~ 50°C, sai số ±2°C
* Tần số lấy mẫu tối đa: 1 Hz ( 1 giây/lần )
* Khoảng cách truyển tối đa: 20m
* **Sơ đồ chân Cảm biến DHT11:** gồm 2 chân cấp nguồn, và 1 chân tín hiệu. Hiện nay, thông dụng ngoài thị trường có hai loại đóng gói cho DHT11: 3 chân và 4 chân.

A blue electronic device with a white background

Description automatically generated

**Hình 3.5** **Cảm biến DHT 11 loại 3 chân**

A close-up of a blue electronic device

Description automatically generated

**Hình 3.6** **Cảm biến DHT 11 loại 4 chân**

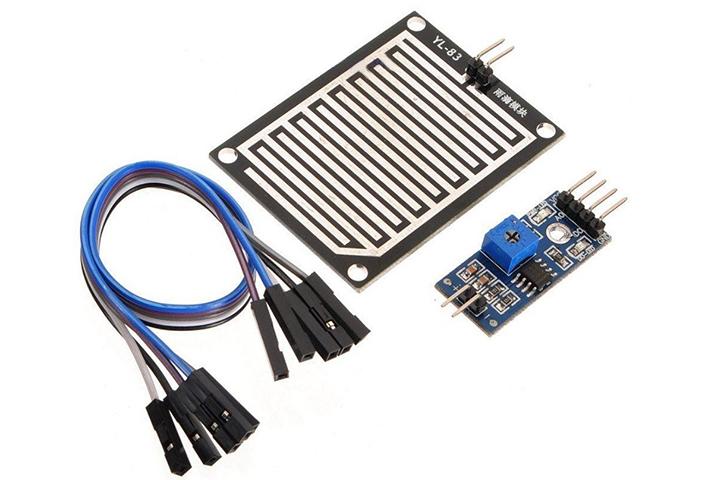
* **Nhận xét:** Cảm biến nhiệt độ, độ ẩm DHT11 với giá thành rẻ, dễ sử dụng, thích hợp sử dụng trong các ứng dụng yêu cầu độ chính xác không cao, môi trường không khắc nghiệt.
* **Nguyên lý hoạt động:**

Để có thể giao tiếp với DHT11 theo chuẩn 2 chân vi xử lý thực hiện theo 2 bước:

* Gửi tín hiệu muốn đo (Start) tới DHT11 xác nhận lại.
* Khi giao tiếp được với DHT11, Cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được.
* **Ứng dụng:**

Cảm biến nhiệt độ, độ ẩm DHT11 được sử dụng trong các máy điều hòa, trong các máy sấy và trong các trạm quan trắc thời tiết, trong công nghiệp vì nó nhỏ gọn giá thành rẻ dễ dàng lắp đặt và sử dụng.

***3.4.3 Cảm biến mưa***



**Hình 3.7** **Module Cảm biến mưa**

Mạch cảm biến mưa là một [module cảm biến](https://chotroihn.vn/module-cam-bien) được sử dụng rộng rãi trong việc phát hiện mưa vì ưu điểm dễ dàng lắp đặt và chi phí thấp. Cảm biến hoạt động bằng cách so sánh điện áp của mạch ngoài trời với giá trị đã được đặt trước thông qua biến trở trên cảm biến. Từ đó module dễ dàng điều khiển đóng cắt [relay](https://chotroihn.vn/module-relay).

* **Thông số kỹ thuật:**
* Kích thước: 5.4\*4.0 mm
* Dày 1.6 mm
* Điện áp: 5V
* Led báo nguồn ( Màu xanh)
* Led cảnh báo mưa ( Màu đỏ)
* Hoạt động dựa trên nguyên lý: Nước rơi  vào phần cảm biến sẽ tạo điện ra áp trên cấc chân D0 A0

Có 2 dạng tín hiệu Analog (A0) và Digital (D0)

Dạng tín hiệu : TTL, đầu ra 100mA ( Có thể sử dụng trực tiếp Relay, Còi công suất nhỏ...)

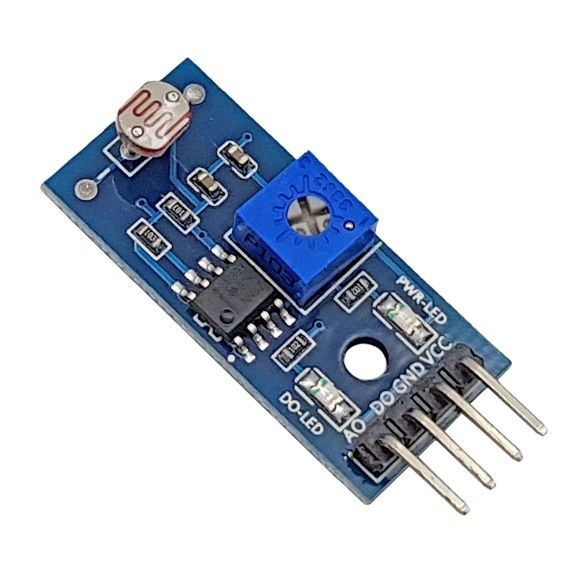
Điều chỉnh độ nhạy bằng biến trở.Sử dụng LM358 để chuyển AO --> DO

Nguyên lí hoạt động:

* Khi có nước rơi trên cảm biến, sẽ có điện áp trong khoảng từ 0V đến 5V trên chân A0 và được đưa vào bộ so sánh sử dụng ic LM393, để đưa ra chân D0 điện áp mức 0 hoặc mức 1. Biến trở có tác dụng điều chỉnh đó nhạy, bạn có thể tùy ý quyết định với lượng mưa nào thì cảm biến sẽ đưa ra mức 1.
* Ngoài ra, cảm biến còn đưa trực tiếp chân A0 ra cho các bạn có thể tiến hành đo lường, xác định lưu lượng mưa bằng cách giao tiếp với vi điều khiển và các bộ chuyển đổi ADC.

Trong đề tài này cảm biến mưa có tác dụng để ngắt máy bơm nếu trời mưa nhằm tiết kiệm nước và đảm bảo cho cây trồng không bị úng nước.

***3.4.4 Cảm biến ánh sáng***



**Hình 3.8 Module Cảm biến ánh sáng**

Cảm biến ánh sáng sử dụng quang trở có khả năng thay đổi điện trở theo cường độ ánh sáng chiếu vào. Tín hiệu xuất ra của cảm biến là digital HIGH (5V) và LOW tượng trưng cho các trạng thái bật, tắt thiết bị điện tự động mà bạn không cần phải thao tác vào !

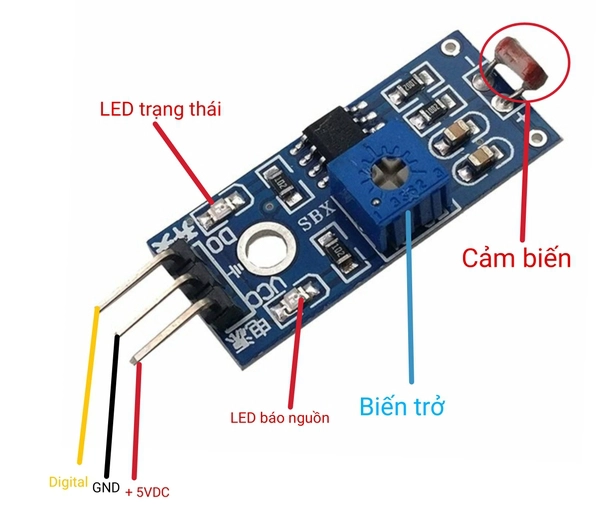
* **Ưu điểm:**
* Nhỏ gọn.
* Độ chính xác cao.
* Các thành phần phụ như điện trở, tụ điện... cần thiết cho mạch đã được gắn đầy đủ. Bạn chỉ cần cấp nguồn, nối dây điều khiển vào relay là có thể tắt/mở bóng đèn hay các thiết bị điện khác theo cường độ ánh sáng chiếu vào cảm biến.
* Giá thành thấp, khoảng 50.000 đồng.
* Sử dụng điện áp chuẩn 5V tương thích với nền tảng Arduino.
* **Cấu tạo:**

Cấu tạo của cảm biến ánh sáng khá đơn giản sẽ bao gồm: mắt cảm biến/ đầu dò ánh sáng, dây dẫn, chíp xử lý biến đổi thành tín hiệu điện. Các linh kiện bên trong được bảo vệ bằng lớp vỏ ngoài bảo vệ sensor ánh sáng khỏi va đập hoặc tác động lực từ dị vật ảnh hưởng.

