|  |  |
| --- | --- |
|  | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI** |
| PHẠM VĂN TRÍ | **---------------------------------------**  ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC  NGÀNH CNKT ĐIỆN TỬ VIỄN THÔNG  **SỬ DỤNG CẢM BIẾN QUÉT MÃ QR, BĂNG TẢI SERVO ĐỂ ĐIỀU KHIỂN VÀ PHÂN LOẠI** |
| **TS. Trần Đình Thông**  **Sinh viên: Phạm Văn Trí**  **Mã số sinh viên: 2018605310** |
| NGÀNH CNKT ĐIỆN TVIỄTHÔNG THÔNG | Hà Nội 2022 |

# LỜI CẢM ƠN

Lời đầu tiên cho em xin trân thành cảm ơn TS. Trần Đình Thông, người đã hết lòng chỉ dẫn, truyền đạt những kiến thức chuyên môn cũng như những kinh nghiệm liên quan cho chúng em trong suốt quá trình thực hiện đồ án này.

Xin chân thành cảm ơn đến tất cả quý thầy, cô nhà trường nói chung và các thầy cô bộ môn khoa điện tử nói riêng của trường Đại Học Công Nghiệp Hà Nội đã giảng dạy, trang bị cho em những kiến thức rất bổ ích và quý báu trong suốt quá trình học tập để em có thể áp dụng nghiên cứu hoàn thành đề tài này.

Trong quá trình thực hiện đồ án, với điều kiện thời gian cũng như kinh nghiệm, kiến thức còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được những ý kiến đóng góp của thầy, cô và các bạn để em có thêm được nhiều kinh nghiệm vào công việc trong tương lai.

Cuối cùng, chúng em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên chúng em trong suốt quá trình học tập và hoàn thành đồ án này.

Em xin chân thành cảm ơn!

Hà Nội, ngày…tháng…năm 2022.

Sinh viên thực hiện đề tài

Phạm Văn Trí

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc103362202)

[MỤC LỤC 2](#_Toc103362203)

[DANH MỤC HÌNH ẢNH VÀ BẢNG BIỂU 5](#_Toc103362204)

[LỜI MỞ ĐẦU 7](#_Toc103362205)

[CHƯƠNG 1: TỔNG QUAN VỀ MÔ HÌNH THU THẬP DỮ LIỆU NHIỆT ĐỘ VÀ ĐỘ ẨM 8](#_Toc103362206)

[1.1 Tổng quan về đề tài 8](#_Toc103362207)

[1.1.1 Lý do lựa chọn đề tài 8](#_Toc103362208)

[1.1.2 Đặt vấn đề nghiên cứu 8](#_Toc103362209)

[1.1.3 Ý nghĩa khoa học và thực tiễn của đề tài 14](#_Toc103362210)

[1.2 Các giao tiếp được sử dụng 14](#_Toc103362211)

[1.2.1 Giao tiếp I2C 14](#_Toc103362212)

[1.2.2 Giao tiếp One – Wire 20](#_Toc103362213)

[1.3 Tổng quan về Blynk App 22](#_Toc103362214)

[1.4 Các phần mềm được sử dụng 24](#_Toc103362215)

[1.5 Kết luận chương 1 25](#_Toc103362216)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VỀ MÔ HÌNH THU THẬP DỮ LIỆU NHIỆT ĐỘ VÀ ĐỘ ẨM 26](#_Toc103362217)

[2.1 Xác định bài toán cho hệ thống 26](#_Toc103362218)

[2.1.1 Xác định bài toán 26](#_Toc103362219)

[2.1.2 Yêu cầu của bài toán thiết kế hệ thống thu thập nhiệt độ và độ ẩm 26](#_Toc103362220)

[2.1.3 Giải pháp công nghệ 26](#_Toc103362221)

[2.1.4 Giải pháp thiết kế 27](#_Toc103362222)

[2.2 Sơ đồ khối của hệ thống thu thập dữ liệu về nhiệt dộ và độ ẩm 27](#_Toc103362223)

[2.3 Một số linh kiện trong mạch 28](#_Toc103362224)

[2.3.1 Vi điều khiển ESP32 28](#_Toc103362225)

[2.3.2 Cảm biến nhiệt độ độ ẩm DHT22 31](#_Toc103362226)

[2.3.3 Màn hình LCD 16x2 32](#_Toc103362227)

[2.3.4 Module chuyển đổi giao tiếp I2C cho LCD 33](#_Toc103362228)

[2.3.5 IC ổn áp LM7805 34](#_Toc103362229)

[2.4 Kết luận chương 2 34](#_Toc103362230)

[CHƯƠNG 3: THIẾT KẾ MÔ HÌNH CỦA HỆ THỐNG 35](#_Toc103362231)

[3.1 Thiết kế mạch nguyên lý 35](#_Toc103362232)

[3.1.1 Khối nguồn và khối ổn áp 5V 36](#_Toc103362233)

[3.1.2 Khối xử lý trung tâm 37](#_Toc103362234)

[3.1.3 Khối cảm biến 38](#_Toc103362235)

[3.1.4 Khối hiển thị 38](#_Toc103362236)

[3.2 Thiết kế mạch in (PCB) 38](#_Toc103362237)

[3.2.1 Đặt luật cho mạch in 38](#_Toc103362238)

[3.2.2 Sơ đồ mạch in (PCB) 39](#_Toc103362239)

[3.3 Lưu đồ thuật toán 40](#_Toc103362240)

[3.4 Mô hình sản phẩm hoàn thiện 41](#_Toc103362241)

[3.5 Kết quả thực nghiệm 41](#_Toc103362242)

[3.6 Kết luận và hướng phát triển 43](#_Toc103362243)

[3.7 Hướng phát triển của đề tài 43](#_Toc103362244)

[KẾT LUẬN 44](#_Toc103362245)

[TÀI LIỆU THAM KHẢO 45](#_Toc103362246)

[PHỤ LỤC 46](#_Toc103362247)

# DANH MỤC HÌNH ẢNH VÀ BẢNG BIỂU

[Hình 1‑1; Cảm biến Xiaomi Mijia 3](#_Toc103362175)

[Hình 1‑2: Mijia Gateway v3 4](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362176)

[Hình 1‑3: Cảm biến nhiệt độ độ ẩm Tuya Smart 5](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362177)

[Hình 1‑4: Thiết bị giám sát chất lượng không khí Tuya 6](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362178)

[Hình 1‑5: Giao tiếp I2C 8](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362179)

[Hình 1‑6: Dạng message của I2C 10](#_Toc103362180)

[Hình 1‑7: Một Master nhiều Slave 12](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362181)

[Hình 1‑8: Nhiều Master nhiều Slave 13](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362182)

[Hình 1‑9: Giao tiếp One-wire 14](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362183)

[Hình 1‑10: Các thao tác cơ bản để gửi bản tin One – wire 15](#_Toc103362184)

[Hình 1‑11: Các thành phần mềm trong Blynk App 17](#_Toc103362185)

[Hình 2‑1: Sơ đồ khối mô tả giao tiếp giữa esp32 và server blynk trên điện thoại 21](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362186)

[Hình 2‑2: Cảm biến DHT22 25](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362187)

[Hình 2‑3: Sơ đồ chân màn hình LCD16x2 26](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362188)

[Hình 2‑4: LM7805 28](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362189)

[Hình 3‑1: Sơ đồ nguyên lý của hệ thống 29](#_Toc103362190)

[Hình 3‑2: Adapter nguồn 5v 30](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362191)

[Hình 3‑3: Khối nguồn và ổn áp nguồn 31](#_Toc103362192)

[Hình 3‑4: Khối xử lý trung tâm 31](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362193)

[Hình 3‑5: Khối cảm biến 32](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362194)

[Hình 3‑6: Khối hiển thị 32](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362195)

[Hình 3‑7: Sơ đồ mạch in dạng 2d và 3d 33](#_Toc103362196)

[Hình 3‑8: Lưu đồ thuật toán của hệ thống 34](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362197)

[Hình 3‑9: Mô hình sản phẩm hoàn thiện 35](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362198)

[Hình 3‑10: Hình ảnh mạch thực tế khi hoạt động 35](file:///D:\THINH\DO_AN\Project_HaUI_2_2022\Pham_Van_Tri_2018605310\Quyen_bao_Cao\edit_khoa\V2_PVTri_quyen_bao_cao.docx#_Toc103362199)

[Hình 3‑11: Giao diện hiển thị trên máy tính 36](#_Toc103362200)

[Hình 3‑12: Giao diện hiển thị trên điện thoại 36](#_Toc103362201)

# LỜI MỞ ĐẦU

Với sự phát triển của khoa học và công nghệ cùng với sự bùng nổ của cuộc cách mạng công nghiệp lần thứ tư đang diễn ra, các công nghệ hiện đại đang được phát triển và được ứng dụng trong đời sống rất nhiều. Đặc biệt, với sự phát triển của Internet, công nghệ Internet vạn vật (IoT) đang được phát triển một cách nhanh chóng. IoT cho phép các tổ chức có thể kết nối, kiểm soát và sử dụng các thiết bị IoT. Trong hệ sinh thái này, một tổ chức có thể sử dụng các thiết bị như điện thoại thông minh, máy tính bản, … để gửi đi các hiệu lệnh, hoặc truy cập thông tin từ một mạng lưới các thiết bị IoT khác.

