

**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**A logo for a university

Description automatically generated**

**ĐỒ ÁN HỆ NHÚNG**

**ĐỀ TÀI:** **THIẾT KẾ HỆ THỐNG MỞ KHÓA CỬA THÔNG MINH & ĐIỀU KHIỂN CÁC THIẾT BỊ TRONG NHÀ**

**GVDH: Vũ Văn Quang**

**SVTH: Nhóm 4:**

**Trần Duy Nam - 21012079**

**Vũ Ngọc Văn - 21012100**

**Phan Thúy Nga - 21012080**

**Lớp: CSE703018-1-2-23(N01)**

**Hà Nội, Tháng 3 Năm 2024**

# LỜI NÓI ĐẦU

**-----****-----**

Trong thời kỳ ngày nay, khi sự tiến bộ khoa học và kỹ thuật đang ngày càng phát triển, thế giới chúng ta đang chứng kiến một cuộc cách mạng về mặt văn minh và hiện đại. Cụ thể, sự tiến bộ đáng kể trong lĩnh vực kỹ thuật điện tử đã mở ra những khả năng mới, tạo ra những thiết bị thông minh với những đặc tính nổi bật như độ chính xác cao, tốc độ nhanh, và tính di động, tất cả đều mang lại những lợi ích quan trọng cho cuộc sống hàng ngày.

Cụ thể, ứng dụng của kỹ thuật điện tử không chỉ giúp tự động hóa nhiều khía cạnh trong cuộc sống hàng ngày của con người mà còn thúc đẩy sự phát triển của các hệ thống thông minh. Các hệ thống nhà cửa, nhà máy, trường học, giao thông,… Những hệ thống này không chỉ mang lại tính an toàn cao mà còn tạo ra môi trường sống thông minh, cho phép người dùng kiểm soát và quản lý các thiết bị trong môi trường sinh hoạt một cách linh hoạt và thuận tiện. Đây là một hướng nghiên cứu mang tính đổi mới, đáp ứng nhu cầu ngày càng cao của xã hội đương đại.

Xuất phát từ ứng dụng trên, nhóm em đã chọn đề tài “Thiết kế hệ thống mở khóa cửa thông minh & điều khiển các thiết bị trong nhà” cho môn Hệ nhúng.

# LỜI CẢM ƠN

**-----****-----**

Nhóm em muốn bày tỏ lòng biết ơn chân thành đến thầy Vũ Văn Quang đã dành thời gian và công sức hướng dẫn tận tình, cũng như tạo điều kiện thuận lợi cho nhóm em trong quá trình thực hiện đề tài. Sự nhiệt huyết và sự hỗ trợ không ngừng từ thầy đã đóng góp quan trọng vào việc xây dựng nền tảng kiến thức của nhóm em, giúp nhóm vượt qua những thách thức trong quá trình nghiên cứu.

Trong suốt hành trình này, nhóm nhận thức được rằng không tránh khỏi những thiếu sót và khuyết điểm. Nhóm 4 trân trọng những góp ý và sự chỉ dẫn từ quý thầy/cô, và nhóm cam kết sẽ tích cực thấu hiểu và áp dụng những phản hồi quý báu đó để đưa đề tài của mình lên một tầm cao mới. Sự hỗ trợ và tình cảm quý báu của quý thầy/cô là động viên lớn, là động lực mạnh mẽ giúp em không ngừng phát triển trong sự nghiệp nghiên cứu. Nhóm em xin chân thành cảm ơn!

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

….......................................................................................................................................

...........................................................................................................................................

................................................................................................................................….......

................................................................................................................................….......

................................................................................................................................….......

................................................................................................................................….......

................................................................................................................................….......

................................................................................................................................….......

................................................................................................................................….......

...............................................................................................................................…........

...............................................................................................................................…........

...............................................................................................................................…........

...........................................................................................................................................

….......................................................................................................................................

….......................................................................................................................................

….......................................................................................................................................

….......................................................................................................................................

….......................................................................................................................................

….......................................................................................................................................

….......................................................................................................................................

Hà Nội, ngày … tháng … năm 2024

**GIẢNG VIÊN HƯỚNG DẪN**

(Ký và ghi rõ họ tên)

# TIẾN ĐỘ THỰC HIỆN ĐỒ ÁN MÔN HỌC

**Tên đề tài:** THIẾT KẾ HỆ THỐNG MỞ KHÓA CỬA THÔNG MINH & ĐIỀU KHIỂN CÁC THIẾT BỊ TRONG NHÀ

**Ngày giao đề tài: 4**/12/2023; **Tuần thứ: 1**

**Ngày hoàn thành đề tài:** 4/3/2024; Tuần thứ: 12

Nhóm sinh viên thực hiện: Nhóm 4

Họ tên sinh viên:

Trần Duy Nam…..MSSV: 21012079

Vũ Ngọc Văn……MSSV: 21012100

Phan Thúy Nga….MSSV: 21012080

|  |  |
| --- | --- |
| ***Tuần/ngày*** | ***Nội dung – công việc thực hiện*** |
| Tuần 1,2,3 | Đăng kí đồ án, gặp giáo viên hướng dẫn để đề xuất đề tài đồ án,  chờ giáo viên hướng dẫn xét duyệt đề tài đồ án. |
| Tuần 4,5,6 | Tìm hiểu lên ý tưởng và thiết kế sơ đồ nguyên lí |
| Tuần 7,8,9 | Kiểm tra sơ đồ nguyên lý, sửa lỗi, mô phỏng và triển khai mạch đơn giản, báo cáo cho giảng viên xem xét và chỉnh sửa |
| Tuần 10,11 | Tiến hành triển khai mạch, code ứng dụng, sửa lỗi, hoàn thiện mạch và ứng dụng |
| Tuần 12 | Gửi báo cáo cho giáo viên hướng dẫn kiểm tra lần cuối xem xét và góp ý kiến, sau đó nộp đồ án và báo cáo trước khi bảo vệ. |

**BẢNG** **PHÂN CÔNG NHIỆM VỤ**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Mã sinh viên** | **Các công việc thực hiện** |
| Trần Duy Nam | 21012079 | Thảo luận thiết kế mạch  Triển khai thực hiện code mạch ESP32, ArduinoUno  Thực hiện code chức năng của ứng dụng quản lý trên Flutter  Làm báo cáo word |
| Vũ Ngọc Văn | 21012100 | Thảo luận thiết kế mạch  Triển khai thực hiện code mạch ArduinoUno  Thực hiện code chức năng của ứng dụng quản lý trên Flutter  Làm báo cáo word |
| Phan Thúy Nga | 21012080 | Kết nối ESP32 với Firebase  Triển khai thực hiện code mạch ArduinoUno  Thực hiện code chức năng của ứng dụng quản lý trên Flutter  Làm báo cáo word |

Xác nhận của giáo viên hướng dẫn

(Ký và ghi rõ họ và tên)

# LỜI CAM ĐOAN

**-----****-----**

Đề tài này là do nhóm em tự thực hiện dựa vào một số tài liệu và không sao chép từ tài liệu hay công trình đã có trước đó. Nếu có sao chép em xin hoàn toàn chịu trách nhiệm.

Hà Nội, ngày 04 tháng 03 năm 2024

**NGƯỜI VIẾT**

**NHÓM 4**

# MỤC LỤC

**-----****-----**

[LỜI NÓI ĐẦU 2](#_Toc160620368)

[LỜI CẢM ƠN 3](#_Toc160620369)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 4](#_Toc160620370)

[TIẾN ĐỘ THỰC HIỆN ĐỒ ÁN MÔN HỌC 5](#_Toc160620371)

[LỜI CAM ĐOAN 8](#_Toc160620372)

[MỤC LỤC 9](#_Toc160620373)

[DANH MỤC HÌNH ẢNH 11](#_Toc160620374)

[DANH MỤC BẢNG 12](#_Toc160620375)

[CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI 13](#_Toc160620376)

[1.1 Giới thiệu đề tài 13](#_Toc160620377)

[1.2 Phương tiện nghiên cứu. 13](#_Toc160620378)

[1.3 Mục đích và yêu cầu đề tài. 13](#_Toc160620379)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 14](#_Toc160620380)

[2.1 Modul ESP32 14](#_Toc160620381)

[2.2 Arduino Uno R3 19](#_Toc160620382)

[2.3 LCD 16x2 20](#_Toc160620383)

[2.4 Bàn phím ma trận mềm 4x4 Keypad 21](#_Toc160620384)

[2.5 Cảm Biến Thân Nhiệt Chuyển Động PIR HC-SR501 22](#_Toc160620385)