* **Chức năng và ứng dụng :**

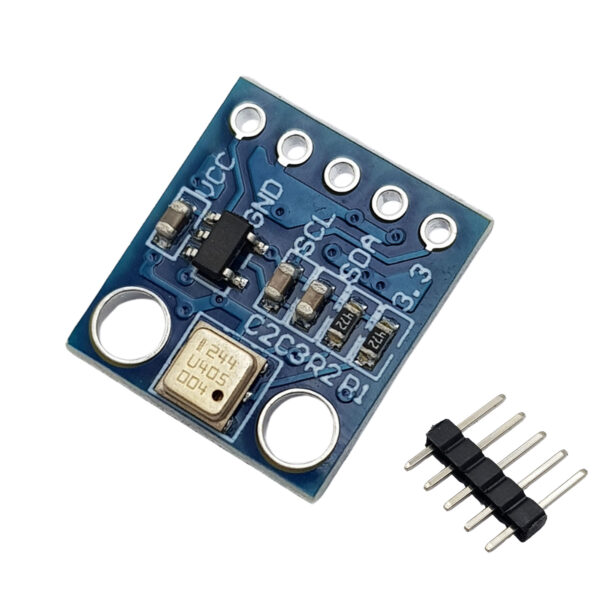
Sensor ánh sáng có độ nhạy cao, có thể xác định vật thể mà không cần tiếp xúc. Ứng dụng Sensor ánh sáng trong thiết bị đếm tốc độ cao trong các băng chuyền sản xuất, kiểm tra hàng hóa.

Sensor ánh sáng có thể điều chỉnh cường độ ánh sáng phù hợp với điều kiện môi trường. Ứng dụng của sensor ánh sáng trong màn hình điện thoại, ipad hay tablet thông minh, kiểm soát mức sáng của màn hình phù hợp với môi trường, giúp người dùng dễ quan sát.

Ứng dụng sensor ánh sáng trong hệ thống an ninh, bảo mật hoặc hệ thống đèn thông minh chiếu sáng hành lang, cầu thang…

**Hình 3.9** **Sơ đồ cảm biến ánh sáng**

***3.4.5 Cảm biến áp suất***



**Hình 3.10 cảm biến áp suất**

Cảm biến áp suất, hay còn được gọi là cảm biến đo áp suất, cảm biến áp lực, sensor áp suất,… – là một thiết bị cơ điện tử được sử dụng vô cùng rộng rãi và phổ biến trong các ứng dụng công nghiệp với mục đích đo lường và kiểm soát giá trị áp suất, nhằm đảm bảo an toàn của hệ thống hoặc thiết bị.

* **Cấu tạo:**
* Cảm biến: là bộ phận nhận tín hiệu từ áp suất và truyền tín hiệu về khối xử lý. Tùy thuộc vào loại cảm biến mà nó chuyển từ tín hiệu cơ của áp suất sang dạng tín hiệu điện trở, điện dung, điện cảm, dòng điện … về khối xử lý.
* Khối xử lý: có chức năng nhận các tính hiệu từ khối cảm biến thực hiện các xử lý để chuyển đổi các tín hiệu đó sang dạng tín hiệu tiêu chuẩn trong lĩnh vực đo áp suất như tín hiệu ngõ ra điện áp 4 ~ 20 mA
* **Nguyên lý hoạt động:**
* Nguyên lý hoạt động cảm biến áp suất cũng gần giống như các loại cảm biến khác là cần nguồn tác động (nguồn áp suất, nguồn nhiệt,… nguồn cần đo của cảm biến loại đó) tác động lên cảm biến, cảm biến đưa giá trị về vi xử lý, vi xử lý tín hiệu rồi đưa tín hiệu ra.
* **Ứng dụng:**
* Cảm biến áp suất dùng để đo trong hệ thống lò hơi , thường được đo trực tiếp trên lò hơi. Khu vực này cần đo chính xác khá cao & phải chịu nhiệt độ cao .
* Các máy nén khí cũng cần phải đo áp suất để giới hạn áp suất đầu ra , tránh trường hợp quá áp dẩn đến hư hỏng & cháy nổ .

***3.4.7 Khối thiết bị***

A small white and black pump

Description automatically generated

**Hình 3.11 Máy bơm nước 1 chiều R385 5-12V**

Máy bơm áp lực mini 12V/12W/2L Smartpumps DP-12W2L là dòng máy bơm mini nhỏ lưu lượng 2 lít/phút có khả năng tự hút. Được dùng bơm thực phẩm, bơm làm mát máy khoan, bơm định lượng, bơm nước hồ cá, bơm tưới và bơm chất lỏng, bơm phun xịt khử trùng, bơm hóa chất diệt khuẩn, máy bơm nước rửa tay, máy bơm nước sạch và bơm phun sương.

Thông số kỹ thuật:

* Điện áp Vào Jack DC 5.5x2.1mm, DC6- 12V (đề nghị 9V-1A hoặc 12V-1A)
* Dòng Tiêu Thụ: 0.5-0.7A
* Đường kính động cơ: 27mm đường kính đầu: 44mm Tổng chiều dài: 60mm
* Lưu lượng: 1,5-2 L / Min
* Chiều dài ống hút tối đa: 2 m
* Đẩy Cao: lên đến 3 mét
* Tuổi Thọ: lên đến 2500H
* Nhiệt độ nước cho phép: lên đến 80 độ C

**3.5 Kết luận chương 3**

Chương 3 này trình bày chức năng của 4 khối:

[1.Khối vi điều khiển](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.23ckvvd)

[2. Khối thu thập dữ liệu](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.1hmsyys)

[3. Khối hiển thị](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.2u6wntf)

[4. Khối thiết bị](https://docs.google.com/document/d/19gfnw5ldz1Pg0KeAm0mtgDcv_HYyLv1e/edit#heading=h.28h4qwu)

Đưa ra thông số chi tiết về các linh kiện sử dụng trong hệ thống

**Chương IV: THI CÔNG HỆ THỐNG**

**4.1 Mô Hình Thực Nghiệm**

***4.1.1 Phần mềm mô phỏng Altium***

Close-up of a computer chip

Description automatically generated

**Hình 4.1 Phần mềm Altium Designer**

Altium Designer trước kia có tên gọi quen thuộc là Protel DXP, là một trong những công cụ vẽ mạch điện tử mạnh nhất hiện nay. Được phát triển bởi hãng Altium Limited. Altium designer là một phần mềm chuyên nghành được sử dụng trong thiết kế mạch điện tử.

Altium ngày nay đang là một trong những phần mềm vẽ mạch điện tử mạnh và được ưa chuộng ở Việt Nam. Ngoài việc hỗ trợ tốt cho hoạt động vẽ mạch,Altium còn hỗ trợ tốt trong việc quản lý mạch, trích xuất file thống kê linh kiện.

Một số điểm vượt trội của Altium:

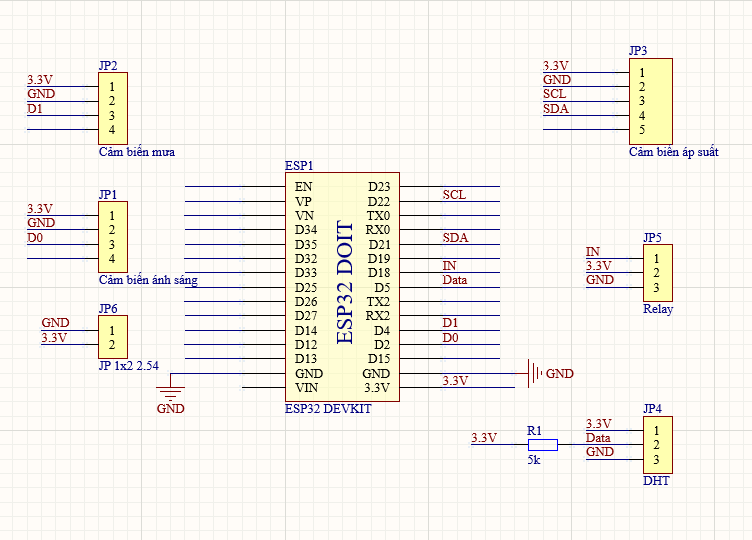
* Giao diện thiết kế, quản lý và chỉnh sửa thân thiện, dễ dàng biên dịch, quản lý file, quản lý phiên bản cho các tài liệu thiết kế.
* Hỗ trợ mạnh mẽ cho việc thiết kế tự động, đi dây tự động theo thuật toán tối ưu, phân tích lắp ráp linh kiện. Hỗ trợ việc tìm các giải pháp thiết kế hoặc chỉnh sửa mạch, linh kiện, netlist có sẵn từ trước theo các tham số mới.
* Mở, xem và in các file thiết kế mạch dễ dàng với đầy đủ các thông tin linh kiện, netlist, dữ liệu bản vẽ, kích thước, số lượng…
* Hệ thống các thư viện linh kiện phong phú, chi tiết và hoàn chỉnh bao gồm tất cả các linh kiện nhúng, số, tương tự…
* Đặt và sửa đối tượng trên các lớp cơ khí, định nghĩa các luật thiết kế, tùy chỉnh các lớp mạch in, chuyển từ schematic sang PCB, đặt vị trí linh kiện trên PCB.
* Mô phỏng mạch PCB 3D, đem lại hình ảnh mạch điện trung thực trong không gian 3 chiều, hỗ trợ MCAD-ECAD, liên kết trực tiếp với mô hình STEP, kiểm tra khoảng cách cách điện, cấu hình cho cả 2D và 3D.