Ngoài ra, trong những năm gần đây, công nghệ vi điện tử phát triển rất mạnh mẽ. Sự ra đời của các vi mạch cỡ lớn, cực lớn với giá thành giảm nhanh, khả năng lập trình ngày càng cao đã mang lại những thay đổi sâu sắc trong ngành kĩ thuật điện tử. Nền công nghiệp thế giới đã đạt được những thành tựu to lớn nhờ ứng dụng những tiến bộ của khoa học kĩ thuật và công nghệ, máy móc đã thay thế con người trong nhiều hoạt động lao động sản xuất.

Sự kết hợp giữa vi điều khiển và công nghệ IoT giúp cho máy móc trở nên thông minh hơn, có nhiều tính năng đa dạng và giải quyết được nhiều vấn đề trong thực tế. Để tìm hiểu về IoT và kết hợp được các kiến thức đã học về vi điều khiển, vi xử lý, các môn học về điều khiển và giám sát không dây, em đã quyết định chọn đề tài: “**Thiết kế mô hình thu thập dữ liệu nhiệt độ và độ ẩm hiển thị trên LCD và điện thoại**” là đề tài đồ án tốt nghiệp.

Nội dung quyển báo cáo này bao gồm 3 chương chính:

Chương 1 - Tổng quan về đề tài thu thập dữ liệu nhiệt độ và độ ẩm

Chương 2 - Thiết kế mạch thu thập dữ liệu nhiệt độ và độ ẩm

Chương 3 - Thiết kế mô hình thu thập dữ liệu nhiệt độ và độ ẩm

# TỔNG QUAN VỀ MÔ HÌNH THU THẬP DỮ LIỆU NHIỆT ĐỘ VÀ ĐỘ ẨM

## Tổng quan về đề tài

### Lý do lựa chọn đề tài

Trong ứng dụng hàng ngày, nhu cầu theo dõi nhiệt độ và độ ẩm ngày càng trở nên phổ biến và thiết thực. Theo dõi, giám sát và thu thập nhiệt độ và độ ẩm ngày càng trở nên phổ biến và thiết thực. Hệ thống đo nhiệt độ và độ ẩm thường được sử dụng trong: Sản xuất chế biến nông nghiệp, hiển thị và thực thi (Quạt gió, máy sấy, điều hòa…), lưu trữ và kiểm soát dữ liệu về môi trường tại một khu vực…

Ngày nay, với sự phổ biến của Smartphone và Internet, các sản phẩm về đo nhiệt độ và độ ẩm mang tính thông minh ra đời và mang lại hiệu quả cao đang được chú trọng phát triển. Vì vậy, việc đo và giám sát các thông số như nhiệt độ, độ ẩm kết hợp với các ứng dụng có trên điện thoại giám sát từ xa đang trở nên phổ biến và linh hoạt.

### Đặt vấn đề nghiên cứu

#### Tính cấp thiết của đề tài

Việc giám sát và thu thập dữ liệu đang là công nghệ mới và là chủ đề được mọi người quan tâm để ứng dụng vào cuộc sống. Ngày nay với công nghệ Internet phát triển, việc giám sát và theo dõi từ xa, phân tích các dữ liệu đang được ứng dụng trong rất nhiều các lĩnh vực.

Việc giám sát nhiệt độ và độ ẩm theo cách truyền thống gây khó khăn và mất thời gian ghi chép. Với việc thu thập và giám sát bằng điện thoại và LCD sẽ trở nên đơn giản và dễ dàng theo dõi các trạng thái, các cảnh báo về nhiệt độ và độ ẩm.

Với một chiếc điện thoại có kết nối Internet, công nghệ giúp cho chúng ta có thể giám sát, theo dõi nhiệt độ và độ ẩm ở bất kỳ nơi đâu.

#### Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu đề tài.

* Vi điều khiển ESP32
* Lập trình vi điều khiển ESP32 trên Arduino IDE
* Blynk App và ứng dụng
* Cảm biến đo nhiệt độ và độ ẩm DHT22

Phạm vi nghiên cứu.

* Đề tài tập trung về vấn đề đo nhiệt độ và độ ẩm của môi trường
* Hiển thị giá trị nhiệt độ và độ ẩm lên LCD và giao diện Blynk App trên điện thoại

#### Một số mô hình đã có trên thị trường

*Xiaomi Mijia Smart Home và Xiaomi Home Kit*

Cảm biến Xiaomi Mijia xác định độ ẩm với độ chính xác cao. Cảm biến có thể theo dõi dữ liệu của nhà / văn phòng bằng điện thoại.



Hình 1‑1; Cảm biến Xiaomi Mijia

Nó cũng có thể liên kết với các sản phẩm khác của Xiaomi giúp chúng hoạt động khi đạt đến nhiệt độ hoặc độ ẩm nhất định giúp không khí căn hộ luôn cân bằng và thông minh



Hình 1‑2: Mijia Gateway v3

Hệ thống có chức năng theo dõi nhiệt độ và độ ẩm thời gian thực. Tự động báo động nếu nhiệt độ và độ ẩm bất thường. Dễ dàng xem sự thay đổi nhiệt độ và độ ẩm theo thời gian được lưu lại trong nhật ký. Độ chính xác đo nhiệt độ là ± 0,3 ° C và độ chính xác đo độ ẩm: ± 3%

*Máy đo ghi nhiệt độ, độ ẩm không dây Wifi* *Tuya SHP-V3*

Máy đo ghi nhiệt độ, độ ẩm không dây **Wifi Tuya SHP-LB3v3** được thiết kế màn hình to, rõ ràng, hiển thị thông số nhiệt độ độ ẩm sắc nét, cực kỳ phù hợp cho các ứng dụng theo dõi và giám sát nhiệt độ, độ ẩm tại **kho lạnh**, kho bảo quản thực phẩm tươi sống, kho chứa dược phẩm, tủ bảo quản mẫu, **giám sát nhiệt độ phòng máy chủ**, phòng server, siêu thị…

Cảm biến với đầu dò nhiệt độ độ ẩm không khí cực nhạy được nối dài, liên tục ghi lại sự thay đổi nhiệt độ độ ẩm từ môi trường và so sánh với mức nhiệt ẩm bạn đã cài đặt từ trước khi giá trị vượt quá giới hạn cho phép thiết bị sẽ tự động kêu loa báo động đồng thời cảnh báo về điện thoại qua App Tuya Smart ở bất cứ đâu có wifi, 3G.

Hình 1‑3: Cảm biến nhiệt độ độ ẩm Tuya Smart

Giám sát nhiệt độ độ ẩm cho phòng server, datacenter hay trung tâm dữ liệu vì đây chính là nơi tập trung nhiều thiết bị điện tử hoạt động, tỏa nhiệt rất nhiều, nguy cơ cháy nổ rất cao. Chính vì thế việc giám sát nhiệt độ cũng như độ ẩm của phòng máy chủ cực kỳ quan trọng đảm bảo phòng máy có thể hoạt động ổn định và liên tục hạn chế sự cố do nhiệt độ và độ ẩm cao gây ra.

Sử dụng trong các hệ thống bảo quản thuốc, bảo quản và vận chuyển vắc-xin, hóa chất, dùng trong lĩnh vực công nghệ sinh học, vườn ươm, tạo giống, phòng thí nghiệm…

* Hệ thống giám sát nhiệt độ kho lạnh hiển thị thông số nhiệt độ độ ẩm bằng màn hình led 7 đoạn.
* Cảnh báo tại chỗ khi nhiệt độ độ ẩm không đáp ứng đủ.
* Cảm biến độ chính xác cao cập nhật trạng thái liên tục.

*Thiết Bị Giám Sát Chất Lượng Không Khí Wifi Tuya Treo Tường Air Quality Monitor*

**Thiết Bị Giám Sát Chất Lượng Không Khí Wifi Tuya Treo Tường Air Quality Monitor có tới 3 cảm biến là cảm biến CO2, cảm biến nhiệt độ và cảm biến độ ẩm. Thiết bị liên tục ghi lại sự thay đổi về các chỉ số Co2, nhiệt độ độ ẩm từ môi trường cho phép xem trên điện thoại qua App Tuya Smart ở bất cứ đâu có wifi, 3G.

Hình 1‑4: Thiết bị giám sát chất lượng không khí Tuya

Giám sát chất lượng không khí, nhiệt độ độ ẩm cho văn phòng, trong nhà hay bệnh viện vì đây chính là những nơi mọi người làm việc và sinh hoạt nhiều nên cần quan sát đảm bảo chất lượng không khí để không ảnh hưởng đến vấn đề sức khỏe.

Giám sát các chỉ số cơ bản nhất của không khí với độ chính xác cao. Màn hình lớn hiển thị màu sắc dễ dàng để đọc các kết quả kiểm tra, thuận tiện hơn để sử dụng quan sát chất lượng không khí trong nhà. Dễ dàng để thực hiện, phù hợp cho nhà ở, xe hơi, văn phòng, nhà máy và môi trường khác.

Hiển thị chất lượng không khí để giúp mọi người kiểm soát tình hình sức khỏe tốt hơn. Nó giúp phát hiện các tác nhân có hại HCHO và TVOC xung quanh phòng của bạn trong thời gian thực.