[2.6 Servo SG90 23](#_Toc160620386)

[2.7 Modul Relay 24](#_Toc160620387)

[2.8 Điện trở. 25](#_Toc160620388)

[2.9 Led. 25](#_Toc160620389)

[2.10 Nút nhấn (Button) 26](#_Toc160620390)

[2.11 Ứng dụng quản lý trên điện thoại di động 27](#_Toc160620391)

[2.12: Firebase - RealTime 29](#_Toc160620392)

[CHƯƠNG III: THIẾT KẾ VÀ THI CÔNG MẠCH 30](#_Toc160620393)

[3.1 Sơ đồ khối 30](#_Toc160620394)

[3.2 Sơ đồ nguyên lý và chức năng của từng khối. 30](#_Toc160620395)

[3.2.1 Khối công suất 30](#_Toc160620396)

[3.2.2Khối hệ nhúng 31](#_Toc160620397)

[3.2.3 Khối thời gian thực 32](#_Toc160620398)

[3.2.4 Khối hiển thị LCD 16X2 34](#_Toc160620399)

[3.2.5. Khối cảm biến 35](#_Toc160620400)

[3.3 Thi công. 36](#_Toc160620401)

[3.3.1 Mạch in. 36](#_Toc160620402)

[3.3.2 Mạch hoàn chỉnh 36](#_Toc160620403)

[3.3.3 Code chương trình hệ nhúng 37](#_Toc160620404)

[CHƯƠNG IV: KẾT QUẢ THỰC HIỆN VÀ KẾT LUẬN 69](#_Toc160620405)

[4.1 Ưu, nhược điểm của sản phẩm 69](#_Toc160620406)

[4.2 Kết luận. 69](#_Toc160620407)

[TÀI LIỆU THAM KHẢO 70](#_Toc160620408)

# DANH MỤC HÌNH ẢNH

**-----****-----**

[Hình 2. 1: Modul Esp32 14](#_bookmark13)

[Hình 2.2: Arduino Uno R3 19](#_bookmark15)

[Hình 2.3: LCD 16x2 20](#_bookmark17)

[Hình 2.4: Bàn phím ma trận mềm 4x4 Keypad 21](#_bookmark19)

[Hình 2.5: Cảm Biến Thân Nhiệt Chuyển Động PIR HC-SR501 22](#_bookmark21)

[Hình 2.6: Động cơ Servo SG90 23](#_bookmark23)

[Hình 2.7 : Modul Relay 1 kênh(5-10V) 24](#_bookmark26)

[Hình 2.8: Điện trở dạng cắm 25](#_bookmark29)

[Hình 2.9: Bóng led đủ màu 25](#_bookmark31)

[Hình 2.10: Button 26](#_bookmark33)

[Hình 2.11: Giao diện ứng dụng quản lý qua điện thoại 27](#_bookmark35)

[Hình 2.12: Giao diện đổi mật khẩu cửa 27](#_bookmark36)

[Hình 2.13: Giao diện theo dõi lịch sử đóng mở cửa 28](#_bookmark39)

[Hình 2.14: Cơ sở dữ liệu ReamTime FiseBaFirebase 29](#_bookmark41)

[Hình 3.1: Sơ đồ khối 30](#_bookmark41)

[Hình 3.2: Khối công suất 30](#_bookmark41)

[Hình 3.3: Arduino R3 31](#_bookmark41)

[Hình 3.4: ESP32 32](#_bookmark41)

[Hình 3.5: Khối thời gian thực 33](#_bookmark41)

[Hình 3.6: Khối hiển thị 34](#_bookmark41)

[Hình 3.7: Cảm biến chuyển động 35](#_bookmark41)

[Hình 3.8: Sơ đồ nguyên lý chung 35](#_bookmark41)

[Hình 3.9: Sơ đồ mạch in 36](#_bookmark41)

[Hình 3.10: Mạch hoàn chỉnh 36](#_bookmark41)

# DANH MỤC BẢNG

BẢNG PHÂN CÔNG NHIỆM VỤ 6  
Bảng 1.1: Bảng khả nắng sử dụng từng chân của ESP32 13

Bảng 2.1: Bảng thông số của Arduino Uno R3… 17

Bảng 2.2: Chức năng từng chân của LCD 6x2 18

# CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu đề tài

* + - Với sự bùng nổ của Internet of Things (IoT), khái niệm ngôi nhà thông minh không chỉ đơn thuần là một xu hướng mà còn là nhu cầu thiết yếu. Ngôi nhà thông minh không chỉ là nơi ẩn mình của gia đình mà còn là trung tâm điều khiển linh hoạt cho mọi hoạt động.
    - Để giải quyết thách thức quản lý an ninh và điều khiển thiết bị, em quyết định tập trung nghiên cứu và phát triển "Hệ Thống Mở Khóa Cửa Thông Minh & Điều Khiển Các Thiết Bị Trong Nhà".

## Phương tiện nghiên cứu.

* Tài liệu tổng hợp từ một số nguồn khác nhau.
* Tìm kiếm thông tin trên internet.
* Các phần mềm hỗ trợ: Protues, Arduino IDE

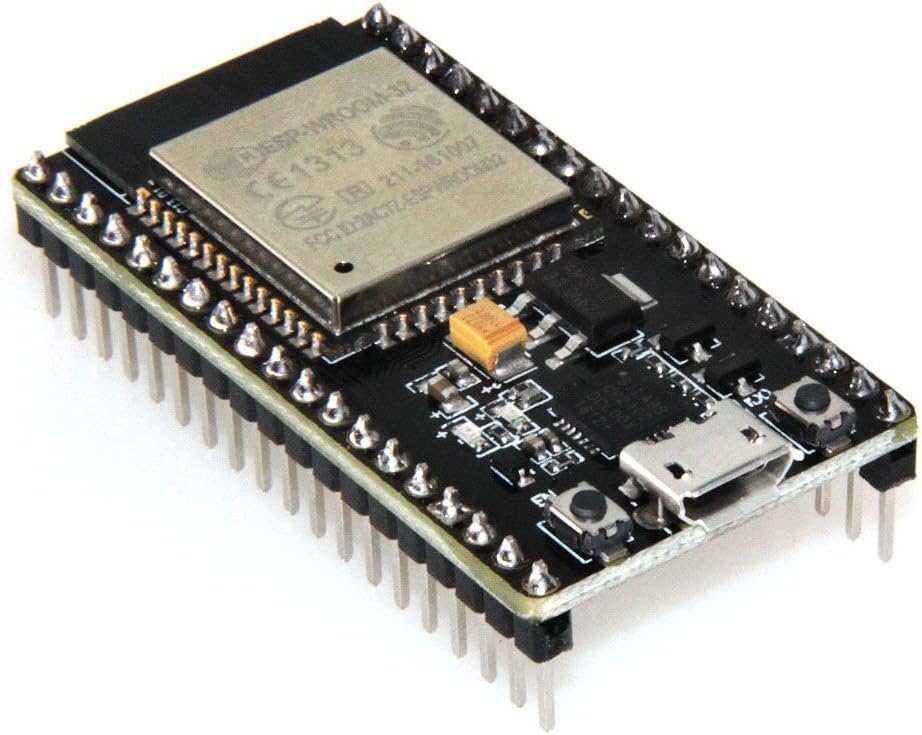
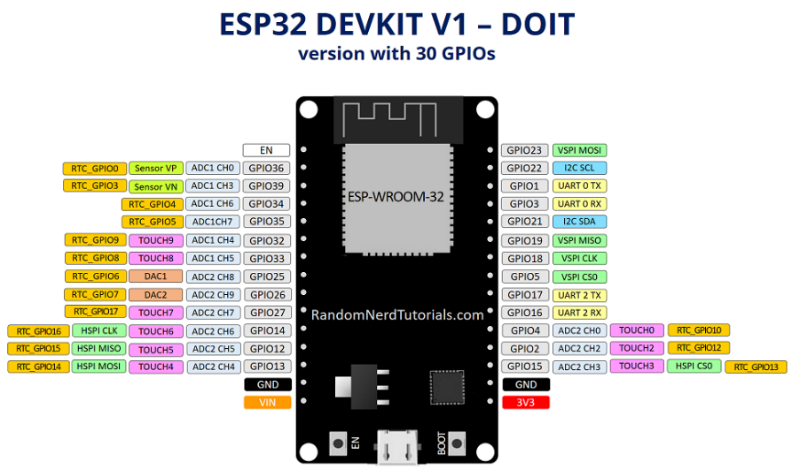
## Mục đích và yêu cầu đề tài.