***4.1.2 Sơ đồ nguyên lí***

Sơ đồ nguyên lý của mạch bao gồm việc kết nối một ESP32 với các cảm biến và thiết bị khác nhau gồm 7 thành phần chính sau:

* **ESP32 DEVKIT** **(ESP32 DOIT)**:
* Đây là trung tâm của hệ thống, nơi ESP32 được sử dụng để thu thập dữ liệu từ các cảm biến và điều khiển relay.
* Các chân GPIO của ESP32 được sử dụng để giao tiếp với các cảm biến và thiết bị ngoại vi khác.
* **Cảm biến mưa (JP2)**:
* Kết nối với chân D1 của ESP32.
* Có hai dây:
  + Chân 1 của JP2 (3.3V) được cấp nguồn 3.3V.
  + Chân 2 là GND.
  + Chân 4 (D1) là ngõ vào của tín hiệu từ cảm biến mưa về ESP32.
* **Cảm biến ánh sáng (JP1)**:
* Kết nối với chân D0 của ESP32.
* Có bốn chân:
  + Chân 1 được cấp nguồn 3.3V.
  + Chân 2 là GND.
  + Chân 4 (D0) kết nối với ESP32 để đọc dữ liệu ánh sáng.
* **Cảm biến áp suất (JP3)**:
* Sử dụng giao thức I2C để giao tiếp với ESP32.
* Chân SCL của cảm biến được nối với chân D22 (SCL) của ESP32.
* Chân SDA của cảm biến được nối với chân D21 (SDA) của ESP32.
* Cảm biến nhận nguồn 3.3V từ ESP32 và GND được nối với đất.
* **Relay (JP5)**:
* Được điều khiển bởi ESP32 qua chân D23 (IN).
* Chân relay IN được nối với chân D23 của ESP32.
* Cấp nguồn 3.3V cho chân 2 của JP5 và chân 3 là GND.
* **Cảm biến DHT11 (JP4)**:
* Đây là cảm biến nhiệt độ và độ ẩm.
* Chân Data của cảm biến được nối với chân D19 của ESP32.
* Chân 1 nhận nguồn 3.3V và chân 3 là GND.
* Có một điện trở 5kΩ (R1) được kết nối giữa chân Data và nguồn 3.3V để giữ mức điện áp ổn định.
* **Nguồn cấp**:
* ESP32 được cấp nguồn 3.3V từ chân VIN và GND được nối đất.

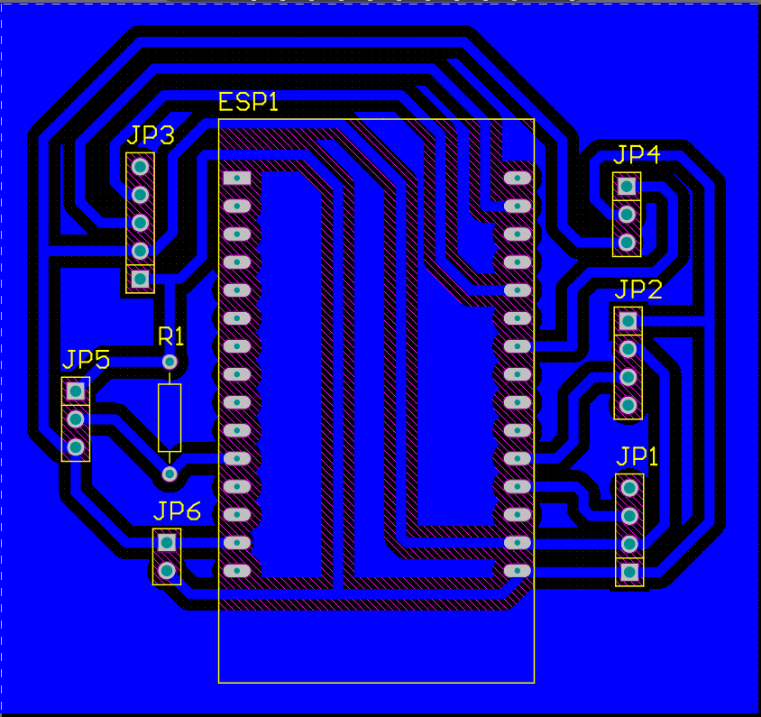
Qua những phân tích trên ta thiết kế được mạch nguyên lý cơ bản của hệ thống.



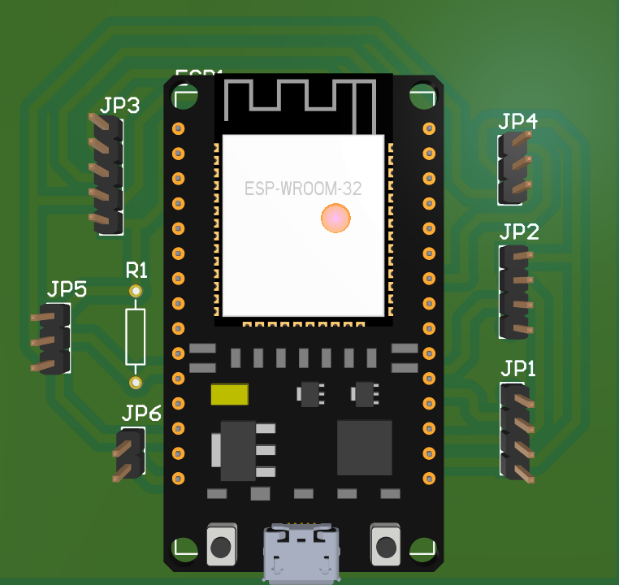
**Hình 4.2 Sơ đồ nguyên lí mạch giám sát thời tiết**

***4.1.3 Nối dây cho mạch***

Sau khi vẽ sơ đồ nguyên lí cho mạch điện ta tiến hành vẽ sơ đồ nối dây cho mạch:



**Hình 4.3** **Giao diện layout linh kiện trên phần mềm altium**



**Hình 4.4 Hình ảnh 3D của mạch điện**

**4.2 Mạch in thủ công**

* Chuẩn bị nguyên liệu:
* Mạch in bằng giấy nhiệt
* Phíp đồng
* Giấy dáp
* Bột sắt
* Cồn rửa, bông lau, dao cắt phíp đồng, thước kẻ, búi cọ nồi
* Bàn là
* Các bước tiến hành:

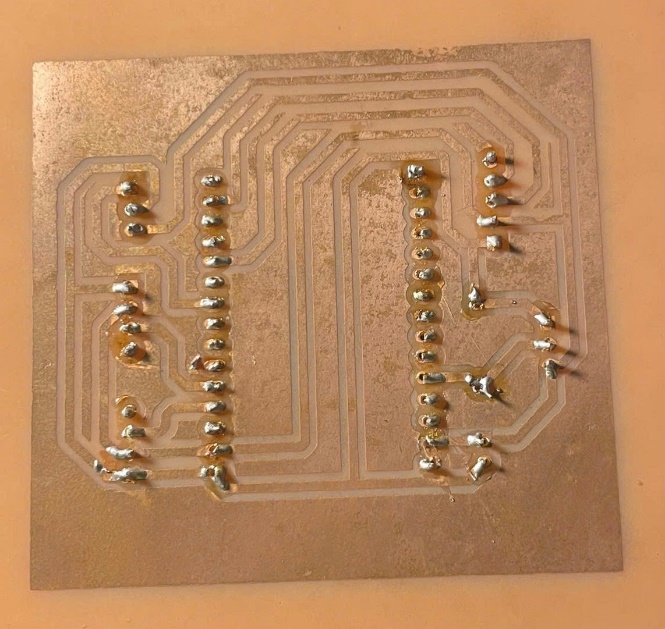
Bước 1: In mạch vào giấy in chuyên dụng, đo kích thước board đồng vừa với mạch in. Đánh sạch mặt miếng phíp đồng bằng giấy ráp để mất hết lớp oxit đồng bám trên bề mặt. Sau đó úp giấy thủ công lên miếng phíp sao cho vừa khít, không để lệch đường mạch in ra ngoài.

Bước 2: Là mạch bằng bàn là đặt nhiệt độ bàn là ở mức trung bình, tránh để nhiệt độ quá cao làm hỏng miếng phíp đồng. Sau khoảng 15 phút bóc nhẹ nhàng giấy thủ công ra để lại các đường mạch in màu đen trên miếng phíp, tô lại các đường mạch in bị đứt do là chưa kĩ bằng bút chết. Sau đó, ngâm miếng phíp (khoảng 30 phút) trong Bột sắt FeCl3 đã được pha vào nước với tỉ lệ vừa đủ (nếu muốn tốc độ nhanh các bạn có thể pha bằng nước nóng), lắc đều tay và những phần đồng không được bảo vệ bằng mực in sẽ bị ăn mòn hết.

Bước 3: Rửa bằng nước sạch sau đó dùng giấy ráp đánh nhẹ nhàng cho bay hêt lớp mực in chỉ để lại những đường mạch bằng đồng.