* Hệ thống hiển thị bằng màn hình LCD .
* Sử dụng pin sạc chạy được sau khi mất điện.
* Cảm biến độ chính xác cao cập nhật trạng thái liên tục.
* Phối hợp ngữ cảnh linh hoạt với các thiết bị Tuya khác.
* Dễ dàng chia sẻ cho nhiều điện thoại khác cùng sử dụng thiết bị.

Một số thông số kỹ thuật chính:

* Pin: pin sạc lithium polymer 1200 Mah, cổng Micro USB.
* Màn hình hiển thị LCD có màu.
* Phạm vi đo không khí: CO2: 0-5000ppm
* Thời gian lấy mẫu: 1.5s
* 3 cảm biến: CO2 + Nhiệt Độ + Độ Ẩm
* Khoảng đo nhiệt độ: -10°C - +50°C.
* Lưu trữ nhiệt độ: -10°C - +50°C
* Khoảng đo độ ẩm: 20% - +60%
* Độ chính xác độ ẩm nhiệt độ: 0,5°C, 1%RH.
* Sai số: ± 1°C, ± 3%.
* App Tuya Smart, Smart Life tương thích hệ điều hành IOS và Android.
* Kích thước: 270\*90\*35mm
* Trọng lượng: 700g.

### Ý nghĩa khoa học và thực tiễn của đề tài

Đề tài: **“Thiết kế mô hình thu thập dữ liệu nhiệt độ và độ ẩm hiển thị trên LCD và điện thoại”** tuy chỉ ngừng tại ở phạm vi nhỏ là đo nhiệt độ độ ẩm, hiển thị nhiệt độ độ ẩm trên điện thoại là LCD nhưng đề tài có ý nghĩa to lớn trong thực tế với sự kết hợp của vi điều khiển và các ứng dụng thông minh trên điện thoại .

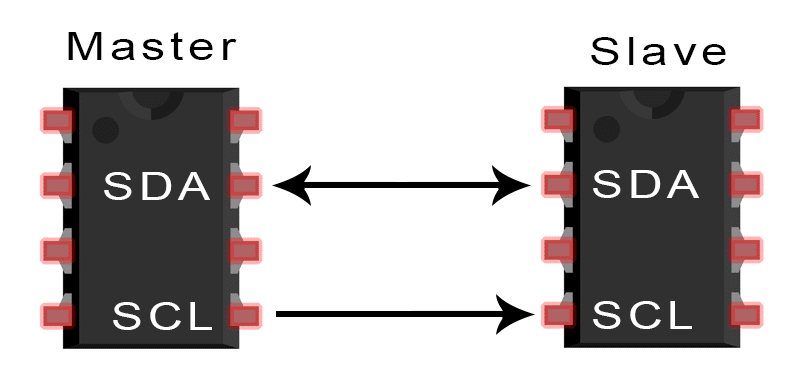
Việc đo nhiệt độ và độ ẩm, thống kê về dữ liệu nhiệt độ và độ ẩm giúp người dùng có cái nhìn trực quan hơn về nhiệt độ trong các phòng cần yêu cầu cao về môi trường làm việc.

Đề tài có nhiều ứng dụng trong thực tế, sinh viên vận dụng những kiến thức tiếp thu được trong quá trình đào tạo tại khoa điện tử và các kiến thức tìm hiểu được bằng cách tự học hỏi.

Đề tài không chỉ dừng lại ở phạm vi nghiên cứu nhất định mà còn có thể mở rộng ra theo rất nhiều hướng khác nhau như: Smarthome, các hệ thống nhà kính, hệ thống tự động …

## Các giao tiếp được sử dụng

### Giao tiếp I2C



Hình 1‑5: Giao tiếp I2C

*Giới thiệu về giao tiếp I2C*

[6] Giao tiếp I2C là sự kết hợp các tính năng tốt nhất của UART và SPI, trong giao tiếp I2C, một thiết bị có thể đóng vai trò là Master (chủ) kết nối tới nhiều Slave (tớ), hoặc có thể nhiều thiết bị đóng vai trò là Master kết nối tới một hoặc nhiều thiết bị là Slave.

I2C chỉ sử dụng 2 dây để kết nối các thiết bị với nhau:

* SDA (tín hiệu Data) - Master và Slave gửi và nhận dữ liệu.
* SCL (xung Clock) - Tín hiệu xung Clock

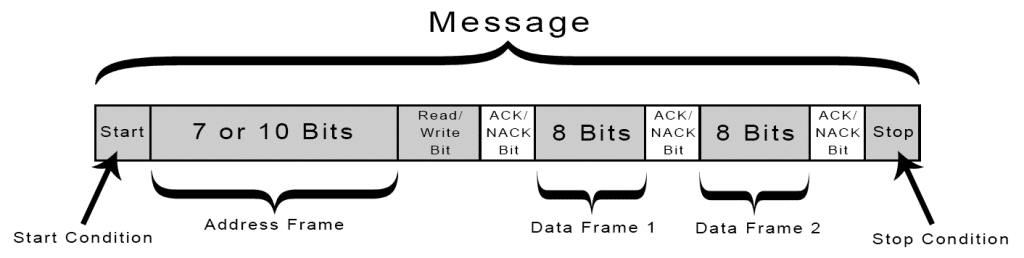
I2C là một giao thức giao tiếp nối tiếp, vì vậy dữ liệu được truyền từng bit theo một dây duy nhất (dây SDA)

Giống như SPI, I2C sử dụng cơ chế đồng bộ do đó, đầu ra của bit được đồng bộ hóa với việc lấy mẫu các bit bằng tín hiệu Clock được chia sẻ giữa Master và Slave. Tín hiệu Clock luôn được kiểm soát bởi Master.

|  |  |
| --- | --- |
| Số lượng dây sử dụng | 2 |
| Tốc độ truyền tối đa | Chế độ tiêu chuẩn = 100 kbps  Chế độ nhanh = 400 kbps  Chế độ tốc độ cao = 3.4 Mbps  Chế độ tốc độ rất cao = 5 Mbps |
| Loại chế độ truyền | Đồng bộ |
| Phương thức truyền | Nối tiếp |
| Số lượng Master tối đa | Không giới hạn về số lượng |
| Số lượng Slave tối đa | 1008 |

*Cách hoạt động của I2C*

Với I2C, dữ liệu được truyền trong các tin nhắn. Tin nhắn được chia thành các khung dữ liệu. Mỗi tin nhắn có một khung địa chỉ chứa địa chỉ nhị phân của địa chỉ slave và một hoặc nhiều khung dữ liệu chứa dữ liệu đang được truyền. Thông điệp cũng bao gồm điều kiện khởi động và điều kiện dừng, các bit đọc / ghi và các bit ACK / NACK giữa mỗi khung dữ liệu:



Hình 1‑6: Dạng message của I2C

* Điều kiện khởi động: Đường SDA chuyển từ mức điện áp cao xuống mức điện áp thấp trước khi đường SCL chuyển từ mức cao xuống mức thấp.
* Điều kiện dừng: Đường SDA chuyển từ mức điện áp thấp sang mức điện áp cao sau khi đường SCL chuyển từ mức thấp lên mức cao.
* Khung địa chỉ: Một chuỗi 7 hoặc 10 bit duy nhất cho mỗi slave để xác định slave khi master muốn giao tiếp với nó.
* Bit đọc / ghi: Một bit duy nhất chỉ định master đang gửi dữ liệu đến slave (mức điện áp thấp) hay yêu cầu dữ liệu từ nó (mức điện áp cao).
* Bit ACK / NACK:Mỗi khung trong một tin nhắn được theo sau bởi một bit xác nhận / không xác nhận. Nếu một khung địa chỉ hoặc khung dữ liệu được nhận thành công, một bit ACK sẽ được trả lại cho thiết bị gửi từ thiết bị nhận.

*Địa chỉ*

I2C không có các đường chọn Slave như SPI, vì vậy cần một cách khác để cho slave biết rằng dữ liệu đang được gửi đến slave này chứ không phải slave khác. Nó thực hiện điều này bằng cách định địa chỉ. Khung địa chỉ luôn là khung đầu tiên sau bit khởi động trong một tin nhắn mới.

Master gửi địa chỉ của slave mà nó muốn giao tiếp với mọi slave được kết nối với nó. Sau đó, mỗi slave sẽ so sánh địa chỉ được gửi từ master với địa chỉ của chính nó. Nếu địa chỉ phù hợp, nó sẽ gửi lại một bit ACK điện áp thấp cho master. Nếu địa chỉ không khớp, slave không làm gì cả và đường SDA vẫn ở mức cao.

*Bit đọc / ghi*

Khung địa chỉ bao gồm một bit duy nhất ở cuối tin nhắn cho slave biết master muốn ghi dữ liệu vào nó hay nhận dữ liệu từ nó. Nếu master muốn gửi dữ liệu đến slave, bit đọc / ghi ở mức điện áp thấp. Nếu master đang yêu cầu dữ liệu từ slave, thì bit ở mức điện áp cao.

*Khung dữ liệu*

Sau khi Master phát hiện bit ACK từ Slave, khung dữ liệu đầu tiên đã sẵn sàng được gửi.