* Hệ thống mở cửa từ xa, quản lý và thay đổi mật khẩu cửa
* Có khả năng ghi lại thời gian ra vào, đóng mở cửa
* Bật tắt các thiết bị trong nhà: Đèn, nóng lạnh, điều hòa
* Hệ thống làm việc ổn định.
* Có thể áp dụng mô hình trong thực tế

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT

Để thực hiện mạch mở khóa cửa thông và điều khiển thiết bị trong nhà cần sử dụng nhiều loại linh kiện khác nhau kết hợp với phần mềm ứng dụng điện thoại di động Flutter để có thể điều khiển và quản lý từ xa một cách linh hoạt, sau đây là một số linh kiện chính có trong mạch và phần mềm ứng dụng điện thoại:

## Modul ESP32

****

#### Hình 2.1: Modul Esp32

* + - ESP32 là một bộ vi điều khiển thuộc danh mục vi điều khiển trên chip công suất thấp và tiết kiệm chi phí. Hầu hết tất cả các biến thể ESP32 đều tích hợp Bluetooth và Wi-Fi chế độ kép, làm cho nó có tính linh hoạt cao, mạnh mẽ và đáng tin cậy cho nhiều ứng dụng.
    - Nó là sự kế thừa của vi điều khiển NodeMCU ESP8266 phổ biến và cung cấp hiệu suất và tính năng tốt hơn. Bộ vi điều khiển ESP32 được sản xuất bởi Espressif Systems và được sử dụng rộng rãi trong nhiều ứng dụng khác nhau như IoT, robot và tự động hóa.
    - ESP32 cũng được thiết kế để tiêu thụ điện năng thấp, lý tưởng cho các ứng dụng chạy bằng pin. Nó có hệ thống quản lý năng lượng cho phép nó hoạt động ở chế độ ngủ và chỉ thức dậy khi cần thiết, điều này có thể kéo dài tuổi thọ pin rất nhiều.

#### Bảng 2.1: Bảng khả nắng sử dụng từng chân của ESP32

|  |  |  |  |
| --- | --- | --- | --- |
| GPIO Pin | Đầu vào INPUT | Đầu ra  OUTPUT | Chức năng |
| 0 | Pull-up | OK | Xuất tín hiệu PWM khi khởi động |
| 1 | Chân TX (nút) | OK | Đầu ra debug khi khởi động |
| 2 | OK |  | Kết nối đến đèn LED trên bo mạch |
| 3 | OK | RX (nút) | Mức HIGH khi khởi động |
| 4 | OK |  |  |
| 5 | OK |  | Xuất tín hiệu PWM khi khởi động |
| 6 | X | X | Kết nối đến bộ nhớ flash tích hợp |
| 7 | X | X | Kết nối đến bộ nhớ flash tích hợp |
| 8 | X | X | Kết nối đến bộ nhớ flash tích hợp |
| 9 | X | X | Kết nối đến bộ nhớ flash tích hợp |
| 10 | X | X | Kết nối đến bộ nhớ flash tích hợp |
| 11 | X | X | Kết nối đến bộ nhớ flash tích hợp |
| 12 | OK |  | Lỗi khởi động nếu kéo lên cao |
| 13 | OK |  |  |
| 14 | OK |  | Xuất tín hiệu PWM khi khởi động |
| 15 | OK |  | Xuất tín hiệu PWM khi khởi động |
| 16 | OK |  |  |
| 17 | OK | OK |  |
| 18 | OK | OK |  |
| 19 | OK | OK |  |
| 20 | OK | OK |  |
| 21 | OK | OK |  |
| 22 | OK | OK |  |
| 23 | OK | OK |  |
| 24 | OK | OK |  |
| 25 | OK | OK |  |
| 26 | OK | OK |  |
| 27 | OK | OK |  |
| 28 | OK | OK |  |
| 29 | OK | OK |  |
| 30 | OK | OK |  |
| 31 | OK | OK |  |
| 32 | OK | OK |  |
| 33 | OK | OK |  |
| 34 | OK | OK |  |
| 35 | OK | OK |  |
| 36 | OK |  | Chỉ đầu vào |
| 37 | OK |  | Chỉ đầu vào |
| 38 | OK |  | Chỉ đầu vào |
| 39 | OK |  | Chỉ đầu vào |

* + - Các chân được đánh dấu là OK(hỗ trợ) để sử dụng.
    - Các chân được đánh dấu là PULL UP, TX(Chân truyền của UART), RX(Chân nhận của UART)thì có thể sử dụng, nhưng cần chú ý vì chúng có thể có hoạt động không mong muốn chủ yếu khi khởi động.
    - Các chân được đánh dấu X không được khuyến khích sử dụng làm đầu vào hoặc đầu ra .

**Mô tả các chân ESP32:**

**Chân Input Only:** GPIO từ 34 đến 39 là GPI – chân chỉ đầu vào. Các chân này không có điện trở kéo lên hoặc kéo xuống bên trong. Chúng không thể được sử dụng làm đầu ra, vì vậy chỉ sử dụng các chân này làm đầu vào:

* GPIO 34
* GPIO 35
* GPIO 36
* GPIO 39

**Chân tích hợp Flash trên ESP32**: GPIO 6 đến GPIO 11 dùng để kết nối Flash SPI, không khuyến khích sử dụng trong các ứng dụng khác

* GPIO 6 (SCK/CLK)
* GPIO 7 (SDO/SD0)
* GPIO 8 (SDI/SD1)
* GPIO 9 (SHD/SD2)
* GPIO 10 (SWP/SD3)
* GPIO 11 (CSC/CMD)

**Chân cảm biến điện dung:** Các chân ESP32 này có chức năng như 1 nút nhấn cảm ứng, có thể phát hiện sự thay đổi về điện áp cảm ứng trên chân. Các cảm biến cảm ứng bên trong đó được kết nối với các GPIO sau:

* T0 (GPIO 4)
* T1 (GPIO 0)
* T2 (GPIO 2)
* T3 (GPIO 15)
* T4 (GPIO 13)
* T5 (GPIO 12)
* T6 (GPIO 14)
* T7 (GPIO 27)
* T8 (GPIO 33)
* T9 (GPIO 32)

**Analog to Digital Converter (ADC):** ESP32 có các kênh đầu vào ADC 18 x 12 bit (trong khi ESP8266 chỉ có ADC 1x 10 bit). Đây là các GPIO có thể được sử dụng làm ADC và các kênh tương ứng:

* ADC1\_CH0 (GPIO 36)
* ADC1\_CH1 (GPIO 37)
* ADC1\_CH2 (GPIO 38)
* ADC1\_CH3 (GPIO 39)
* ADC1\_CH4 (GPIO 32)
* ADC1\_CH5 (GPIO 33)
* ADC1\_CH6 (GPIO 34)
* ADC1\_CH7 (GPIO 35)
* ADC2\_CH0 (GPIO 4)
* ADC2\_CH1 (GPIO 0)
* ADC2\_CH2 (GPIO 2)
* ADC2\_CH3 (GPIO 15)
* ADC2\_CH4 (GPIO 13)
* ADC2\_CH5 (GPIO 12)
* ADC2\_CH6 (GPIO 14)
* ADC2\_CH7 (GPIO 27)
* ADC2\_CH8 (GPIO 25)
* ADC2\_CH9 (GPIO 26)

**Digital to Analog Converter (DAC):** Có các kênh DAC 2 x 8 bit trên ESP32 để chuyển đổi tín hiệu kỹ thuật số thành đầu ra tín hiệu điện áp tương tự. Các kênh này chỉ có độ phân giải 8 bit, nghĩa là có giá trị từ 0 – 255 tương ứng với 0 – 3.3V. Đây là các kênh DAC

* DAC1 (GPIO25)
* DAC2 (GPIO26)

**Các chân thời gian thực RTC:** Các chân này có tác dụng đánh thức ESP32 khi trong chế độ Low Power Mode. Sử dụng như 1 chân ngắt ngoài.