Bước 4: Bước cuối cùng là khoan các lỗ chân linh kiện. Sau khi khoan xong, cần rửa sạch mạch và bôi lên đó một ít nhựa thông đã được pha loãng bằng axeton để bảo vệ mạch tránh bị oxi hóa.

Kết quả thu được



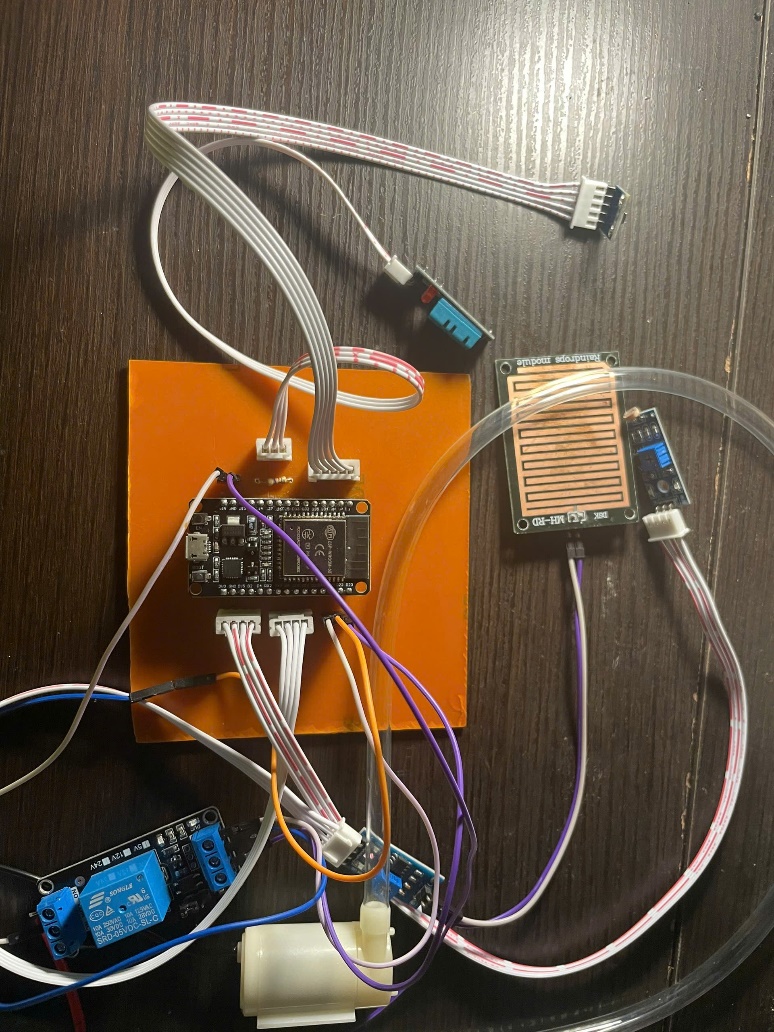
**Hình 4.5 Mạch in thủ công**

**4.3 Gắn linh kiện**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên linh kiện** | **Số lượng** |
| 1 | NodeMCU ESP32 | 1 |
| 2 | Moldule cảm biến áp suất | 1 |
| 3 | Moldule cảm biến ánh sáng | 1 |
| 4 | DHT-11 | 1 |
| 5 | Module cảm biến mưa | 1 |
| 6 | Jack DC | 1 |
| 7 | Relay 1 kênh | 1 |
| 8 | Máy bơm | 1 |
| 9 | Trở 5k | 1 |
| 10 | Jump 3 | 2 |
| 11 | Jump 4 | 2 |
| 12 | Jump 5 | 1 |

**Bảng 4.1 Thống kê linh kiện**

Mạch in sau khi gắn linh kiện



**Hình 4.6** **Mạch điện hoàn chỉnh sau khi hiệu chỉnh**

**4.4 Lập trình hệ thống**

***4.4.1 Giới thiệu phần mềm Arduino***

IDE (Integrated Development Environment) nguyên văn là môi trường tích hợp dùng để phát triển phần mềm, nó cũng tương tự như những phần mềm bình thường khác nhưng mục đích của IDE là dùng để viết mã nguồn.

IDE không chỉ đơn giản là dùng để viết mã nguồn không mà bản thân nó còn kèm theo các công cụ hỗ trợ khác như trình biên dịch (Compiler), trình thông dịch (Interpreter), công cụ kiểm tra lỗi (Debugger), định dạng hoặc highlight mã nguồn, tổ chức thư mục chứa mã nguồn, tìm kiếm mã nguồn, ...v.v.

Arduino là một nền tảng nguyên mẫu (mã nguồn mở) dựa trên nền phần mềm và phần cứng dễ sử dụng. Nó bao gồm một bo mạch - thứ mà có thể được lập trình (đang đề cập đến vi điều khiển) và một phần mềm hỗ trợ gọi là Arduino IDE (Môi trường phát triển tích hợp cho Arduino), được sử dụng để viết và nạp từ mã máy tính sang bo mạch vật lý.

Bạn có thể điều khiển các chức năng của bo mạch của mình bằng cách nạp các tập lệnh đến vi điều khiển trên bo mạch. Thông qua phần mềm hỗ trợ là Arduino IDE.

Hơn nữa, phần mềm Arduino IDE sử dụng phiên bản giản thể của C++, làm việc học lập trình nó trở nên dễ dàng hơn rất nhiều.

* **3.1.7.1 Cài đặt Arduino IDE**

Bước 1: Dowload trực tiếp phần mềm Arduino tại trang chủ

**Link:** [**https://www.arduino.cc/en/Main/Software**](https://www.arduino.cc/en/Main/Software)

Bước 2: Cài đặt và chạy file vừa tải xuống

A screenshot of a software license

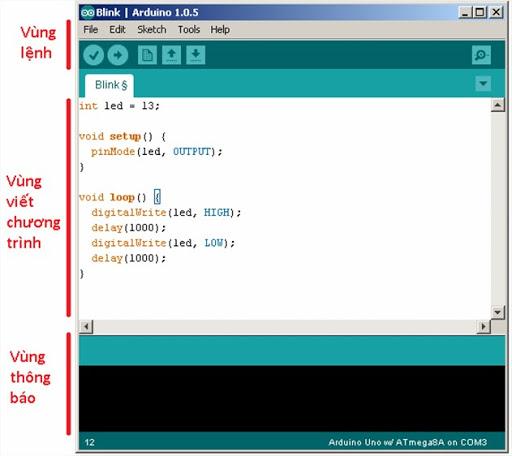
Description automatically generated

A screenshot of a computer

Description automatically generated

**Hình 4.7 Hướng dẫn cài đặt Arduino IDE bằng hình ảnh**

* **Giao diện Arduino IDE**



**Hình 4.8** **Giao diện Arduino IDE**

**Vùng lệnh**: Bao gồm các lệnh trên menu ( File, Edit, Sketch, Tools, Help). Phía dưới là các icon cho phép sử dụng nhanh các chức năng thường dùng của IDE.

Chức năng của các icon:

|  |  |  |
| --- | --- | --- |
|  | Verify | Kiểm tra lỗi và biên dịch code |
|  | Upload | Dịch và upload code vào bo mạch đã được cài đặt sẵn |
|  | New | Tạo sketch mới |
|  | Open | Mở một sketch có sẵn |
|  | Save | Lưu sketch |
|  | Serial Monitor | Mở serial monitor. |

**Vùng viết chương trình:**

* Thiết lập (Void Setup): Phần này dùng để thiết lập cho một chương trình Arduino IDE. Phần này dùng để thiết lập các tốc độ truyền dữ liệu, kiểu chân là chân ra hay chân vào…
* Vòng lặp (Void loop ): Dùng để viết các lệnh trong chương trình để mạch Arduino thực hiện các nhiệm vụ mà chúng ta mong muốn.

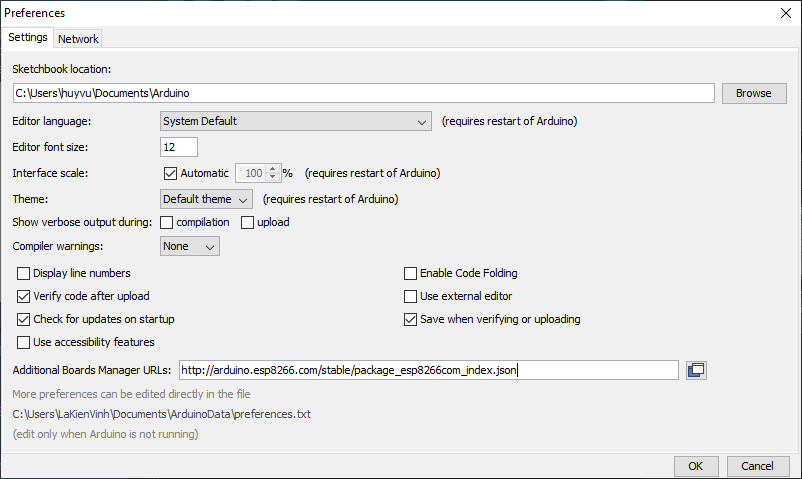
**Vùng thông báo (Debug):** Hiển thị những thông báo từ IDE.