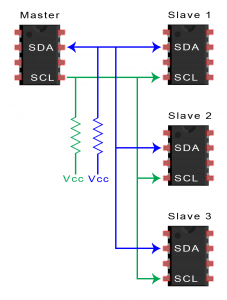
Khung dữ liệu luôn có độ dài 8 bit và được gửi với bit quan trọng nhất trước. Mỗi khung dữ liệu ngay sau đó là một bit ACK / NACK để xác minh rằng khung đã được nhận thành công. Bit ACK phải được nhận bởi Master hoặc Slave (tùy thuộc vào cái nào đang gửi dữ liệu) trước khi khung dữ liệu tiếp theo có thể được gửi.

Sau khi tất cả các khung dữ liệu đã được gửi, Master có thể gửi một điều kiện dừng cho slave để tạm dừng quá trình truyền. Điều kiện dừng là sự chuyển đổi điện áp từ thấp lên cao trên đường SDA sau khi chuyển tiếp từ thấp lên cao trên đường SCL , với đường SCL vẫn ở mức cao.

*Các bước truyền dữ liệu I2C*

1. Master gửi điều kiện khởi động đến mọi slave được kết nối bằng cách chuyển đường SDA từ mức điện áp cao sang mức điện áp thấp trước khi chuyển đường SCL từ mức cao xuống mức thấp.
2. Master gửi cho mỗi slave địa chỉ 7 hoặc 10 bit của slave mà nó muốn giao tiếp, cùng với bit đọc / ghi.
3. Mỗi slave sẽ so sánh địa chỉ được gửi từ master với địa chỉ của chính nó. Nếu địa chỉ trùng khớp, slave sẽ trả về một bit ACK bằng cách kéo dòng SDA xuống thấp cho một bit. Nếu địa chỉ từ master không khớp với địa chỉ của slave, slave rời khỏi đường SDA cao.
4. Master gửi hoặc nhận khung dữ liệu.
5. Sau khi mỗi khung dữ liệu được chuyển, thiết bị nhận trả về một bit ACK khác cho thiết bị gửi để xác nhận đã nhận thành công khung.
6. Để dừng truyền dữ liệu, master gửi điều kiện dừng đến slave bằng cách chuyển đổi mức cao SCL trước khi chuyển mức cao SDA.

***Một Master với nhiều Slave***

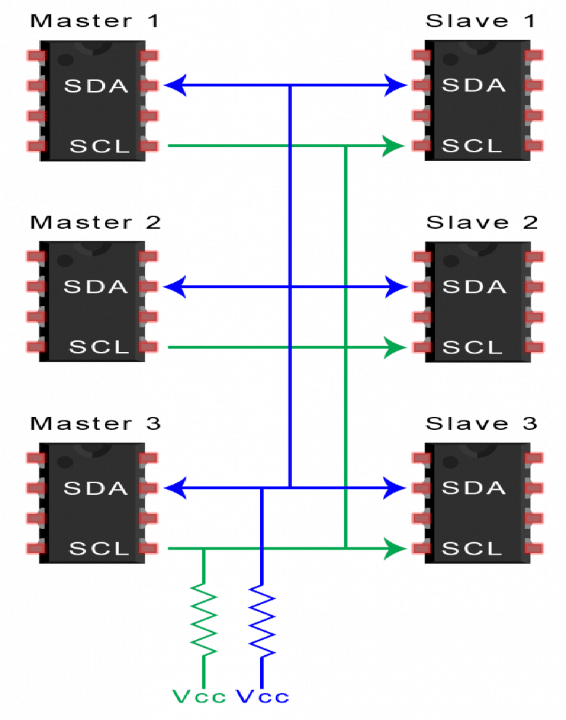
Vì I2C sử dụng định địa chỉ nên nhiều Slave có thể được điều khiển từ một master duy nhất.

Với địa chỉ 7 bit sẽ có 128 (2 mũ 7) địa chỉ duy nhất. Việc sử dụng địa chỉ 10 bit không phổ biến, nhưng nó cung cấp 1.024 (2 mũ 10) địa chỉ duy nhất.

Để kết nối nhiều slave đến một master duy nhất, bạn có thể đấu dây như thế này, với điện trở kéo lên 4.7K Ohm kết nối đường SDA và SCL với Vcc:

Hình 1‑7: Một Master nhiều Slave

*Nhiều master với nhiều slave*

Nhiều master có thể được kết nối với một slave hoặc nhiều slave. Sự cố với nhiều master trong cùng một hệ thống xảy ra khi hai master cố gắng gửi hoặc nhận dữ liệu cùng một lúc qua đường SDA. Để giải quyết vấn đề này, mỗi master cần phải phát hiện xem đường SDA thấp hay cao trước khi truyền tin nhắn. Nếu đường SDA thấp, điều này có nghĩa là một master khác có quyền điều khiển bus và master đó phải đợi để gửi tin nhắn. Để kết nối nhiều master với nhiều slave, hãy sử dụng sơ đồ sau, với các điện trở kéo lên 4,7K Ohm kết nối các đường SDA và SCL với Vcc:

Hình 1‑8: Nhiều Master nhiều Slave

*Ưu điểm*

* Chỉ sử dụng hai dây
* Hỗ trợ nhiều Master và nhiều Slave
* Bit ACK / NACK xác nhận mỗi khung được chuyển thành công
* Phần cứng ít phức tạp hơn so với UART
* Giao thức nổi tiếng và được sử dụng rộng rãi

*Nhược điểm*

* Tốc độ truyền dữ liệu chậm hơn SPI
* Kích thước của khung dữ liệu bị giới hạn ở 8 bit
* Cần phần cứng phức tạp hơn để triển khai so với SPI

### Giao tiếp One – Wire

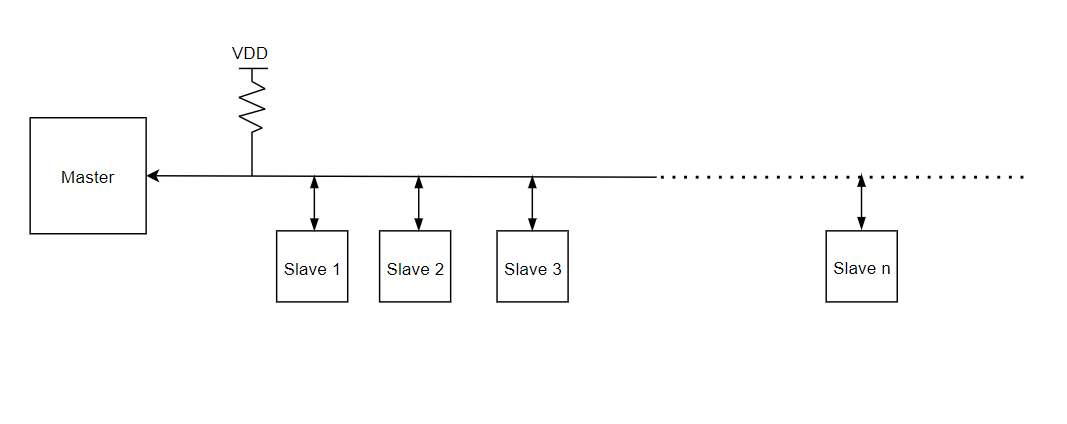
*Giới thiệu về giao tiếp One – Wire (1 – Wire)*

One - Wire (1-Wire) là hệ thống bus giao tiếp được thiết kế bởi Dallas Semiconductor Corp. Giống như tên gọi, hệ thống bus này chỉ sử dụng 1 dây để truyền nhận dữ liệu.

Chính vì chỉ sử dụng 1 dây nên giao tiếp này có tốc độ truyền thấp nhưng dữ liệu lại truyền được khoảng cách xa hơn.

OneWire chủ yếu sử dụng để giao tiếp với các thiết bị nhỏ, thu thập và truyền nhận dữ liệu thời tiết, nhiệt độ,… các công việc không yêu cầu tốc độ cao.

Giống như các chuẩn giao tiếp khác, 1-Wire cho phép truyền nhận dữ liệu với nhiều Slave trên đường truyền. Tuy nhiên chỉ có thể có 1 Master (giống với SPI).



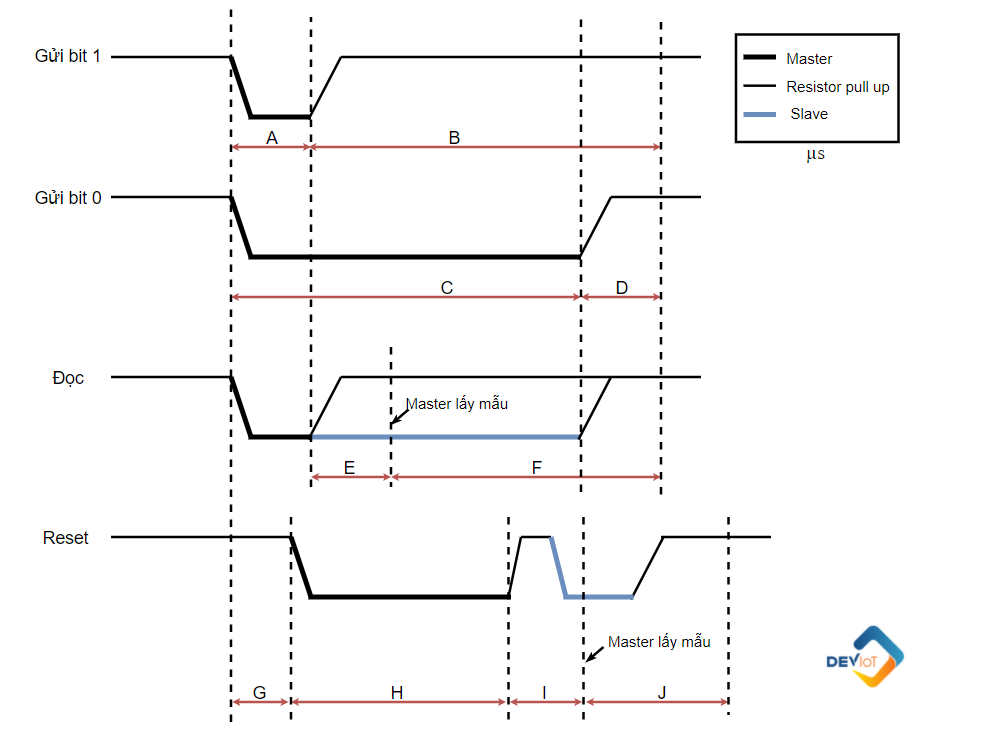
Hình 1‑9: Giao tiếp One-wire

*Cách hoạt động của giao tiếp One – Wire*

So với các chuẩn giao tiếp cơ bản như UART, SPI, I2C mà chúng ta đã biết, cách thức hoạt động của OneWire có hơi “ lạ “ 1 chút.