* RTC\_GPIO0 (GPIO36)
* RTC\_GPIO3 (GPIO39)
* RTC\_GPIO4 (GPIO34)
* RTC\_GPIO5 (GPIO35)
* RTC\_GPIO6 (GPIO25)
* RTC\_GPIO7 (GPIO26)
* RTC\_GPIO8 (GPIO33)
* RTC\_GPIO9 (GPIO32)
* RTC\_GPIO10 (GPIO4)
* RTC\_GPIO11 (GPIO0)
* RTC\_GPIO12 (GPIO2)
* RTC\_GPIO13 (GPIO15)
* RTC\_GPIO14 (GPIO13)
* RTC\_GPIO15 (GPIO12)
* RTC\_GPIO16 (GPIO14)
* RTC\_GPIO17 (GPIO27)

**Chân PWM:** ESP32 LED PWM có 16 kênh độc lập có thể được định cấu hình để tạo tín hiệu PWM với các thuộc tính khác nhau. Tất cả các chân có thể hoạt động như đầu ra đều có thể được sử dụng làm chân PWM (GPIO từ 34 đến 39 không thể tạo PWM). Để xuất PWM, cần xác định các thông số này trong code:

* Frequency – tần số
* Duty cycle
* Kênh PWM
* Chân GPIO nơi bạn muốn xuất tín hiệu

**Chân I2C:** ESP32 có hai kênh I2C và bất kỳ chân nào cũng có thể được đặt làm SDA hoặc SCL. Khi sử dụng ESP32 với Arduino IDE, các chân I2C mặc định là:

* GPIO 21 (SDA)
* GPIO 22 (SCL)

**Chân Ngắt Ngoài:** Tất cả các chân ESP32 đều có thể sử dụng ngắt ngoài

## Arduino Uno R3

|  |  |
| --- | --- |
|  | **Ảnh có chứa Linh kiện điện, mạch điện, Thành phần mạch điện, Kỹ thuật điện  Mô tả được tạo tự động** |

#### Hình 2.2: Arduino Uno R3

#### Bảng 2.2: Bảng thông số của Arduino Uno R3

|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 họ 8bit |
| Điện áp hoạt động | 5V DC (chỉ được cấp qua cổng USB) |
| Tần số hoạt động | 16 MHz |
| Dòng tiêu thụ | khoảng 30mA |
| Điện áp vào khuyên dung | 7-12V DC |
| Điện áp vào giới hạn | 6-20V DC |
| Số chân Digital I/O | 14 (6 chân hardware PWM) |
| Số chân Analog | 6 (độ phân giải 10bit) |
| Dòng tối đa trên mỗi chân I/O | 30 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa (3V) | 50 mA |
| Bộ nhớ Flash | 32 KB (ATmega38) với 0.5KB dùng bởi bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |

## LCD 16x2



#### Hình 2.3: LCD 16x2

* + - LCD16x2 là một màn hình hiển thị bao gồm nhiều ma trận nhỏ, khi hoạt động thì LCD16x2 sẽ hiển thị các kí tựTrong bảng mã ASCII . Hệ nhúng gửi các tín hiệu khởi tạocho LCD 6X2, sau đó hiển thị các được kí tự lên màn hình hiển thị.
    - Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của VĐK. 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 (chữ, số và kí tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống và giá thành rẻ.

#### Bảng 2.3: Chức năng từng chân của LCD 16x2

## Bàn phím ma trận mềm 4x4 Keypad

**Ảnh có chứa biểu đồ, hình vuông, Hình chữ nhật, hàng

Mô tả được tạo tự động**

#### Hình 2.4: Bàn phím ma trận mềm 4x4 Keypad

* + - Bàn phím mềm 4×4 keypad là một thiết bị đầu vào thường được sử dụng trong các dự án Arduino để nhập liệu từ người dùng. Đây là một loại bàn phím cảm ứng, thường được thiết kế với ma trận 4 hàng và 4 cột, tạo thành tổng cộng 16 nút nhấn. Mỗi nút nhấn có thể đại diện cho một ký tự, số hoặc chức năng khác.
    - Bàn phím mềm 4×4 keypad có thiết kế nhỏ gọn, dễ kết nối và sử dụng, các chân của 16 phím được nối theo ma trận, tín hiệu khi nhấn phím sẽ là tín hiệu GND (0VDC) hoặc Vcc (5VDC) tùy vào cách quét phím kích vào chân Vi điều khiển, bàn phím còn tích hợp vị trí để lắp thêm tụ chống dội (chống nhiễu), phù hợp cho các ứng dụng điều khiển bằng phím bấm.
    - Chức năng key phím 4×4:
      * + R1 : Ngõ ra hàng 1.
        + R2 : Ngõ ra hàng 2.
        + R3 : Ngõ ra hàng 3.
        + R4 : Ngõ ra hàng 4.
        + C1 : Ngõ vào hàng 1.
        + C2 : Ngõ vào hàng 2.
        + C3 : Ngõ vào hàng 3.
        + C4 : Ngõ vào hàng 4.

## Cảm Biến Thân Nhiệt Chuyển Động PIR HC-SR501

|  |  |
| --- | --- |
| Ảnh có chứa biểu đồ, bản phác thảo, hình vẽ, hàng  Mô tả được tạo tự động | Ảnh có chứa bánh xe, đồ chơi, phương tiện  Mô tả được tạo tự động |

#### Hình 2.5: Cảm Biến Thân Nhiệt Chuyển Động PIR HC-SR501

* + - Nguyên tắc hoạt động:
      * Cảm biến HC-SR501 hoạt động dựa trên nguyên lý cảm biến hồng ngoại không hoạt động.
      * Nó sử dụng các cảm biến PIR để phát hiện sự chuyển động trong lĩnh vực quét của nó.
    - Đặc điểm kỹ thuật cơ bản:
      * Khoảng cách phát hiện: Thường có thể phát hiện chuyển động trong khoảng từ vài mét đến vài chục mét, tùy thuộc vào mô hình cụ thể.
      * Thời gian hoạt động: Có thể điều chỉnh thời gian hoạt động sau khi phát hiện chuyển động, thường từ một vài giây đến vài phút.
      * Góc quét: Cảm biến thường có một góc quét rộng, cho phép phát hiện chuyển động trong một diện tích lớn.
    - Chân kết nối:
      * HC-SR501 thường có ba chân kết nối chính: VCC (nguồn), GND (đất), và OUT (đầu ra).
      * Nó có thể được nối với các vi điều khiển như Arduino thông qua các chân kết nối này.
    - Chức năng đặc biệt:
      * Điều chỉnh độ nhạy: Một số mô hình có thể được điều chỉnh để tăng hoặc giảm độ nhạy của cảm biến.
      * Điều chỉnh thời gian hoạt động: Người dùng có thể điều chỉnh thời gian mà cảm biến sẽ giữ trạng thái báo động sau khi phát hiện chuyển động.

## Servo SG90

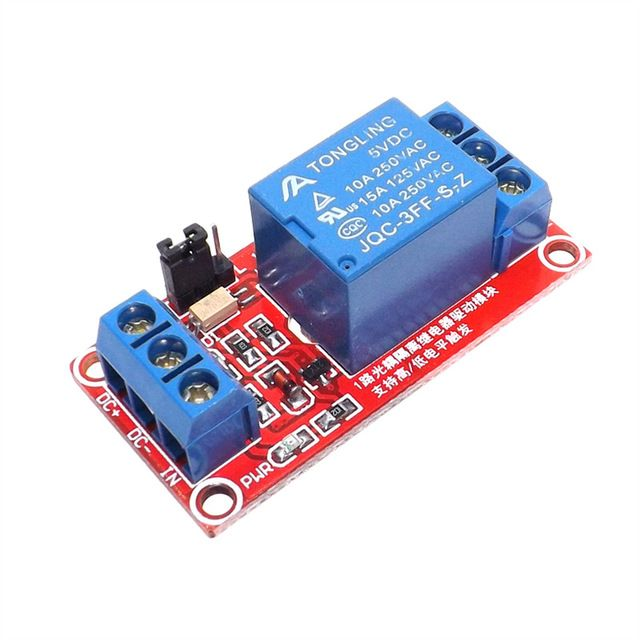
Ảnh có chứa nhựa, dây cáp, Xanh điện, màu xanh lam

Mô tả được tạo tự động

#### Hình 2.6: Động cơ Servo SG90

* + - Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ 0o - 180o. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô mình), có loại thì sở hữu một momen lực bá đạo (vài chục Newton/m), hoặc có loại thì khỏe và nhông sắc chắc chắn,...
    - Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bầt kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác. Các động cơ servo điều khiển bằng liên lạc vô tuyến được gọi là động cơ servo RC (radio-controlled). Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.
    - Nguyên tắc hoạt động
      * Động cơ servo sử dụng một hệ thống phản hồi để duy trì vị trí cố định hoặc thực hiện chuyển động theo yêu cầu.
      * Bên trong động cơ servo thường có một bộ cảm biến (thường là potentiometer) để đo góc quay hoặc vị trí của trục động cơ.
    - Chân kết nối
      * Nguồn điện: Động cơ servo thường được cung cấp điện áp DC, và có thể yêu cầu điện áp cụ thể tùy thuộc vào mô hình.
      * Chân điều khiển: Có thể điều khiển thông qua các chân điều khiển như PWM (Pulse Width Modulation) từ các vi điều khiển như Arduino.