* **Cài đặt cấu hình cho Arduino IDE:**

Bước 1: Tải package cho NodeMCU vào Arduino IDE:

Từ màn hình chính chọn File → Preferences, thêm đường dẫn bên dưới vào mục Addition Boards Manager URLs.

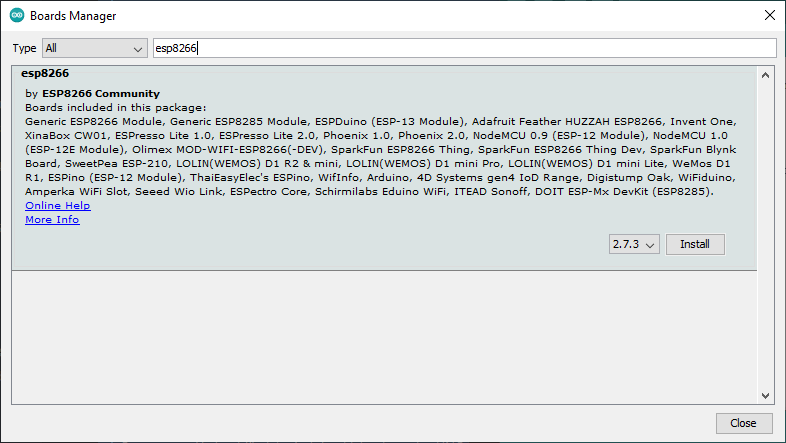
Link: http://arduino.esp8266.com/stable/package\_esp8266com\_index.json



**Hình 4.9 Tải package cho NodeMCU**

Bước 2: Tải thư viện hỗ trợ.

Từ màn hình chính chọn Tool→ Board→ Board managers. Tại thanh tìm kiếm của hộp thoại Board Managers nhập vào esp8266, chọn Install để tiến hành tải và cài đặt thư viện.



**Hình 4.10** **Hộp thoại Board Manager**

Bước 3: Chọn Tool→  board → ESP 8266 boards →  NodeMCU 0.9

A screenshot of a computer

Description automatically generated

**Hình 4.11 Chọn board**

Bước 4: Vào menu chọn → Tool →port →chọn cổng COM để kết nối với máy tính

A screenshot of a computer

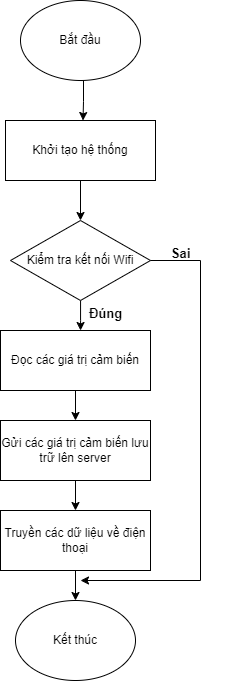
Description automatically generated

**Hình 4.12 Chọn cổng kết nối**

Hoàn tất cài đặt.

**4.4.2 Lưu đồ giải thuật**

**4.4.2.1. Lưu đồ thuật toán chương trình chính**

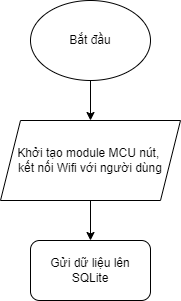


**Hình 4.13 Lưu đồ thuật toán chương trình chính**

**Giải thích nguyên lý hoạt động của chương trình:**

Chương trình bắt đầu bằng cách khởi tạo hệ thống, bao gồm kết nối WiFi. Sau khi kết nối WiFi thành công, các giá trị cảm biến được đọc và gửi lên server thông qua giao thức MQTT. Dữ liệu cảm biến sau đó được truyền tiếp đến điện thoại để hiển thị hoặc xử lý. Nếu kết nối WiFi thất bại, chương trình sẽ lặp lại quá trình kiểm tra kết nối cho đến khi thành công rồi tiếp tục quy trình. Kết thúc chương trình sau khi hoàn thành các bước truyền dữ liệu.

**4.4.2.3 Lưu đồ gửi dữ liệu**



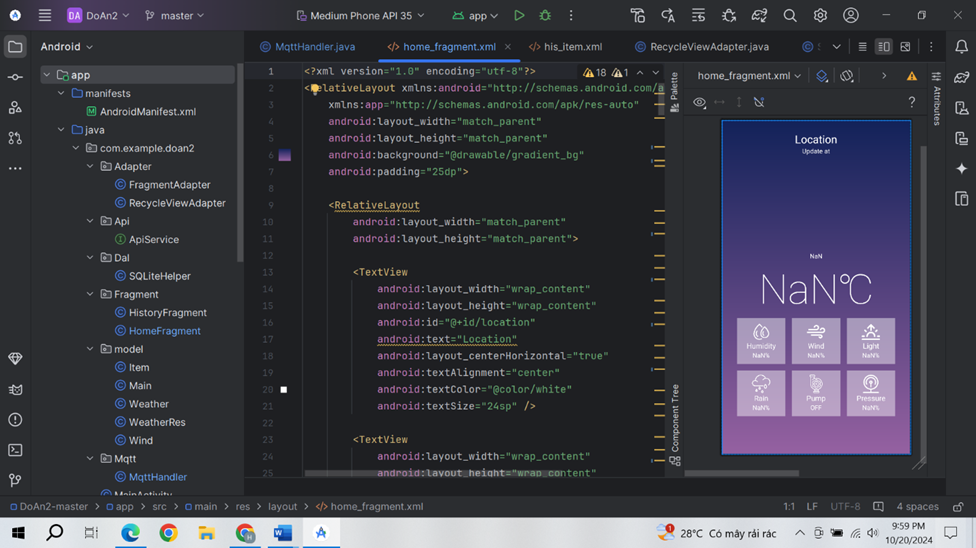
**Hình 4.14 Lưu đồ gửi dữ liệu**

**Giải thích nguyên lý hoạt động của chương trình**

Chương trình bắt đầu bằng việc khởi tạo module MCU và nút điều khiển, sau đó kết nối WiFi với người dùng để thiết lập liên lạc. Khi kết nối thành công, dữ liệu được thu thập từ các nút hoặc cảm biến sẽ được gửi lên cơ sở dữ liệu SQLite để lưu trữ.

***4.4.3 Giới thiệu về*** ***Android Studio***

Android Studio là môi trường phát triển tích hợp (IDE - Integrated Development Environment) chính thức được Google phát triển cho ứng dụng Android. Được xây dựng dựa trên IntelliJ IDEA, Android Studio cung cấp các công cụ và tính năng mạnh mẽ giúp lập trình viên phát triển, thử nghiệm và triển khai ứng dụng Android một cách hiệu quả.



**Hình 4.15 Giao diện Android Studio**

Dưới đây là một số chức năng chính của Android Studio:

* **Trình biên dịch mã:** Android Studio hỗ trợ biên dịch mã nguồn với ngôn ngữ lập trình Java, Kotlin và C++. Nó cung cấp tính năng kiểm tra cú pháp, hoàn thành mã tự động, và gợi ý mã giúp lập trình viên tiết kiệm thời gian và tránh lỗi.
* **Trình giả lập Android** (Android Emulator): Giúp mô phỏng các thiết bị Android khác nhau ngay trên máy tính để thử nghiệm ứng dụng mà không cần thiết bị vật lý. Người dùng có thể chọn nhiều cấu hình thiết bị với các phiên bản hệ điều hành khác nhau.
* **Hệ thống quản lý build dựa trên Gradle**: Android Studio tích hợp hệ thống build mạnh mẽ dựa trên Gradle, giúp tự động hóa quá trình biên dịch và xây dựng ứng dụng, đồng thời quản lý các thư viện và phụ thuộc dễ dàng.
* **Android Layout Editor:** Trình chỉnh sửa giao diện kéo thả giúp lập trình viên dễ dàng xây dựng giao diện người dùng (UI) trực quan mà không cần phải viết mã XML thủ công.
* **Trình quản lý gói SDK:** Hỗ trợ việc cài đặt và cập nhật các phiên bản Android SDK, công cụ build, và các thư viện liên quan.
* **Trình kiểm tra và gỡ lỗi:** Android Studio cung cấp công cụ kiểm tra và gỡ lỗi mạnh mẽ với khả năng theo dõi sự kiện, kiểm tra bộ nhớ, phân tích hiệu suất ứng dụng, và logcat để xem chi tiết log trong quá trình chạy ứng dụng.
* **Tích hợp Git và các hệ thống kiểm soát phiên bản:** Android Studio hỗ trợ tích hợp Git, GitHub, và các hệ thống kiểm soát phiên bản khác, giúp quản lý dự án hiệu quả và cộng tác dễ dàng hơn trong nhóm phát triển.
* **Tối ưu hóa ứng dụng:** Công cụ phân tích APK (APK Analyzer) giúp kiểm tra và tối ưu hóa ứng dụng để giảm kích thước APK và cải thiện hiệu suất.  
   Nhược điểm của app Inventor là:

Với các tính năng đa dạng và mạnh mẽ, Android Studio là công cụ lý tưởng cho lập trình viên phát triển ứng dụng Android từ những bước khởi đầu đến giai đoạn triển khai.