Như chúng ta thấy ở hình…, đường dây luôn được giữ ở mức cao (High). Các thao tác hoạt động cơ bản của bus sẽ được quy định bởi thời gian kéo đường truyền xuống mức thấp (Low) như hình vẽ dưới.

Có 4 thao tác cơ bản như sau:



Hình 1‑10: Các thao tác cơ bản để gửi bản tin One – wire

Giải thích ý nghĩa :

Gửi bit 1: Khi muốn gửi đi bit 1, thiết bị Master sẽ kéo bus xuống mức 0 trong một khoảng thời gian A (µs) và trở về mức 1 trong khoảng B (µs).

Gửi bit 0: Thiết bị Master kéo bus xuống mức 0 trong một khoảng thời gian C (µs) và trở về mức 1 trong khoảng D (µs).

Đọc bit: Thiết bị Master kéo bus xuống 1 khoảng A (µs). Trong khoảng thời gian E (µs) tiếp theo, thiết bị master sẽ tiến hành lấy mẫu. Có nghĩa trong E (µs) này, nếu bus ở mức 1, thiết bị master sẽ đọc bit 1. Ngược lại, nếu bus ở mức 0 thì master sẽ đọc được bit 0.

Reset: Thiết bị Master kéo bus xuống 1 khoảng thời gian H (µs) và sau đó về mức 1. Khoảng thời gian này gọi là tín hiệu reset. Trong khoảng thời gian I (µs) tiếp theo, thiết bị master tiến hành lấy mẫu. Nếu thiết bị slave gắn với bus gửi về tín hiệu 0, (tức bus ở mức 0), master sẽ hiểu rằng slave vẫn có mặt và quá trình trao đổi dữ liệu lại tiếp tục. Ngược lại nếu slave gửi về tin hiệu 1 ( bus ở mức 1) thì master hiểu rằng không có thiết bị slave nào tồn tại và dừng quá trình.

*Chế độ hoạt động:*

Chế độ Standard (Chế độ tiêu chuẩn)

* 15.4 Kb/s
* 65 µs bit

Chế độ Overdrive (Chế độ tốc độ nhanh)

* 125 Kb/s
* 8 µs bit

## Tổng quan về Blynk App

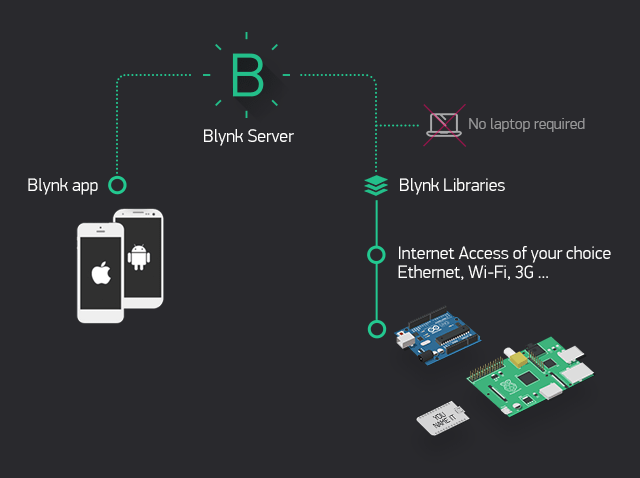
*Blynk là gì?*

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 Web.

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

Blynk không bị ràng buộc với một số bo hoặc defend cụ thể. Thay vào đó, nó hỗ trợ phần cứng mà bạn lựa chọn. Cho dù Arduino hoặc Raspberry Pi của bạn được liên kết với Web qua Wi-Fi, Ethernet hoặc chip ESP8266, Blynk sẽ giúp bạn on-line và sẵn sàng cho IoT.

*Cách hoạt động của Blynk*



Hình 1‑11: Các thành phần mềm trong Blynk App

Có ba thành phần chính trong nền tảng:

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

Blynk Server – chịu trách nhiệm về tất cả các giao tiếp giữa điện thoại thông minh và phần cứng. Bạn 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.

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 bạn 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 bạn.

*Tính năng, đặc điểm*

Cung cấp API & giao diện người dùng tương tự cho tất cả các thiết bị và phần cứng được hỗ trợ

Kết nối với server bằng cách sử dụng:

* Wifi
* Bluetooth và BLE
* Ethernet
* USB (Serial)
* GSM

Các tiện ích trên giao diện :

* Thao tác kéo thả trực tiếp giao diện mà không cần lập trình
* Dễ dàng tích hợp và thêm chức năng mới bằng cách sử dụng các cổng kết nối ảo được tích hợp trên blynk app
* Theo dõi lịch sử dữ liệu
* Thông tin liên lạc từ thiết bị đến thiết bị bằng Widget
* Gửi email, tweet, thông báo realtime

## Các phần mềm được sử dụng

[10] Đề tài này sử dụng 3 phần mềm chính: phần mềm Arduino IDE để biên dịch và nạp chương trình cho vi điều khiển ESP32 và phần mềm Altium Designer để thiết kế mạch nguyên lý và mạch in và phần mềm Blynk App đã được nêu ở trên.

Việc thiết kế mạch điện tử trên phần mềm altium designer có thể được tóm tắt gồm các bước như sau:

* Đặt ra các yêu cầu bài toán.
* Lựa chọn linh kiện.
* Thiết kế mạch nguyên lý.
* Lựa chọn các chân linh kiện để chuyển sang mạch in Update mạch nguyên lý sang mạch in.
* Lựa chọn kích thước mạch in Sắp sếp các vị trí các loại linh kiện như điện trở , tụ điện, IC...
* Đặt kích thước các loại dây nối.
* Đi dây trên mạch.
* Kiểm tra toàn mạch.

## Kết luận chương 1

Để bắt kịp xu thế của công nghệ, đặc biệt là tổng hợp lại kiến thức đã học ở các môn, hơn nữa mục đích của đề tài giúp việc tìm hiểu và phát triển về vi điều khiển ESP32 - một vi điều khiển giúp xây dựng các ứng dụng về IoT (Internet of Things) trở nên dễ dàng và nhanh chóng, tạo ra sản phẩm công nghệ cao, có tính ứng dụng cao, có tính mới.

# CƠ SỞ LÝ THUYẾT VỀ MÔ HÌNH THU THẬP DỮ LIỆU NHIỆT ĐỘ VÀ ĐỘ ẨM

## Xác định bài toán cho hệ thống

### Xác định bài toán

Đề tài tập trung vào nghiên cứu, thiết kế mạch với chức năng thu thập dữ liệu nhiệt độ và độ ẩm của môi trường từ xa cụ thể là qua wifi cụ thể là app blink và hiển thị lên màn hình LCD.

Thiết kế hệ thống thu thập dữ liệu nhiệt độ và độ ẩm này bao gồm 4 khối: khối nguồn và khối ổn áp 5V, khối xử lý trung tâm, màn hình LCD và khối cảm biến.

### Yêu cầu của bài toán thiết kế hệ thống thu thập nhiệt độ và độ ẩm

Để đảm bảo đượccác yêu cầu và lợi ích của người sử dụng, hệ thu thập nhiệt độ và độ ẩm từ xa qua wifi có các yêu cầu sau:

* Nhiệt độ và độ ẩm được cập nhật một cách liên tục.
* Sai số của hệ thống không quá lớn.
* Thiết bị phải cung cấp đủ nguồn tránh xảy ra trường hợp hỏng, cháy các linh kiện trong mạch để đảm bảo an toàn cho người dùng.
* Kết cấu phải đơn giản, dễ sử dụng.

### Giải pháp công nghệ

* Mạch dùng bộ điều khiển trung tâm là esp32.
* Cảm biến phục vụ việc đọc nhiệt độ và độ ẩm là DHT22.
* LCD hiển thị dữ liệu là loại lCD 16 cột 2 hàng.
* Giao diện giám sát là server trên APP Blynk.
* Cảm biến được đọc thông qua chuẩn giao tiếp 1 wire.
* LCD hiển thị dữ liệu qua chuẩn giao tiếp I2C.