## Modul Relay



#### Hình 2.7: Modul Relay 1 kênh(5-10V)

* + - Module Relay 1 kênh là một thiết bị điện tử được sử dụng để kiểm soát các thiết bị điện khác thông qua một tín hiệu điều khiển, thường là từ một vi điều khiển như Arduino hoặc Raspberry Pi. Dưới đây là một giới thiệu về một Module Relay 1 kênh với dải điện áp từ 5V đến 10V:
    - Nguyên tắc hoạt động:
      * Module Relay hoạt động dựa trên nguyên tắc sử dụng một cuộn dây cuộn để tạo ra một từ trường khi có điện áp được cấp.
      * Khi từ trường này được tạo ra, nó làm chuyển động một cơ cấu cơ khí hoặc điện từ để mở hoặc đóng các tiếp điểm relay.
    - Cấu trúc cơ bản:
      * Tiếp điểm Relay: Module này thường có một hoặc nhiều tiếp điểm relay, có thể là Normally Open (NO), Normally Closed (NC) hoặc có thể là cả hai.
      * Đèn LED báo trạng thái: Có thể có một đèn LED tích hợp để báo hiệu trạng thái hoạt động của relay.
    - Chân kết nối:
      * VCC và GND: Chân cấp nguồn cho relay.
      * IN (hoặc Signal): Chân này dùng để kết nối với nguồn điều khiển như Arduino, Raspberry Pi để kiểm soát relay.

## Điện trở.

#### Hình 2.8: Điện trở dạng cắm

* + - Điện trở là linh kiện thụ động có tác dụng cản trở cả dòng và áp. Điện trở đựơc sử dụng rất nhiều trong các mạch điện tử.
    - Điện trở của dây dẫn có trị số điện trở lớn hay nhỏ tùy thuộc vào vật liệu làm dây, tỉ lệ thuận với chiều dài và tỉ lệ nghịch với tiết diện dây dẫn.

## 2.9 Led.



#### Hình 2.9: Bóng led đủ màu

* + - Bóng đèn LED trong Arduino thường được sử dụng như một nguồn sáng dễ kết nối và kiểm soát. Bằng cách sử dụng chân digital output, người dùng có thể bật tắt bóng đèn LED và thậm chí thực hiện các hiệu ứng ánh sáng đơn giản trong các dự án điện tử.

## 2.10 Nút nhấn (Button)



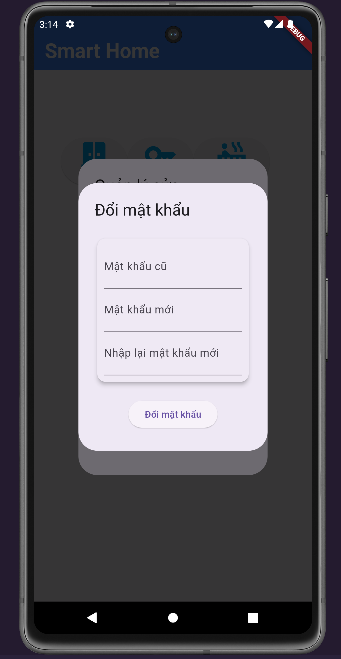
#### Hình 2.10: Button

* + - Nút nhấn (button) trong Arduino là một cảm biến đơn giản được sử dụng để nhận tín hiệu từ người dùng. Khi được nhấn, nút nhấn tạo ra một tín hiệu digital mà Arduino có thể sử dụng để kích thích các hành động hoặc điều khiển trong các dự án điện tử.

## 2.11 Ứng dụng quản lý trên điện thoại di động

|  |  |  |  |
| --- | --- | --- | --- |
| Ảnh có chứa văn bản, ảnh chụp màn hình, đa phương tiện, đồ phụ tùng  Mô tả được tạo tự động |  |  |  |

#### Hình 2.11: Giao diện ứng dụng quản lý qua điện thoại



#### Hình 2.12: Giao diện đổi mật khẩu cửa

|  |  |  |
| --- | --- | --- |
|  |  |  |

#### Hình 2.13: Giao diện theo dõi lịch sử đóng mở cửa

* + - Ứng dụng quản lý nhà thông minh là một tổ hợp hiện đại và linh hoạt, tích hợp nhiều thiết bị và cảm biến để tối ưu hóa trải nghiệm sống. Có thể điều khiển cửa ra vào thông qua ứng dụng di động, đồng thời theo dõi trạng thái mở/đóng của cửa và lưu trữ lịch sử sử dụng. Ánh sáng trong nhà có thể được điều chỉnh thông qua công tắc thông minh.
    - Ứng dụng quản lý qua điện thoại di động bao gồm 5 chức năng chính:
      * Ứng dụng quản lý qua điện thoại di động bao gồm 5 chức năng chính

Hiển thị trạng thái cửa, quản lý đóng, mở

Hiển thị trạng thái đèn, quản lý bật tắt hệ thống đèn trong nhà

Hiển thị trạng thái nóng lạnh, quản lý bật tắt nóng lạnh

Đổi mật khẩu cửa

Xem lịch sử đóng mở cửa

* + - * Hoạt động dựa trên kết nối giữa Model Esp32 – FireBase – Flutter
      * An toàn và bảo mật cao
      * Giao diện thân thiện dễ dàng sử dụng
    - Tất cả các thông tin về trạng thái và lịch sử sử dụng được lưu trữ an toàn trong cơ sở dữ liệu điện toán đám mây, giúp tôi dễ dàng theo dõi và quản lý hệ thống từ bất kỳ đâu. Hệ thống nhà thông minh không chỉ tăng cường sự thoải mái mà còn mang lại sự hiệu quả và an ninh trong mọi khía cạnh của cuộc sống hàng ngày.

## 2.12: Firebase - RealTime

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

#### Hình 2.14: Cơ sở dữ liệu ReamTime FiseBaFirebase

* + - Firebase Realtime Database của Google là một dịch vụ cơ sở dữ liệu thời gian thực linh hoạt và dễ tích hợp, được thiết kế để hỗ trợ xây dựng ứng dụng web và di động với khả năng đồng bộ dữ liệu ngay lập tức. Dữ liệu được tổ chức dưới định dạng JSON, giúp dễ dàng quản lý và truy cập. Điểm mạnh của nó nằm ở khả năng đồng bộ thời gian thực, hỗ trợ chế độ offline, tích hợp chặt chẽ với Firebase Authentication để quản lý danh tính và quyền truy cập. Firebase Realtime Database cũng cung cấp quy tắc an ninh, cho phép người phát triển kiểm soát quyền truy cập dữ liệu. Đồng thời, nó có tích hợp tốt với các dịch vụ khác của Firebase, tạo nên một hệ sinh thái mạnh mẽ để phát triển ứng dụng.
    - Một điểm đặc biệt quan trọng là khả năng tích hợp với các sự kiện thay đổi dữ liệu, giúp người phát triển dễ dàng thực hiện các tác vụ như cập nhật giao diện người dùng khi có sự thay đổi dữ liệu từ phía máy chủ. Điều này tăng tính tương tác và trải nghiệm người dùng trong ứng dụng.
    - Với Firebase Realtime Database, việc quản lý và theo dõi dữ liệu trở nên hiệu quả, giúp đẩy nhanh quá trình phát triển ứng dụng thời gian thực mà không cần lo lắng về việc xây dựng và duy trì hệ thống cơ sở dữ liệu.