A screenshot of a weather app

Description automatically generatedKết quả thu được



**Hình 4.16 Giao diện chính của phần mềm**

**4.5 Chức năng chính của ứng dụng**

Giới thiệu ứng dụng:

Ứng dụng này được xây dựng trên nền tảng Android Studio, kết hợp với giao thức **MQTT** để nhận và điều khiển các thông số môi trường như nhiệt độ, độ ẩm , ánh sáng , cũng như cảnh báo mưa và điều khiển bơm nước từ xa . Dữ liệu sau đó được lưu trữ trong cơ sở dữ liệu **SQLite** trên thiết bị để có thể xem lại lịch sử hoạt động một cách dễ dàng.

**Các chức năng chính**

1. **Nhận và hiển thị dữ liệu môi trường qua MQTT**:
   * Ứng dụng nhận dữ liệu từ các cảm biến môi trường như nhiệt độ, độ ẩm, ánh sáng, gió và áp suất qua giao thức MQTT. Dữ liệu được cập nhật liên tục và hiển thị trên màn hình chính.
   * Trên giao diện **HomeFragment** (hình 1), các thông số về nhiệt độ, độ ẩm, ánh sáng và các yếu tố khác được hiển thị trong các ô rõ ràng để người dùng theo dõi. Nếu có sự cố (ví dụ không nhận được dữ liệu từ cảm biến), các giá trị sẽ hiển thị là "NaN" (Not a Number) để cảnh báo người dùng về lỗi kết nối.
2. **Cảnh báo mưa và điều khiển bơm nước**:
   * Ứng dụng có khả năng nhận cảnh báo mưa từ cảm biến và hiển thị thông báo trên giao diện. Khi phát hiện có mưa, người dung có thể thủ công điều khiển bơm nước thông qua app điện thoại .
   * Người dùng có thể kiểm soát trạng thái bơm nước bằng cách bật/tắt bơm trực tiếp từ giao diện ứng dụng. Tùy chọn này giúp người dùng linh hoạt trong việc quản lý tưới tiêu dựa trên điều kiện thời tiết.
3. **Lưu trữ dữ liệu vào SQLite**:
   * Mỗi lần nhận dữ liệu từ các cảm biến qua MQTT, ứng dụng sẽ tự động lưu các thông tin như nhiệt độ, độ ẩm, trạng thái bơm và thời gian vào cơ sở dữ liệu SQLite cục bộ trên thiết bị.
   * Giao diện **HistoryFragment** (hình 2) hiển thị lịch sử các thông tin được lưu trữ, bao gồm thời gian, nhiệt độ, độ ẩm và trạng thái bơm, giúp người dùng dễ dàng xem lại dữ liệu đã lưu.
4. **Lịch sử trạng thái và điều khiển**:
   * Người dùng có thể xem lại lịch sử hoạt động của hệ thống trong phần "History" của ứng dụng. Mỗi mục trong danh sách lịch sử bao gồm ngày giờ, giá trị nhiệt độ, độ ẩm và trạng thái của bơm nước.
   * Giao diện trực quan này giúp người dùng dễ dàng quản lý và theo dõi sự thay đổi của các thông số môi trường qua thời gian.

**4.6 Giao thức MQTT và vai trò trong ứng dụng**

**MQTT** (Message Queuing Telemetry Transport) là giao thức truyền thông nhẹ, hiệu quả cho các thiết bị IoT (Internet of Things). Trong ứng dụng này, MQTT đóng vai trò:

* **Nhận dữ liệu cảm biến**: Các cảm biến được kết nối với ESP8266/ESP32 hoặc các thiết bị IoT khác gửi dữ liệu thông qua MQTT broker (máy chủ trung gian) tới ứng dụng.
* **Điều khiển bơm từ xa**: Người dùng có thể gửi lệnh bật/tắt bơm từ giao diện ứng dụng, thông qua MQTT broker, tới thiết bị điều khiển (ESP32 hoặc các bộ điều khiển khác).

**Lưu trữ cục bộ với SQLite**

Việc lưu trữ dữ liệu môi trường và trạng thái hệ thống trong SQLite có các lợi ích sau:

* **Dữ liệu không bị mất khi không có kết nối mạng**: Người dùng vẫn có thể xem lại lịch sử ngay cả khi không kết nối internet.
* **Quản lý lịch sử**: Các dữ liệu lưu trữ theo thời gian giúp phân tích xu hướng môi trường và hiệu quả của hệ thống điều khiển.

**4.7 Các luồng hoạt động của ứng dụng**

**Luồng hoạt động trong HomeFragment**

**1. Khởi tạo giao diện và quyền truy cập vị trí:**

* Trong phương thức onViewCreated(), ứng dụng kiểm tra quyền truy cập vị trí bằng cách sử dụng ContextCompat.checkSelfPermission(). Nếu chưa được cấp quyền, ứng dụng yêu cầu quyền **ACCESS\_FINE\_LOCATION** từ người dùng. Nếu quyền đã được cấp, ứng dụng sẽ tiến hành thiết lập thời gian và vị trí hiện tại.

**2. Thiết lập thời gian hiện tại:**

* Phương thức setTime() được gọi để hiển thị thời gian hiện tại trên ứng dụng và cập nhật mỗi giây bằng cách sử dụng Handler và Runnable. Chu kỳ cập nhật này lặp lại mỗi giây để luôn hiển thị thời gian chính xác.

**3. Lấy vị trí GPS:**

* Ứng dụng sử dụng LocationManager để lấy thông tin vị trí từ GPS (LocationManager.GPS\_PROVIDER). Phương thức request Location Updates() được sử dụng để yêu cầu cập nhật vị trí theo thời gian thực, với khoảng cách tối thiểu và thời gian giữa các lần cập nhật.
* Khi vị trí thay đổi, phương thức onLocationChanged() sẽ được gọi, cập nhật kinh độ (lon) và vĩ độ (lat).

**4. Gọi API thời tiết:**

* Phương thức callApi() gọi API từ ApiService để lấy dữ liệu thời tiết dựa trên tọa độ vị trí. API này trả về các thông số thời tiết như tốc độ gió, áp suất không khí và trạng thái thời tiết. Nếu thành công, các thông số sẽ được hiển thị trên giao diện.

**5. Kết nối với MQTT Broker:**

* Phương thức connectMQTT() được gọi để kết nối tới **MQTT broker** (ở đây là broker.emqx.io) với địa chỉ **BROKER\_URL** và **CLIENT\_ID** xác định.
* Ứng dụng đăng ký lắng nghe các chủ đề MQTT như "esp32/led" và "esp32/relay". Khi có tin nhắn MQTT đến từ các chủ đề này, dữ liệu cảm biến sẽ được nhận và hiển thị lên giao diện.
* Nút điều khiển bơm (pumpIcon) được thiết lập để gửi lệnh bật/tắt bơm qua chủ đề "esp32/relay". Dựa trên trạng thái hiện tại của bơm (ON/OFF), ứng dụng sẽ gửi tín hiệu MQTT để điều khiển bơm, đồng thời cập nhật trạng thái của bơm trên giao diện.

**6. Lưu trữ dữ liệu vào SQLite:**

* Khi người dùng nhấn vào pumpIcon để thay đổi trạng thái bơm, ứng dụng sẽ lấy thông tin hiện tại về nhiệt độ, độ ẩm, vị trí và trạng thái bơm. Thông tin này được lưu vào cơ sở dữ liệu **SQLite** sử dụng đối tượng SQLiteHelper thông qua phương thức addItem().

**7. Nhận và hiển thị dữ liệu cảm biến qua MQTT:**

* Khi có dữ liệu MQTT mới đến (ví dụ: nhiệt độ, độ ẩm, mưa, ánh sáng), ứng dụng xử lý thông qua phương thức onMessageReceived() của lớp MqttHandler. Dữ liệu sẽ được hiển thị trực tiếp lên giao diện:
  + Nhiệt độ và độ ẩm được cập nhật vào temperatureTv và humidityTv.
  + Trạng thái mưa và ánh sáng được hiển thị trong rainTv và lightTv tương ứng.

**8. Dừng hoạt động khi kết thúc:**

* Trong phương thức onDestroy(), các tài nguyên như handler, mqttHandler và executorService được giải phóng để đảm bảo ứng dụng hoạt động hiệu quả và không gây rò rỉ bộ nhớ.

**4.8 Cơ sở dữ liệu SQLite**

Quản lý cơ sở dữ liệu SQLite trong ứng dụng Android

Để lưu trữ dữ liệu cục bộ trong ứng dụng Android, tôi đã sử dụng SQLite, một cơ sở dữ liệu quan hệ nhẹ và tích hợp sẵn trong Android. Trong ứng dụng này, tôi đã triển khai lớp SQLiteHelper để quản lý việc tạo và truy xuất dữ liệu. Dưới đây là chi tiết về cấu trúc và chức năng của cơ sở dữ liệu.