### Giải pháp thiết kế

* Công cụ lập trình: Phần mềm Arduino IDE.
* Phần mềm thiết kế mạch: Altium Designer 18.
* Phần mềm Blynk APP

## Sơ đồ khối của hệ thống thu thập dữ liệu về nhiệt dộ và độ ẩm

**Mạch ESP32**

**Request**

**Response**



**Response**

**Request**

**Blynk Server**

Hình 2‑1: Sơ đồ khối mô tả giao tiếp giữa esp32 và server blynk trên điện thoại

Khi đó ESP32 và điện thoại đóng vai trò là Client và Server là Blynk

Điện thoại/máy tính sẽ gửi Request tới Server Blynk, Server Blynk sẽ gửi Response về cho ESP32 sẽ trả về dữ liệu nhiệt độ, độ ẩm

Khối nguồn và ổn áp 5v

Khối xử lý trung tâm (ESP32)

Màn hình LCD

Khối cảm biến

Mạch ESP32 bao gồm 4 khối chính:

* Khối nguồn và ổn áp 5V: mạch được cấp nguồn 12V, sau đó nguồn 12V sẽ được đi qua khối ổn áp về 5V để cung cấp cho vi điều khiển và cảm biến
* Khối xử lý trung tâm: Nhận và xử lý tín hiệu từ cảm biến, hiển thị giá trị ra LCD và gửi tín hiệu lên Server Blynk App
* Màn hình LCD: Hiển thị giá trị nhiệt độ và độ ẩm
* Khối cảm biến: Cảm biến nhiệt độ độ ẩm DHT22

## Một số linh kiện trong mạch

### Vi điều khiển ESP32

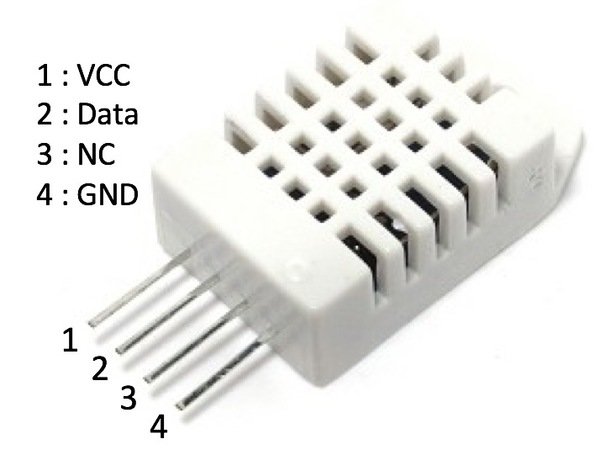
[7] ESP32 là một trong các loại vi điều khiển trên thị trường có giá rẻ, năng lượng thấp có hỗ trợ WiFi và Bluetooth chế độ kép. Dòng ESP32 sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng.

ESP32 được chế tạo và phát triển bởi Espressif Systems, một công ty Trung Quốc có trụ sở tại Thượng Hải, và được sản xuất bởi TSMC bằng cách sử dụng công nghệ 40 nm. ESP32 là sản phẩm kế thừa từ vi điều khiển ESP8266. Các tính năng của ESP32 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) 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)
* *Hệ thống xung nhịp: CPU Clock, RTC Clock và Audio PLL Clock*
* *Bộ nhớ nội:*
* 448 KB bộ nhớ ROM cho việc booting và các tính năng lõi
* 520 KB bộ nhớ SRAM trên chip cho dữ liệu và tập lệnh
* *Kết nối không dây:*
* Wi-Fi: 802.11 b/g/n
* Bluetooth: v4.2 BR/EDR và BLE (chia sẻ sóng vô tuyến với Wi-Fi)
* *34 GPIO vật lý với các ngoại vi*
* ADC SAR 12 bit, 18 kênh
* DAC 2 × 8-bit
* Cảm biến cảm ứng (touch sensor) (GPIO cảm ứng điện dung)
* 3 SPI (SPI, HSPI và VSPI) hoạt động ở cả 2 chế độ master/slave. Module ESP32 hỗ trợ 4 ngoại vi SPI với SPI0 và SPI1 kết nối đến bộ nhớ flash của ESP32 còn SPI2 và SPI3 tương ứng với HSPI và VSPI.
* 2 I²S
* 2 I²C, hoạt động được ở cả chế độ master và slave, với chế độ Standard mode (100 Kbit/s) và Fast mode (400 Kbit/s). Hỗ trợ 2 chế độ định địa chỉ là 7-bit và 10-bits. Các GPIO đều có thể được dùng để triển khai I²C.
* 3 UART (UART0, UART1, UART2) với tốc độ lên đến 5 Mbps
* SD/SDIO/CE-ATA/MMC/eMMC host controller
* SDIO/SPI slave controller
* Ethernet MAC interface cho DMA và IEEE 1588 Precision Time Protocol (tạm dịch: Giao thức thời gian chính xác IEEE 1588)
* CAN bus 2.0
* Bộ điều khiển hồng ngoại từ xa (TX/RX, lên đến 8 kênh)
* PWM cho điều khiển động cơ
* LED PWM (lên đến 16 kênh)
* Cảm biến hiệu ứng Hall
* Bộ tiền khuếch đại analog công suất cực thấp (Ultra low power analog pre-amplifier)
* *Bảo mật:*
* Hỗ trợ tất cả các tính năng bảo mật chuẩn IEEE 802.11, bao gồm WFA, WPA/WPA2 và WAPI.
* Secure boot (tạm dịch: khởi động an toàn)
* Mã hóa flash
* 1024-bit OTP, lên đến 768-bit cho khách hàng
* Tăng tốc mã hóa phần cứng: AES, SHA-2, RSA, elliptic curve cryptography (ECC, tạm dịch: mật mã đường cong ellip), trình tạo số ngẫu nhiên (random number generator)
* *Quản lý năng lượng*
* Bộ ổn áp nội với điện áp rơi thấp (internal low-dropout regulator)
* Miền nguồn riêng (individual power domain) cho RTC
* Dòng 5 μA cho chế độ deep sleep
* Trở lại hoạt động từ ngắt GPIO, timer, đo ADC, ngắt với cảm ứng điện dung

### Cảm biến nhiệt độ độ ẩm DHT22

Hình 2‑2: Cảm biến DHT22



DHT22 (AM2302) là cảm biến độ ẩm và nhiệt độ tương đối đầu ra kỹ thuật số. Nó sử dụng cảm biến độ ẩm điện dung và nhiệt điện trở để đo không khí xung quanh và phun ra tín hiệu kỹ thuật số trên chân dữ liệu

[1] Cảm biến độ ẩm và nhiệt độ DHT22 sử dụng giao tiếp 1 Wire dễ dàng kết nối và giao tiếp với Vi điều khiển để thực hiện các ứng dụng đo nhiệt độ, độ ẩm môi trường, cảm biến có chất lượng tốt, kích thước nhỏ gọn, độ bền và độ ổn định cao.

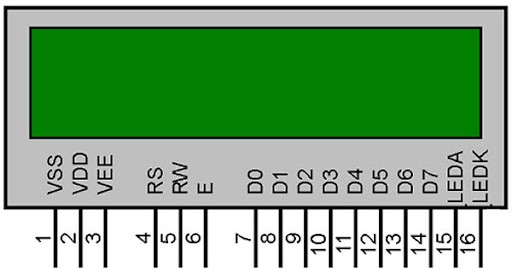
Thông số kỹ thuật:

* Nguồn sử dụng: 3~5VDC.
* Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).
* Đo tốt ở độ ẩm 0100%RH với sai số 2-5%.
* Đo tốt ở nhiệt độ -40 to 80°C sai số ±0.5°C.
* Tần số lấy mẫu tối đa 0.5Hz (2 giây 1 lần)
* Kích thước 27mm x 59mm x 13.5mm (1.05" x 2.32" x 0.53")
* 4 chân, khoảng cách chân 0.1''.

### Màn hình LCD 16x2

* *Giới thiệu:*

[6] LCD có rất nhiều ưu điểm so với các dạng hiển thị khác: Nó có khả năng hiển thị kí tự đa dạng, trực quan dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn ít tài nguyên hệ thống và giá thành rẻ.

* *Cấu tạo:*

Hình 2‑3: Sơ đồ chân màn hình LCD16x2

* LCD 16x2 có 2 hàng, mỗi hàng 16 ký tự, trong 16 chân của LCD được chia làm 3 dạng tín hiệu như sau:
* Các chân cấp nguồn: Chân số 1 nối mass (0V), chân số 2 là VDD nối với nguồn 5V, chân số 3 dùng để chỉnh contrast thường nối với biến trở.
* Các chân điều khiển: Chân số 4 là chân RS dùng để điều khiển lựa chọn thanh ghi. Chân R/W dùng để điều khiển quá trình đọc và ghi. Chân E là chân cho phép dạng xung chốt.
* Các chân dữ liệu DB0 - DB7: Là chân từ số 7 đến 14 dùng để trao đổi dữ liệu giữa thiết bị điều khiển và LCD.
* Chân 15 nối nguồn +5V hoặc 4.2V nối với led, chân 16 nối GND.
* *Ứng dụng:* LCD thường được sử dụng trong các mạch điện tử, hiển thị thời gian thực, giá trị, kết quả, hiệu ứng.