# CHƯƠNG III: THIẾT KẾ VÀ THI CÔNG MẠCH

## 3.1 Sơ đồ khối

**KHỐI NGUỒN**

**KHỐI HIỂN THỊ**

**KHỐI HỆ NHÚNG**

**KHỐI THỜI GIAN THỰC**

**KHỐI CÔNG SUẤT**

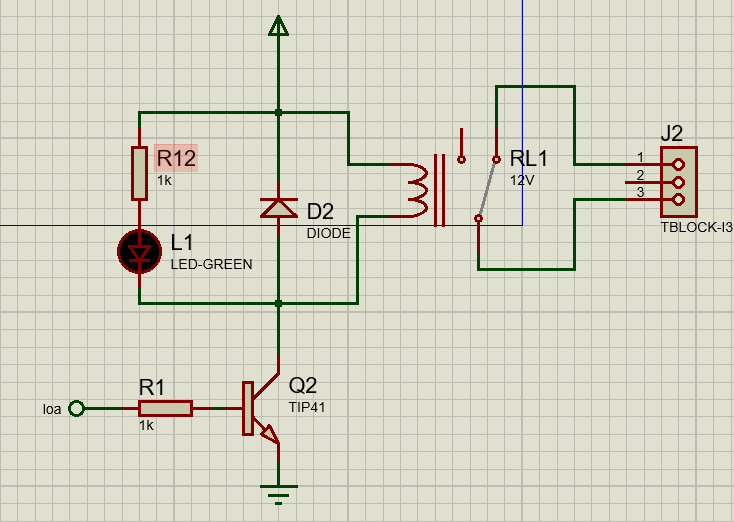
**KHỐI CẢM BIẾN**

#### Hình 3.1: Sơ đồ khối

## 3.2 Sơ đồ nguyên lý và chức năng của từng khối.

### 3.2.1 Khối công suất

Do cuộn hút của chuông điện sử dụng nguồn điện xoay chiều 220VAC nên ta dụng Tranzitor điều khiển cuộn hút relay hoặc công tắc tơ, relay và công tắc tơ có tác dụng cách li về điện với mạch động lực và nó điều khiển đóng ngắt chuông điện.

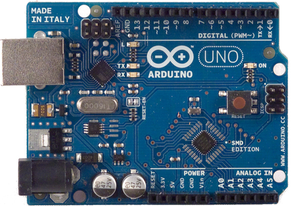


#### Hình 3.2: Khối công suất

* Nguyên lý hoạt động:
  + Khi tín hiệu đưa vào là mức 0 (Tức =0V) thì Q2 không dẫn do không có dòng IBE >> Relay không làm việc.
  + Khi tín hiệu đưa vào là mức (Tức =5V) thì sẽ qua R hạn dòng làm cho Q2 dẫn thông lúc này ta có dòng Ice là dòng điện chạy qua cuộn dây >> Q2 >> MASS, Relay( 2V) đóng tiếp điểm thường mở (điều khiển thiết bị nào đó).
  + Diode D2 trong mạch có tác dụng chống lại dòng điện cảm ứng do cuộn đây sinh ra làm hỏng tranzitor. Relay dùng để điều khiển chuông điện 220V
  + Mục đích của R là tạo dòng vào cực B của trans tới ngưỡng bão hòa để trans hoạt động như chiếc khóa có điều kiện.

### 3.2.2Khối hệ nhúng

#### **3.2.2a Arduino Uno R3**



##### *Hình 3.3: Arduino Uno R3*

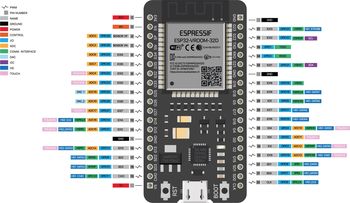
- GND (Ground): cực âm của nguồn điện cấp cho Arduino UNO.   
- 5V: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.  
- 3.3V: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.  
- RESET: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

- 2 chân Serial: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với ESP32 thông qua 2 chân này.   
- Chân 2 - 9: là 8 chân được kết nối với keypad 4x4   
- Chân 10 : Chân cảm biến

- Chân 11: là chân kết nối để điều khiển servo

- Chân 12: là chân nhận dữ liệu từ nút bấm cơ

#### **3.2.2b: ESP32**

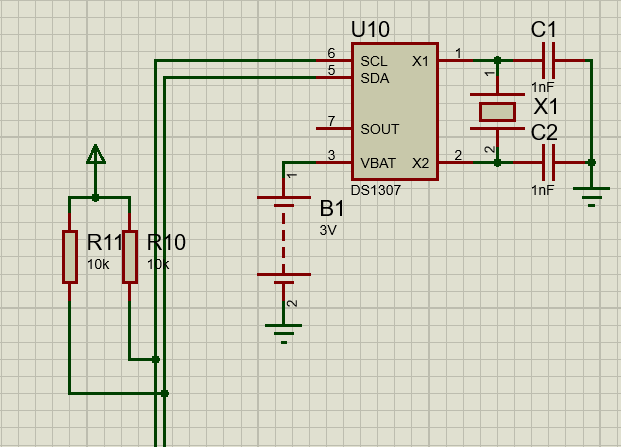


##### *Hình 3.4: ESP32*

* Chân 16,17 : Được nối với 2 chân RX TX của Arduino
* 3 chân đèn là 12 27 25
* Chân SDA 21 và SCL 22 được nối với thời gian thực(PIR)

### 3.2.3 Khối thời gian thực

* + - * VCC,GND: nguồn 1 chiều được cung cấp tới các chân này. VCC là đầu vào 5V. Khi 5V được cung cấp thì thiết bị đó có thể truy cập hoàn chỉnh và dữ liệu có thể đọc và viết.
      * Khi pin 3V được kết nối tới thiết và vcc nhỏ hơn .25Vbat thì quá trình đọc và viết không được thực thi, tuy nhiên chức năng timekeeping không bị ảnh hưởng bởi điện áp vào thấp khi VCC nhỏ hơn Vbat thì RAM và time keeper sẽ được ngắt tới nguồn cung cấp( 3-5VDC).



#### Hình 3.5: Khối thời gian thực

* Vbat: đầu vào pin cho bất kỳ một chuẩn pin 3V. Điện áp pin phải giữ trong khoảng 2.5-3V để đảm bảo cho thiết bị hoạt động tốt.
* SCL(serial clock input): SCL được xử dụng để đồng bộ sự chuyển dữ liệu trên đường dây nối tiếp.
* SDA(serial data input/output): là chân ra vào cho 2 đường dây nối tiếp. chân SDA được thiết kế theo kiêu cực máng hở, vì vậy phải có điện trở R10=10K và R11=10K kéo lên trong khi hoạt động.
* X1,X2: được nối với thạch anh với tần số 32,768 kHz. là một mạch tạo dao động ngoài, để hoạt động ổn định ta nói thêm 2 tụ C1,C2 với giá trị 33pF.

### 3.2.4 Khối hiển thị LCD 16X2

#### Hình 3.6: Khối hiển thị

* Các chân 1,2,3 là các chân VSS , VDD, VEE trong đó VSS chân nối đất , VEE chân chọn độ tương phản chân này dc chọn qua 1 biến trở 5K một đầu nối VEE

, một đầu nối mass chân VSS, một đầu nối nguồn chân VDD.

* Các chân D4, D5, D6, D7 được nối lần lượt với PORTD.4, PORTD.5, PORTD.6, PORTD.7
* Khối hiển thị dùng để hiển thị các thông tin: giờ, phút, giây và thứ, ngày, tháng, năm. Có 3 nút nhấn để điều chỉnh các thời gian hiển thị trên LCD.

### 3.2.5. Khối cảm biến



#### Hình 3.7. Cảm biến chuyển động

Trên đây là đầu dò PIR, loại bên trong gắn 2 cảm biến tia nhiệt, nó có 3 chân ra, một chân nối masse, một chân nối với nguồn volt DC, mức áp làm việc có thể từ 3 đến 15V. Góc dò lớn. Để tăng độ nhậy cho đầu dò, Bạn dùng kính Fresnel, nó được thiết kế cho loại đầu có 2 cảm biến, góc dò lớn, có tác dụng ngăn tia tử ngoại.

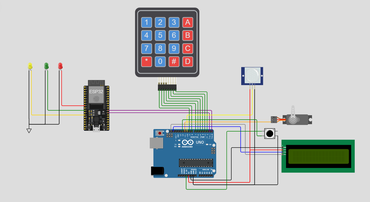
Ảnh có chứa văn bản, biểu đồ, Kế hoạch, hàng