**4.8.1 Cấu trúc cơ sở dữ liệu**

Cơ sở dữ liệu của ứng dụng có tên là Data.db, bao gồm một bảng dữ liệu tên là data. Cấu trúc của bảng này bao gồm các cột sau:

• id: Số nguyên, tự động tăng (PRIMARY KEY) để định danh duy nhất mỗi bản ghi.

• time: Chuỗi ký tự (TEXT) lưu thời gian ghi nhận dữ liệu.

• location: Chuỗi ký tự (TEXT) lưu vị trí địa lý hoặc tên của địa điểm.

• temperature: Số nguyên (INTEGER) lưu thông tin nhiệt độ.

• humidity: Số nguyên (INTEGER) lưu thông tin độ ẩm.

• pump: Số nguyên (INTEGER) lưu trạng thái của máy bơm (bật/tắt).

Câu lệnh SQL để tạo bảng này là :

CREATE TABLE data (

id INTEGER PRIMARY KEY AUTOINCREMENT,

time TEXT,

location TEXT,

temperature INTEGER,

humidity INTEGER,

pump INTEGER

);

**4.8.2 Chức năng của lớp SQLiteHelper**

Lớp SQLiteHelper mở rộng từ lớp SQLiteOpenHelper của Android và cung cấp các phương thức để tương tác với cơ sở dữ liệu:

**•** onCreate(SQLiteDatabase db): Phương thức này được gọi khi cơ sở dữ liệu được tạo lần đầu. Nó thực hiện câu lệnh SQL để tạo bảng data.

• onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion): Phương thức này được dùng để nâng cấp cấu trúc cơ sở dữ liệu khi có phiên bản mới, tuy nhiên trong trường hợp này, chưa có thay đổi nào.

• getAll(): Truy vấn toàn bộ dữ liệu trong bảng data, sắp xếp theo thời gian (cột time) giảm dần, và trả về một danh sách các đối tượng Item. Phương thức này sử dụng con trỏ (Cursor) để di chuyển qua các bản ghi trong bảng.

• addItem(Item i): Thêm một bản ghi mới vào bảng data. Phương thức này nhận đối tượng Item và chèn các giá trị tương ứng vào cơ sở dữ liệu.

• delete(Item i): Xóa một bản ghi khỏi bảng dựa trên id của đối tượng Item.

• deleteAll(): Xóa toàn bộ dữ liệu trong bảng data.

**4.8.3 Tính linh hoạt và khả năng mở rộng**

Với lớp SQLiteHelper, việc quản lý dữ liệu cục bộ trong ứng dụng trở nên dễ dàng và có khả năng mở rộng. Người dùng có thể lưu trữ, truy xuất và xóa dữ liệu một cách hiệu quả mà không cần kết nối internet, phù hợp với các ứng dụng cần lưu trữ dữ liệu tạm thời hoặc không yêu cầu kết nối từ xa.

**4.9 Kết luận chương 4**

Chương này giới thiệu Phần mềm mô phỏng Altium ,phần mềm Arduino, thiết kế app android studio code, cơ sở dữ liệu **SQLite**.Chương cũng đưa ra các lưu đồ thuật toán của hệ thống: Lưu đồ thuật toán chương trình chính, Lưu đồ thuật toán, Lưu đồ gửi dữ liệu.

**Chương V: KẾT QUẢ**

**5.1 Kết quả thu được**

***5.1.1 Sử dụng module ESP32 Node MCU***

ESP32 là một mạch vi điều khiển có thể giúp chúng ta điều khiển các thiết bị điện tử. Điều đặc biệt của nó, đó là sự kết hợp của module Wifi tích hợp sẵn bên trong con vi điều khiển chính. Hiện nay, ESP32 rất được giới nghiên cứu tự động hóa Việt Nam ưa chuộng vì giá thành cực kỳ rẻ (chỉ bằng một con Arduino Nano), nhưng lại được tích hợp sẵn Wifi, bộ nhớ flash 8Mb.

***5.1.2 Sử dụng cảm biến***

Trong quá quá trình thực hiện đề tài, bản thân đã có thêm được nhiều kiến thức về các loại cảm biến, nguyên lí hoạt động cũng như ứng dụng của nó trong thực tế.

***5.1.3 Lập trình app android studio code***

Ứng dụng đã hoàn thành tốt các chức năng chính, đảm bảo khả năng giám sát môi trường và điều khiển từ xa thông qua giao thức MQTT một cách ổn định và hiệu quả. Các cảm biến như nhiệt độ, độ ẩm, ánh sáng, và áp suất cung cấp dữ liệu theo thời gian thực, giúp người dùng theo dõi điều kiện môi trường liên tục. Chức năng cảnh báo mưa và điều khiển bơm nước thủ công từ ứng dụng điện thoại mang lại sự linh hoạt, giúp người dùng quản lý hệ thống tưới tiêu thông minh dựa trên tình hình thời tiết thực tế.

Việc lưu trữ dữ liệu vào cơ sở dữ liệu **SQLite** cục bộ giúp người dùng dễ dàng truy cập và theo dõi lại các thông tin đã thu thập, với giao diện **HistoryFragment** trực quan và rõ ràng. Tính năng này cung cấp khả năng quản lý tốt hơn, không chỉ hiện tại mà còn cả quá trình phát triển môi trường trong quá khứ, hỗ trợ cho việc ra quyết định và điều khiển tối ưu.

Tổng thể, ứng dụng không chỉ đáp ứng yêu cầu theo dõi dữ liệu môi trường mà còn tăng cường tính linh hoạt và khả năng kiểm soát thông qua thiết kế giao diện thân thiện, khả năng lưu trữ và hiển thị lịch sử thông tin. Điều này giúp cải thiện trải nghiệm người dùng và tối ưu hóa hiệu quả quản lý môi trường trong các hoạt động tưới tiêu hoặc các ứng dụng liên quan khác.

**5.2 Kết quả thực nghiệm**

Trong quá trình nghiên cứu, thiết kế và thi công, kết quả của đề tài đáp ứng được các yêu cầu đặt ra



**Hình 5.1 Sản phầm sau khi hoàn thành**



**Hình 5.2 Lịch sử lưu trong App**

 Thông tin dữ liệu sau khi được thu thập từ các cảm biến sẽ được gửi lên màn hình chính của mạch, đồng thời gửi lên app



**Hình 5.3 Thông tin được gửi lên app**

Thông tin gửi lên app gồm có nhiệt độ độ ẩm, trạng thái mưa, áp suất, trạng thái của máy bơm. Ngoài ra, phần mềm còn có thế cho phép người dùng cài đặt thời gian tự động bất máy bơm thông qua phần set time. Nếu như nhiệt độ thực tế thấp hơn nhiệt độ cài đặt và độ ẩm thực tế cao hơn độ ẩm cài đặt thì máy bơm sẽ không hoạt động.

**Chương VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**6.1. Giới thiệu chương.**

Chương này sẽ trình bày về kết luận và hướng phát triển của đề tài Thiết kế , xây dựng hệ thống giám sát thời tiết sử dụng iot.

**6.2. Kết luận.**

Hệ thống giám sát thời tiết là một giải pháp công nghệ giúp thu thập, theo dõi và phân tích các dữ liệu môi trường, cung cấp khả năng cảnh báo sớm khi có biến động thời tiết. Ứng dụng điện thoại thông minh giúp người dùng dễ dàng tiếp cận và quản lý dữ liệu , theo dõi trực quan và nâng cao nhận thức về biến đổi thời tiết giúp giảm thiểu rủi ro và chủ động bảo vệ môi trường sống .

Sau một thời gian nghiên cứu và hoàn thành đề tài, chúng em đã nhận thấy mô hình đã hoạt động tương đối ổn định, trong thời gian nghiên cứu và thực hiện đề tài, chúng em đã học hỏi và tìm hiểu thêm được nhiều kiến thức mới cũng như củng cố lại kiến thức đã học giúp hoàn thành đề tài này. Vì đây là đề tài hướng đến việc giúp cho những người nông dân giảm bớt thiệt hại do thiên tai gây ra đến nông nghiệp, trong quá trình canh tác nông nghiệp nên phải chú trọng độ ổn định và chính xác dẫn đến nhiều khó khăn trong quá trình lập trình. Nhưng nhờ sự hướng dẫn của giảng viên hướng dẫn và các tài liệu tham khảo thì chúng em cũng phần nào giải quyết được tương đối yêu cầu của đề tài.

Tuy rằng sản phẩm đã được hoàn thành nhưng chúng em vẫn nhận thấy sản phẩm còn nhiều thiếu sót, cần được chỉnh sửa và cải tiến hơn.

Xây dựng hoàn thành mô hình Hệ thống giám sát thời tiết sử dụng công nghệ IOT.