### Module chuyển đổi giao tiếp I2C cho LCD

Module I2C LCD ra đời và giải quyết vấn đề LCD có quá nhiều nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển. [6]

Thay vì phải mất 6 chân vi điều khiển để kết nối với LCD 16x2 (RS, EN, D7, D6, D5 và D4) thì module IC2 bạn chỉ cần tốn 2 chân (SCL, SDA) để kết nối

Module I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 16x2, LCD 20x4, ...) và tương thích với hầu hết các vi điều khiển hiện nay.

Ưu điểm của nó là tiết kiệm chân cho vi điều khiển, dễ dàng kết nối với LCD.

Một số thông số kỹ thuật :

- Điện áp hoạt động: 2.5-6V DC.

- Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780).

- Giao tiếp: I2C.

- Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).

- Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt.

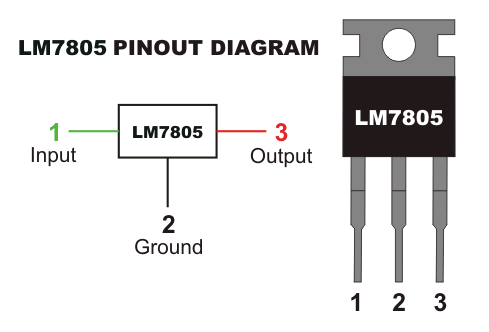
- Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

### IC ổn áp LM7805

[8] LM 7805 là IC điều chỉnh điện áp dương đầu ra 5V. IC này được sử dụng rộng rãi trong các thiết bị thương mại và giáo dục. Nó cũng được sử dụng bởi cho nhiều mạch điện tử thông dụng do giá rẻ, dễ sử dụng và không cần nhiều linh kiện bên ngoài. IC có nhiều tính năng tích hợp lý tưởng để sử dụng trong nhiều ứng dụng điện tử như dòng điện đầu ra 1.5A

Các thông số kỹ thuật của IC LM7805

Hình 2‑4: LM7805



* Dòng điện đầu ra là 1.5A
* Chức năng tắt ngắn mạch
* Chức năng tắt quá nhiệt tức thì
* Đầu ra 5V chính xác và cố định
* Điện áp đầu vào tối đa là 35V DC
* Dòng điện tĩnh thấp chỉ 8mA

## Kết luận chương 2

Chương 2 giúp người đọc xác định được yêu cầu của bài toán, đưa ra giải pháp công nghệ, giải pháp thiết kế, sơ đồ khối tổng quan của hệ thống và các linh kiện phục vụ quá trình thiết kế mạch.

# THIẾT KẾ MÔ HÌNH CỦA HỆ THỐNG

## Thiết kế mạch nguyên lý



Hình 3‑1: Sơ đồ nguyên lý của hệ thống

Mạch nguyên lý bao gồm 4 khối chính:

* Khối nguồn và ổn áp nguồn: Cấp nguồn cho hệ thống hoạt động. Nguồn sử dụng cho hệ thống là 12V
* Khối xử lý trung tâm: Xử lý các tín hiệu từ ngoại vi và gửi tín hiệu lên Server
* Khối cảm biến: Chứa cảm biến nhiệt độ DHT22 có tác dụng biến đổi tín hiệu nhiệt độ và độ ẩm thành tín hiệu điện dạng One – wire để vi xử lý có thể giao tiếp được
* Khối hiển thị: Gồm một màn hình LCD và một module chuyển đổi LCD sang I2C để dễ dàng giao tiếp với vi xử lý

Nguyên lý hoạt động của mạch điện: Khối nguồn sẽ biến đổi trực tiếp điện áp xoay chiều 220V về điện áp một chiều 12V để cấp cho toàn bộ mạch. Do module ESP32 cần cấp nguồn 5V nên cần một khối ổn áp 5v để cấp nguồn 5v cho mạch. Khối cảm biến có nhiệm vụ đọc các tín hiệu về nhiệt độ và độ ẩm và chuyển tín hiệu về chuẩn One-wire để vi xử lý có thể đọc được. Khối Hiển thị bao gồm LCD 16x2 và module I2C giúp LCD có thể giao tiếp với vi điều khiển bằng chuẩn I2C. Khối xử lý trung tâm với vi điều khiển chính là ESP32 có nhiệm vụ biến xử lý các tín hiệu nhận được từ cảm biến, gửi tín hiệu lên Server của Blynk App và gửi ra màn hình LCD

Chi tiết từng khối trong mạch được thể hiện như sau:

### Khối nguồn và khối ổn áp 5V

*Khối nguồn:*

Mạch sử dụng một Adapter nguồn để biến đổi trực tiếp nguồn 220V về 12V để cấp cho toàn bộ hệ thống

Các thông số chính của nguồn:

* Điện áp ngõ ra: 12V
* Điện áp ngõ vào: 100~240VAC
* Đầu cắm AC: 2 chân dẹt Mỹ
* Đầu DC ngõ ra: 5.5x2.5mm
* Dòng điện ngõ ra: 1A
* Tần số hoạt động: 50-60 Hz

Hình 3‑2: Adapter nguồn 5v

* Kích thước jack đầu ra: 5.5 x 2.1mm

*Khối ổn áp nguồn*

**

Hình 3‑3: Khối nguồn và ổn áp nguồn

IC chính được sử dụng để ổn áp mạch 5V là LM7805

* Tụ C3 được thêm vào để lọc nguồn đầu và cải thiện nguồn nếu LM7805 nằm cách bộ lọc của nguồn cung cấp một khoảng cách đáng kể.
* Tụ C2 giúp bộ ổn áp nguồn cải thiện được độ ổn định và phản ứng nhất thời.

### Khối xử lý trung tâm

Khối xử lý trung tâm gồm có 1 vi xử lý chính là ESP32. ESP32 được sử dụng trong mạch ở dạng Kit cắm có tên là ESP32 Devkit V1. Khối xử lý trung tâm – ESP32 xử lý tín hiệu đầu vào cảm biến được đọc vào theo chuẩn One - Wire ở chân D15 (GPIO15) và giao tiếp với màn hình ở các chân D21(SDA) và chân 22(SCL)

Hình 3‑4: Khối xử lý trung tâm

Khối xử lý trung tâm ngoài xử các tín hiệu ở tầng vật lý thì còn gửi và nhận tín hiệu thông qua Server của Blynk App để hiển thị lên điện thoại.

### Khối cảm biến

Chứa cảm biến DHT22 để biến đổi tín hiệu tương tự về tín hiệu One-Wire để ESP32 có thể đọc và tương tác

Hình 3‑5: Khối cảm biến

Các chân kết nối:

* GND: nối GND với vi điều khiển
* VCC5: nối với 5V của nguồn
* DATA\_SS: nối với chân dữ liệu của vi điều khiển

### Khối hiển thị

Sử dụng các header XH2.54 để kết nối tới cảm biến và màn hình:

Kết nối tới màn hình LCD I2C:

Hình 3‑6: Khối hiển thị

* Chân GND: nối GND
* Chân VCC5: nối với VCC5
* Chân SDA, SCL: 2 chân dữ liệu của giao tiếp I2C

## Thiết kế mạch in (PCB)

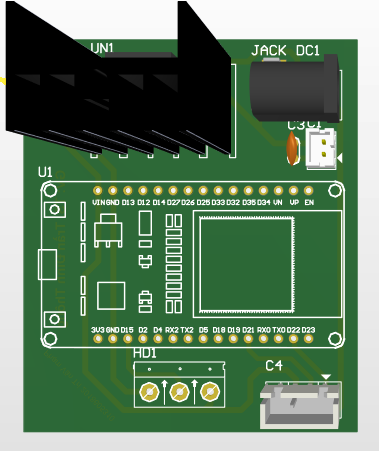
Sử dụng phần mềm Altium Designer 21.2.1 để thiết kế mạch in (PCB) cho đề tài.

### Đặt luật cho mạch in

* Khoảng cách giữa các thành phần: 1mm.
* Độ rộng đường VCC: 1 mm.
* Độ rộng đường GND: 1 mm.
* Độ rộng các đường còn lại: 0.8mm.