Mô tả được tạo tự động

#### Hình 3.8: Sơ đồ nguyên lý chung

## 3.3 Thi công.

### 3.3.1 Mạch in.



#### Hình 3.9: Sơ đồ mạch in

### 3.3.2 Mạch hoàn chỉnh

A computer and electronics on a table

Description automatically generated

#### Hình 3.10: Mạch hoàn chỉnh

### 3.3.3 Code chương trình hệ nhúng

#### 3.3.3a Code ESP32

1. //code chương trình esp32:
2. #include <Arduino.h>
3. #include <WiFi.h>
4. #include <WiFiClient.h>
5. //#include <ESP8266WiFi.h>
6. //#include "FirebaseESP8266.h"
7. #include "FirebaseESP32.h"
8. #include <ArduinoJson.h>
9. #include <LiquidCrystal\_I2C.h>
10. #include <Wire.h>
11. #include "RTClib.h"
12. #include "addons/TokenHelper.h"
13. #include "addons/RTDBHelper.h"
14. // #include<SoftwareSerial.h> //Included SoftwareSerial Library
15. // //Started SoftwareSerial at RX and TX pin of ESP8266/NodeMCU
16. // //SoftwareSerial s(3, 1);
17. #define WIFI\_SSID "Dam Giang T5"
18. #define WIFI\_PASSWORD "17181921"
19. // #define WIFI\_SSID "P1202"
20. // #define WIFI\_PASSWORD "88888888"
21. // #define WIFI\_SSID "Bunn"
22. // #define WIFI\_PASSWORD "20032022"
23. #define FIREBASE\_HOST "https://henhung01-default-rtdb.firebaseio.com/"  //Thay bằng địa chỉ firebase của bạn
24. #define FIREBASE\_AUTH "AIzaSyAgXEgWLu3swzvs0q6LHSVbYwukdBiqthw"         //projec setting > service account > database secrets
25. #define USER\_EMAIL "ptn@gmail.com"
26. #define USER\_PASSWORD "01102003"
27. // #define fan D1
28. FirebaseData firebaseData;
29. FirebaseAuth auth;
30. FirebaseConfig config;
31. RTC\_DS1307 RTC;
32. void setup() {
33. Serial.begin(115200);
34. Serial2.begin(9600);
35. //s.begin(9600);
36. // Két nối wifi.
37. WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);
38. Serial.print("connecting");
39. while (WiFi.status() != WL\_CONNECTED) {
40. Serial.print(".");
41. delay(500);
42. }
43. Serial.println();
44. Serial.print("connected: ");
45. Serial.println(WiFi.localIP());
46. Wire.begin();
47. Wire.beginTransmission(0x68);  // địa chỉ của ds1307
48. Wire.write(0x07);              //
49. Wire.write(0x10);              //
50. Wire.endTransmission();
51. RTC.begin();
52. config.api\_key = FIREBASE\_AUTH;
53. auth.user.email = USER\_EMAIL;
54. auth.user.password = USER\_PASSWORD;
55. config.database\_url = FIREBASE\_HOST;
56. Firebase.reconnectWiFi(true);
57. firebaseData.setResponseSize(4096);
58. config.token\_status\_callback = tokenStatusCallback;
59. config.max\_token\_generation\_retry = 5;
60. Firebase.begin(&config, &auth);
61. pinMode(12, OUTPUT);
62. pinMode(27, OUTPUT);
63. pinMode(25, OUTPUT);
64. }
65. bool checkdoor = false;
67. void loop() {
68. Firebase.getInt(firebaseData, "/led");
69. int changed = firebaseData.intData();
70. if (changed) {
71. Firebase.getString(firebaseData, "/password");
72. String passValue = firebaseData.stringData();
73. char charArray[passValue.length() + 1];
74. passValue.toCharArray(charArray, passValue.length() + 1);
75. Serial2.write(charArray);
76. delay(500);
77. int data = Serial2.read();
78. if (data) {
79. Firebase.setInt(firebaseData, "/led", 0);
80. }
81. }
82. Firebase.getBool(firebaseData, "/door");
83. bool door = firebaseData.boolData();
84. //Serial.println(door);
85. DateTime now = RTC.now();
86. String x = String(int(now.hour()));
87. x += ':';
88. x += String(int(now.minute()));
89. x += ':';
90. x += String(int(now.second()));
91. x += '-';
92. x += String(int(now.day()));
93. x += '/';
94. x += String(int(now.month()));
95. x += '/';
96. x += String(int(now.year()));
97. //Serial.println(x);
98. if (checkdoor == false) {
99. if (door == true) {
100. String pass = "ope";
101. Serial.println(checkdoor);
102. char charArray[pass.length() + 1];
103. pass.toCharArray(charArray, pass.length() + 1);
104. Serial2.write(charArray);
105. checkdoor = true;
106. Serial.println(checkdoor);
107. delay(2500);
108. //Firebase.pushString(firebaseData, "/open", x);
109. }
110. char data = Serial2.read();
111. Serial.println(data);
112. if (data == '5') {
113. Firebase.pushString(firebaseData, "/open", x);
114. } else if (data == '9') {
115. Firebase.pushString(firebaseData, "/close", x);
116. }
117. } else {
118. char data = Serial2.read();
119. Serial.println(data);
120. if (data == '9') {
121. checkdoor = false;
122. Firebase.setBool(firebaseData, "/door", false);
123. Firebase.pushString(firebaseData, "/close", x);
124. Serial.println(checkdoor);
125. }
126. }
127. Firebase.getBool(firebaseData, "/R1");
128. bool R1 = firebaseData.boolData();
129. if (R1) {
130. digitalWrite(12, HIGH);
131. } else {
132. digitalWrite(21, LOW);
133. }
134. Firebase.getBool(firebaseData, "/R2");
135. bool R2 = firebaseData.boolData();
136. if (R2) {
137. digitalWrite(27, HIGH);
138. } else {
139. digitalWrite(27, LOW);
140. }
141. Firebase.getBool(firebaseData, "/R3");
142. bool R3 = firebaseData.boolData();
143. if (R3) {
144. digitalWrite(25, HIGH);
145. } else {
146. digitalWrite(25, LOW);
147. }
148. }