Đề tài “Thiết kế , xây dựng hệ thống giám sát thời tiết sự dụng iot” với mục đích giám sát theo dõi các thông số của thời tiết đồng thời cảnh báo đến người sử dụng những tình huống xấu sắp diễn ra nhằm chủ động phòng tránh, giúp người nông dân tăng năng suất và chất lượng cây trồng. Kết quả thực nghiệm đã đáp ứng được những yêu cầu đề ra. Dưới đây là một số ưu-nhược điểm của hệ thống:

**Ưu điểm:**

* Giải quyết được những yêu cầu đề ra của đề tài
* Mô hình đơn giản dễ sử dụng
* Chi phí thấp
* Tiết kiệm thời gian, chi phí nhân công

Tuy nhiên, đây là mô hình nên sự thiếu sót là điều không thể tránh khỏi

**Nhược điểm:**

* Không gian thu thập dữ liệu của cảm biến còn hạn chế
* Độ sai số của các cảm biến còn lớn
* Mô hình chưa gọn
* Thông tin dữ liệu chưa được ổn định

**6.3. Hướng phát triển đề tài**

Đề tài cơ bản đáp ứng được những yêu cầu đặt ra tuy nhiên để sản phẩm hoàn thiện được hơn nữa thì đòi hỏi cần được cải tiến và nghiên cứu thêm.

Hệ thống cần được chỉnh sửa để hoàn chỉnh hơn, dưới đây là những vấn đề được đề ra nhằm hoàn thiện hệ thống hơn.

* Thiết kế giao diện app sao cho dễ nhìn và dễ tiếp cận người dùng
* Tích hợp nhiều cảm biến để thông tin dữ liệu được chính xác hơn
* Ngoài thiết bị bơm và hệ thống chiếu sáng có thể tích hợp thêm hệ thống lưới che và quạt có thể kết hợp thêm một máy bơm phun sương.
* Ứng dụng trong tưới tiêu thủy canh- Mô tả: Kết hợp hệ thống quan trắc thời tiết với cảm biến độ ẩm đất để tự động điều chỉnh lượng nước tưới cho cây trồng. Dữ liệu thời tiết sẽ được sử dụng để dự đoán nhu cầu nước trong từng giai đoạn sinh trưởng của cây.
* Dự báo sâu bệnh và dịch hại- Mô tả: Phát triển hệ thống dự báo dựa trên dữ liệu thời tiết và điều kiện sinh thái để phát hiện khi nào có thể xảy ra sâu bệnh hay dịch hại. Hệ thống sẽ cung cấp thông tin cảnh báo sớm để nông dân có biện pháp xử lý kịp thời.
* Phát triển mô hình với quy mô lớn hơn
* Hoàn thiện mô hình và tối ưu một cách kinh tế hơn.
* Có thể dùng  năng lượng từ pin mặt trời để tạo ra nguồn điện

**TÀI LIỆU THAM KHẢO**

[1] Phạm Quang Huy, Nguyễn Cảnh Trung, *Lập Trình Điều Khiển Với Esp8266*, NXB Khoa Học và kĩ Thuật.

[2] <https://www.alldatasheet.com/>

[3]https://srituhobby.com/iot-based-weather-monitoring-system-using-nodemcu-and-blynk/

[4] T. Wooley, “*A Comparative Study of the Android and iphone Operating Systems*”, University of Central Florida, 2010.

[5] <https://jst.tnu.edu.vn/>

[6] [https:// MySQL.google.com/](https://firebase.google.com/)

[7]youtube.com

**PHỤ LỤC**

**Lập trình hệ thống**

#include <DHT22.h>

#include <WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

#include <DHT.h>

#include <Wire.h>

#include <Adafruit\_BMP085.h>

#define camBienMua 4

#define camBienAnhSang 2

#define RELAY\_PIN 18

// WiFi

const char \*ssid = "VNA";

const char \*password = "20042003";

// MQTT Broker

const char \*mqtt\_broker = "broker.emqx.io";

const char \*topic = "esp32/led";

//const char \*mqtt\_relay\_topic = "esp32/relay";

const char \*mqtt\_username = "emqx";

const char \*mqtt\_password = "public";

const int mqtt\_port = 1883;

// DHT sensor setup

#define DHTPIN 14

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

int relay = 0;

// BMP180 sensor setup

Adafruit\_BMP085 bmp;

float seaLevelPressure = 101325.0;

WiFiClient espClient;

PubSubClient client(espClient);

unsigned long lastPublishTime = 0;

const long publishInterval = 2000;

void reconnectMQTT() {

  while (!client.connected()) {

    String client\_id = "esp32-client-";

    client\_id += String(WiFi.macAddress());

    Serial.printf("The client %s connects to the public MQTT broker\n", client\_id.c\_str());

    if (client.connect(client\_id.c\_str(), mqtt\_username, mqtt\_password)) {

      Serial.println("Public EMQX MQTT broker connected");

      client.subscribe(topic);

    } else {

      Serial.print("Failed with state ");

      Serial.print(client.state());

      delay(2000);

    }

  }

}

void reconnectWiFi() {

  if (WiFi.status() != WL\_CONNECTED) {

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL\_CONNECTED) {

      delay(500);

      Serial.println("Reconnecting to WiFi...");

    }

    Serial.println("Reconnected to the WiFi network");

  }

}

void setup() {

  Serial.begin(9600);

  delay(1000);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.println("Connecting to WiFi...");

  }

  Serial.println("Connected to the WiFi network");

  pinMode(camBienAnhSang, INPUT);

  pinMode(camBienMua, INPUT);

  pinMode(RELAY\_PIN, OUTPUT);

  digitalWrite(RELAY\_PIN, LOW);

  dht.begin();

  if (!bmp.begin()) {

    Serial.println("Không tìm thấy cảm biến BMP180, vui lòng kiểm tra kết nối!");

    while (1) {}

  }

  Serial.println("Cảm biến BMP180 đã sẵn sàng.");

  client.setServer(mqtt\_broker, mqtt\_port);

  client.setCallback(callback);

  reconnectMQTT();

}

void callback(char \*topic, byte \*payload, unsigned int length) {

  Serial.print("Message arrived in topic: ");

  Serial.println(topic);

  Serial.print("Message: ");

  String message;

  for (int i = 0; i < length; i++) {

    message += (char)payload[i];

  }

  Serial.println(message);

  //Serial.print("?????");

  if (message == "1" && !relay) {

    //digitalWrite(RELAY\_PIN, HIGH);

    Serial.println("Relay ON");

    relay = 1;

  } else if (message == "0" && relay) {

    //digitalWrite(RELAY\_PIN, LOW);

    Serial.println("Relay OFF");

    relay = 0;

  }

}

void loop() {

  reconnectWiFi();

  if (!client.connected()) {

    reconnectMQTT();

  }

  client.loop();

  unsigned long currentTime = millis();

  if(relay == 0){

    digitalWrite(RELAY\_PIN, LOW);

  }else{

    digitalWrite(RELAY\_PIN, HIGH);

  }

  if (currentTime - lastPublishTime >= publishInterval) {

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    float f = dht.readTemperature(true);

    if (isnan(h) || isnan(t) || isnan(f)) {

      Serial.println("Failed to read from DHT sensor!");

    }

    int mua = digitalRead(camBienMua);

    int anhsang = digitalRead(camBienAnhSang);

    float temperatureBMP = bmp.readTemperature();

    float pressure = bmp.readPressure();

    float adjustedPressure = pressure \* (seaLevelPressure / 101325.0);

    Serial.print("Humidity: ");

    Serial.print(h);

    Serial.print(" %\t");

    Serial.print("Temperature: ");

    Serial.print(t);

    Serial.print(" \*C ");

    Serial.print(f);

    Serial.print(" \*F\t");

    Serial.print("  Heat index: ");

    float hic = dht.computeHeatIndex(t, h, false);

    Serial.print(hic);

    Serial.print(" \*C ");

    Serial.println();

    Serial.print("Mưa: ");

    Serial.println(mua);

    Serial.print("Ánh sáng: ");

    Serial.println(anhsang);

    Serial.print("Nhiệt độ BMP180: ");

    Serial.print(temperatureBMP);

    Serial.println(" °C");

    Serial.print("Áp suất: ");

    Serial.print(adjustedPressure);

    Serial.println(" Pa");

    DynamicJsonDocument data(1024);

    data["humidity"] = h;

    data["temperature"] = t;

    data["rain"] = mua;

    data["light"] = anhsang;

    data["bmp\_temperature"] = temperatureBMP;

    data["pressure"] = adjustedPressure;

    data["pump"] = relay;

    char json\_string[1024];

    serializeJson(data, json\_string);

    Serial.println(json\_string);

    client.publish(topic, json\_string, false);

    lastPublishTime = currentTime;

  }

}

**Thư viện sử dụng:**

[env:esp32doit-devkit-v1]

platform = espressif32

board = esp32doit-devkit-v1

framework = arduino

lib\_deps =

    bblanchon/ArduinoJson@^7.2.0

    adafruit/DHT sensor library@^1.4.6

    mobizt/FirebaseJson@^3.0.9

    knolleary/PubSubClient@^2.8

    adafruit/Adafruit BMP085 Library@^1.2.4

    markruys/DHT@^1.0.0

    winlinvip/SimpleDHT@^1.0.15

    beegee-tokyo/DHT sensor library for ESPx@^1.19