### Sơ đồ mạch in (PCB)





Hình 3‑7: Sơ đồ mạch in dạng 2d và 3d

## Lưu đồ thuật toán

* Khởi tạo và khai báo các thư viện sử dụng
* In ra màn hình LCD 5s/1 lần liên tục khi bật hệ thống
* Gửi vào Server của Blynk App để trả về giá trị nhiệt độ và độ ẩm trên điện thoại

Hình 3‑8: Lưu đồ thuật toán của hệ thống

## Mô hình sản phẩm hoàn thiện

Hình 3‑9: Mô hình sản phẩm hoàn thiện

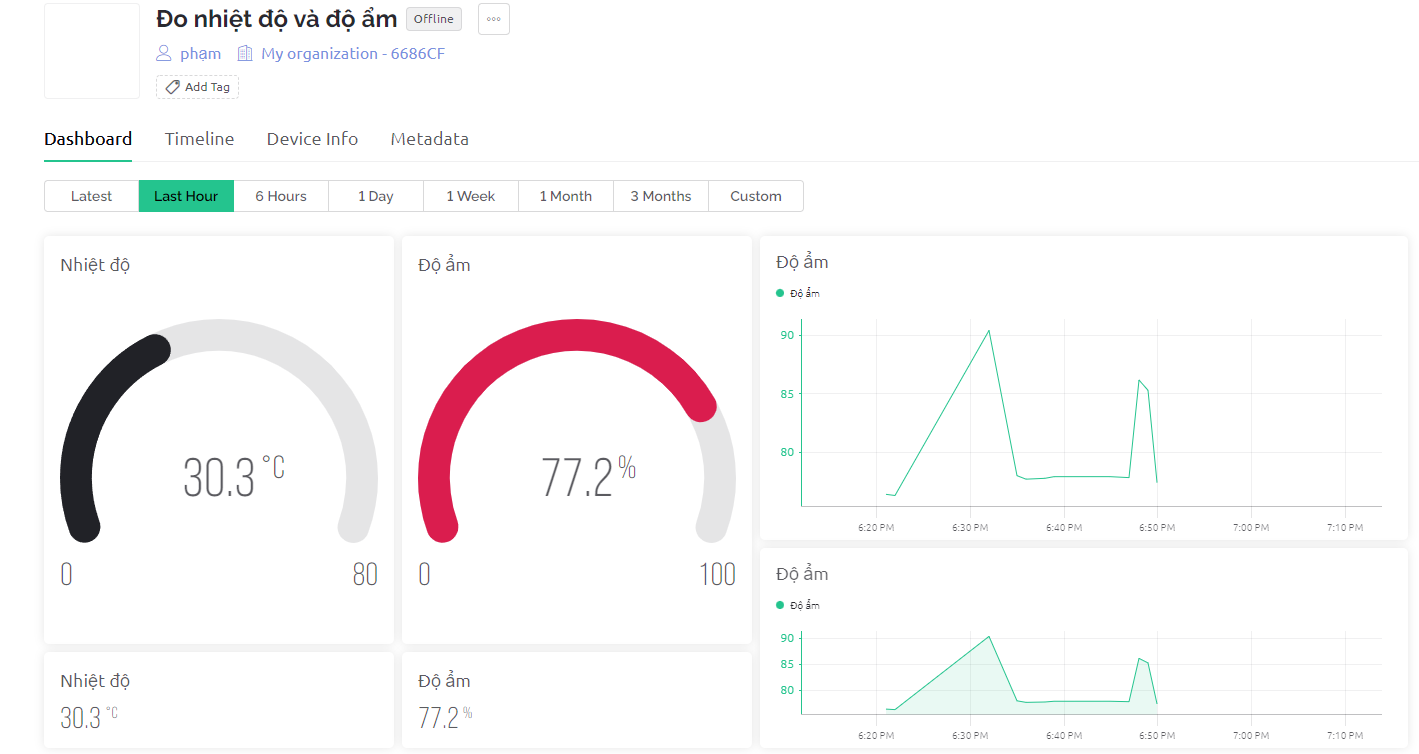


## Kết quả thực nghiệm

Hình ảnh mạch thực tế khi hoạt động

Hình 3‑10: Hình ảnh mạch thực tế khi hoạt động





Hình 3‑11: Giao diện hiển thị trên máy tính



Hình 3‑12: Giao diện hiển thị trên điện thoại

## Kết luận và hướng phát triển

Sau một thời gian nghiên cứu và thực hiện đồ án, em đã đạt được những kết quả như sau:

* Thiết kế thành công mô hình thu thập dữ liệu nhiệt độ và độ ẩm hiển thị trên LCD và điện thoại
* Mạch và giao diện hiển thị trên Blynk App đúng chuẩn yêu cầu kĩ thuật đã đưa ra.
* Mạch thiết kế chạy ổn định cao, đo nhiệt độ và độ ẩm chính xác, thời gian lấy mẫu nhanh
* Mạch đáp ứng được các yêu cầu của người dùng, sử dụng thuận tiện  
  và hữu ích.

## Hướng phát triển của đề tài

* Phát triển đa nền tảng: Đề tài không chỉ giám sát nhiệt độ độ ẩm trên điện thoại mà còn có thể giám sát được trên máy tính, trên các thiết bị khác
* Liên kết các hệ thống lại với nhau để giám sát tập trung
* Phát triển thêm đề tài bằng cách khắc phục những nhược điểm đã nêu ở trên và cập nhật những công nghệ mới, hợp thời đại.
* Xây dựng mô hình tiến bộ hơn để phù hợp với mọi môi trường
* Kết hợp với các hệ thống khác để tăng hiệu quả của đề tài

.

# KẾT LUẬN

Qua quá trình thực hiện làm đồ án, em đã trình bày các cơ sở lý thuyết liên quan và chạy thành công hệ thống “**Thiết kế mô hình thu thập dữ liệu nhiệt độ và độ ẩm hiển thị trên LCD và điện thoại**”. Tuy thời gian làm đồ án thực sự không quá dài nhưng được sự giúp đỡ tận tình của TS. Trần Đình Thông cùng với sự nỗ lực và cố gắng của bản thân, sự chỉ bảo của các Thầy Cô trong khoa Điện tử em đã hoàn thành đề tài theo yêu cầu và đúng thời gian quy định với những nội dung sau:

* Nghiên cứu và tìm hiểu về các hệ thống giám sát nhiệt độ và độ ẩm trên thực tế, ưu điểm và nhược điểm của từng hệ thống
* Nghiên cứu và tìm hiểu về vi điều khiển ESP32, các ngoại vi và lập trình ESP32
* Tìm hiểu về Blynk App giúp việc giám sát nhiệt độ và độ ẩm, giao tiếp với ESP32 dễ dàng và thuận tiện
* Xây dựng được giao diện điều khiển và giám sát trên Blynk App
* Thiết kế và vận hành thành công mạch giám sát nhiệt độ và độ ẩm sử dụng ESP32

Thông qua quá trình làm đồ án, em đã được vận dụng những kiến thức chuyên ngành trong 4 năm học. Qua đó đã giúp cho em rèn luyện được kỹ năng, cách tiếp cận với các vấn đề, các bài toán thực tế phức tạp tại các doanh nghiệp, nhà máy khi ra trường làm việc.

Việc xây dựng mô hình đã đáp ứng được yêu cầu đặt ra, tuy nhiên do trình độ và kinh nghiệm thực tiễn còn hạn chế nên không thể tránh khỏi sai sót và thiếu hoàn chỉnh. Rất mong được đón nhận sự đóng góp ý kiến từ thầy cô và các bạn.

Em xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

[1] Bùi Thị Thu Hà, Ngô Mạnh Tiến, Trương Thị Bích Liên (2019), Giáo trình kỹ thuật cảm biến, KHTN&CN.

[2] Bùi Văn Vinh(2020), Mô đun: Lập trình vi điều khiển, Trường Cao đẳng Kỹ thuật Công Nghệ, Vũng Tàu.

[3] “Altium Designer”, https://www.altium.com/altium-designer

[4] “Visual Studio Code”, https://code.visualstudio.com/

[5] “dientututonglai.com”, https://dientutuonglai.com/chuan-giao-tiep-i2c-la-gi.html

[6] “Kme.com.vn” https://kme.com.vn/blogs/news/tong-quan-lcd1602-va-giao-tiep-i2c-lcd-su-dung-arduino

[7] “wikipedia.org ” https://en.wikipedia.org/wiki/ESP32

[8] “dientutuonglai.com”, https://dientutuonglai.com/mach-on-ap-su-dung-lm7805.html

[9] “arduinokit.vn”, https://arduinokit.vn/giao-tiep-i2c-lcd-arduino/

[10] “arduino.cc”, https://www.arduino.cc/en/software

# PHỤ LỤC

Code ESP32

#define BLYNK\_TEMPLATE\_ID "TMPLkc0eU8vo"

#define BLYNK\_DEVICE\_NAME "Quickstart Template"

#define BLYNK\_AUTH\_TOKEN "T6wIAD-bAbAVR5pOYo9A\_Tj6Gnc5Rao4"

#define BLYNK\_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <DHT.h>

#include <LiquidCrystal\_I2C.h>

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "phamtri";

char pass[] = "12345678";

#define DHTPIN 15

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

BlynkTimer timer;

LiquidCrystal\_I2C lcd(0x27, 16, 2);

float temper = 0.0;

float humi = 0.0;

void sendSensor()

{

Blynk.virtualWrite(V5, humi);

Blynk.virtualWrite(V6, temper);

}

void setup()

{

Serial.begin(115200);

dht.begin();

sendSensor();

timer.setInterval(1000L, sendSensor);

lcd.init();

lcd.backlight();

sendDataToLCDTask();

Blynk.begin(auth, ssid, pass);

}

void loop()

{

timer.run();

Blynk.run();

}

void sendDataToLCDTask()

{

xTaskCreatePinnedToCore(

sendDataToLCD,

" sendDataToLCD "

2048,

NULL,

5,

NULL,

1);

}

void sendDataToLCD(void \*parameter) {

while (1) {

humi = dht.readHumidity();

temper = dht.readTemperature();

lcd.setCursor(0, 0);

lcd.print("Temp: ");

lcd.setCursor(6, 0);

lcd.print(temper);

lcd.setCursor(13, 0);

lcd.write(0xDF);

lcd.print("C");

lcd.setCursor(0,1);

lcd.print("Humi: ");

lcd.setCursor(6, 1);

lcd.print(humi);

lcd.setCursor(13, 1);

lcd.print("%");

vTaskDelay(2000/ portTICK\_PERIOD\_MS);

}

}