#### 3.3.3b Code chương trình Arduino

1. #include <Wire.h>
2. #include "RTClib.h"
3. #include <Keypad.h>
4. #include <LiquidCrystal\_I2C.h>
5. // #include <EEPROMWearLevel.h>
6. #include <EEPROM.h>
7. #include <Servo.h>
8. // EEPROMWearLevel eeprom;
9. LiquidCrystal\_I2C lcd(0x27, 16, 2);
10. const byte rows = 4;     //số hàng
11. const byte columns = 4;  //số cột
12. int holdDelay = 700;  //Thời gian trễ để xem là nhấn 1 nút nhằm tránh nhiễu
13. int n = 3;            //
14. int state = 0;        //nếu state =0 ko nhấn,state =1 nhấn thời gian nhỏ , state = 2 nhấn giữ lâu
15. char key = 0;
16. //Định nghĩa các giá trị trả về
17. char keys[rows][columns] = {
18. { '1', '2', '3', 'A' },
19. { '4', '5', '6', 'B' },
20. { '7', '8', '9', 'C' },
21. { '\*', '0', '#', 'D' },
22. };
23. byte rowPins[rows] = { 2, 3, 4, 5 };  //Cách nối chân với Arduino
24. byte columnPins[columns] = { 6, 7, 8, 9 };
25. //cài đặt thư viện keypad
26. Keypad keypad = Keypad(makeKeymap(keys), rowPins, columnPins, rows, columns);
27. int LDR = 10;
28. Servo sv;
29. String pass;
30. void writeToEEPROM(String data, int address, int length) {
31. for (int i = 0; i < length; i++) {
32. EEPROM.write(address + i, data[i]);
33. }
34. }
35. String readFromEEPROM(int address, int length) {
36. String result = "";
37. for (int i = 0; i < length; i++) {
38. char ch = EEPROM.read(address + i);
39. result += ch;
40. }
41. return result;
42. }
43. void setup() {
44. Serial.begin(9600);  //bật serial, baudrate 9600
45. String existingData = readFromEEPROM(0, 8);
46. if (existingData.length() == 0) {
47. // Nếu chưa có dữ liệu, lưu chuỗi vào EEPROM
48. writeToEEPROM("12345678", 0, 8);
49. //Serial.println("Data written to EEPROM");
50. pass = "12345678";
51. } else {
52. // Nếu có dữ liệu, in ra Serial Monitor
53. //Serial.println("Existing data in EEPROM: " + existingData);
54. pass = existingData;
55. }
56. pinMode(12, INPUT\_PULLUP);
57. pinMode(LDR, INPUT);
58. sv.attach(11);
59. sv.write(180);
60. lcd.init();
61. lcd.backlight();   //đèn nền bật
62. lcd.begin(16, 2);  // cài đặt số cột và số dòng
63. // in logo lên màn hình
64. lcd.print("nhom 4");
65. lcd.setCursor(0, 1);
66. lcd.print("PHENIKAA");
67. delay(2500);
68. lcd.clear();
69. }
70. String str = "";
71. String new\_pass = "";
72. //String passWord = "1234";
73. bool close = true;
74. bool open = false;
75. bool new\_password = false;
76. String data = "";
77. bool closed = true;
78. int ldr;
79. int time = 0;
80. int vitri;
81. String hidden = "";
82. int x = 0;
83. bool checkdoor = false;
84. void loop() {
85. if (digitalRead(12) == 1) {
86. if (open == false) {
87. lcd.clear();
88. lcd.setCursor(0, 0);
89. lcd.print("Opening!");
90. for (vitri = 180; vitri > 0; vitri--) {
91. sv.write(vitri);
92. delay(15);
93. }
94. close = false;
95. open = true;
96. closed = false;
97. checkdoor = true;
98. Serial.write('5');
99. delay(2500);
100. lcd.clear();
101. delay(500);
102. }
103. }
104. if (close) {
105. lcd.setCursor(0, 0);
106. lcd.print("Enter Password:");
107. char temp = keypad.getKey();
108. if ((int)keypad.getState() == PRESSED) {
109. if ((char)temp != 'A' && (char)temp != 'B' && (char)temp != 'C' && (char)temp != 'D' && temp != 0) {
110. str += temp;
111. hidden += "\*";
112. lcd.setCursor(x++, 1);
113. lcd.print(temp);
114. delay(250);
115. lcd.setCursor(0, 1);
116. lcd.print(hidden);
117. //Serial.println(str);
118. if (str.length() == 8 && str == pass) {
119. delay(600);
120. lcd.clear();
121. lcd.setCursor(0, 0);
122. lcd.print("Opening!");
123. for (vitri = 180; vitri > 0; vitri--) {
124. sv.write(vitri);
125. delay(15);
126. }
127. close = false;
128. open = true;
129. closed = false;
130. checkdoor = true;
131. Serial.write('5');
132. str = "";
133. hidden = "";
134. x = 0;
135. delay(2500);
136. lcd.clear();
137. } else if (str.length() == 8 && str != pass) {
138. delay(500);
139. lcd.clear();
140. lcd.setCursor(0, 0);
141. lcd.print("incorrect");
142. lcd.setCursor(0, 1);
143. lcd.print("Password!");
144. str = "";
145. hidden = "";
146. x = 0;
147. delay(2000);
148. lcd.clear();
149. }
150. }
151. if (temp == 'C' && temp != 0) {
152. if (str.length() > 0) {
153. str.remove(str.length() - 1);
154. hidden.remove(hidden.length() - 1);
155. x--;
156. lcd.clear();
157. }
158. lcd.setCursor(0, 1);
159. lcd.print(hidden);
160. //Serial.println(str);
161. }
162. if (temp == 'D' && temp != 0) {
163. str = "";
164. new\_password = true;
165. close = false;
166. lcd.clear();
167. }
168. }
169. }
170. while (open) {
171. lcd.setCursor(0, 0);
172. lcd.print("not closed!");
173. if (time < 50) {
174. time += 1;
175. //Serial.println(time);
176. delay(100);
177. if (digitalRead(LDR)) {
178. time = 0;
179. }
180. }
181. if (time == 50) {
182. for (vitri = 0; vitri < 180; vitri++) {
183. int ldr1 = digitalRead(LDR);
184. sv.write(vitri);
185. delay(20);
186. if (ldr1) {
187. time = 0;
188. for (int j = vitri; j >= 0; j--) {
189. sv.write(j);
190. //Serial.println(j);
191. delay(15);
192. }
193. break;
194. }
195. }
196. if (vitri == 180) {
197. open = false;
198. close = true;
199. time = 0;
200. lcd.clear();
201. lcd.setCursor(0, 0);
202. lcd.print("closed!");
203. if (checkdoor){
204. Serial.write('9');
205. checkdoor = false;
206. }
208. delay(2000);
209. lcd.clear();
210. }
211. }
212. }
213. if (new\_password) {
214. //lcd.clear();
215. lcd.setCursor(0, 0);
216. lcd.print("New Password:");
217. char temp = keypad.getKey();
218. if ((int)keypad.getState() == PRESSED) {
219. if ((char)temp != 'A' && (char)temp != 'B' && (char)temp != 'C' && (char)temp != 'B' && temp != 0) {
220. new\_pass += temp;
221. lcd.setCursor(0, 1);
222. lcd.print(new\_pass);
223. //Serial.println(new\_pass);
224. }
225. if (temp == 'B' && temp != 0) {
226. new\_password = false;
227. close = true;
228. new\_pass = "";
229. lcd.clear();
230. }
231. if (temp == 'C' && temp != 0) {
232. if (new\_pass.length() > 0) {
233. new\_pass.remove(new\_pass.length() - 1);
234. lcd.clear();
235. }
236. lcd.setCursor(0, 1);
237. lcd.print(new\_pass);
238. //Serial.println(new\_pass);
239. }
240. if (temp == 'A' && temp != 0 && new\_pass.length() == 8) {
241. writeToEEPROM(new\_pass, 0, 8);
242. pass = new\_pass;
243. //Serial.println("Data written to EEPROM");
244. //Serial.println(readFromEEPROM(0, 8));
245. delay(500);
246. lcd.clear();
247. lcd.setCursor(0, 0);
248. lcd.print("changed!");
249. new\_pass = "";
250. delay(2500);
251. lcd.clear();
252. }
253. }
254. delay(100);
255. }
256. while (Serial.available() > 0) {
257. char receivedChar = Serial.read();
258. data += receivedChar;
259. }
260. if (!data.equals("")) {
261. if (data.length() == 8) {
262. //Serial.println("New Password: " + data);
263. lcd.clear();
264. lcd.println("changed pass!");
265. delay(500);
266. lcd.clear();
267. writeToEEPROM(data, 0, 8);
268. pass = data;
269. Serial.write(1);
270. data = "";
271. } else if (data.equals("ope")){
272. lcd.clear();
273. lcd.setCursor(0, 0);
274. lcd.print("Opening!");
275. for (vitri = 180; vitri > 0; vitri--) {
276. sv.write(vitri);
277. delay(15);
278. }
279. close = false;
280. open = true;
281. closed = false;
282. checkdoor = true;
283. delay(2500);
284. lcd.clear();
285. delay(500);
286. //Serial.println(data);
287. data = "";
288. //Serial.println(data);
290. }
291. }
292. }

#### 3.3.3c Code chương trình flutter

*import* 'dart:async';  
*import* 'dart:io';

*import* 'package:firebase\_core/firebase\_core.dart';  
*import* 'package:firebase\_database/firebase\_database.dart';  
*import* 'package:flutter/material.dart';  
  
*void* initializeFirebase() *async* {  
 WidgetsFlutterBinding.*ensureInitialized*();  
 Platform.*isAndroid* ? *await* Firebase.*initializeApp*(  
 options: *const* FirebaseOptions(  
 apiKey: 'AIzaSyAgXEgWLu3swzvs0q6LHSVbYwukdBiqthw',  
 appId: '1:948180596215:android:cd402baeb7d8ec0bbbaaf2',  
 messagingSenderId: '948180596215',  
 projectId: 'henhung01')  
 ) : *await* Firebase.*initializeApp*();  
}  
  
*class* MyHomePage *extends* StatefulWidget {  
  
 @override  
 \_MyHomePage createState() => \_MyHomePage();  
}  
*class* \_MyHomePage *extends* State<MyHomePage> {  
 *final* DatabaseReference \_database = FirebaseDatabase.*instance*.reference();  
 *static late* StreamController<bool> *\_doorStreamController* = StreamController<bool>();  
 *static late* StreamController<bool> *\_lr1StreamController* = StreamController<bool>();  
 *static late* StreamController<bool> *\_lr2StreamController* = StreamController<bool>();  
 *static late* StreamController<bool> *\_lr3StreamController* = StreamController<bool>();  
 *static late* StreamController<bool> *\_nlStreamController* = StreamController<bool>();  
 *static late* StreamController<List<String>> *\_openStreamController* = StreamController<List<String>>();  
 *static* Stream<List<String>> *get openStream* => *\_openStreamController*.stream;  
  
 *static late* StreamController<List<String>> *\_closeStreamController* = StreamController<List<String>>();  
 *static* Stream<List<String>> *get closeStream* => *\_closeStreamController*.stream;  
 *static* Stream<bool>? *get doorStream* => *\_doorStreamController*.stream;  
 *static* Stream<bool>? *get lr1Stream* => *\_lr1StreamController*.stream;  
 *static* Stream<bool>? *get lr2Stream* => *\_lr2StreamController*.stream;  
 *static* Stream<bool>? *get lr3Stream* => *\_lr3StreamController*.stream;  
 *static* Stream<bool>? *get nlStream* => *\_nlStreamController*.stream;  
 *late* DatabaseReference \_ledReference;  
 *late* DatabaseReference \_passReference;  
 *late* DatabaseReference \_doorReference;  
 *late* DatabaseReference \_lr1Reference;  
 *late* DatabaseReference \_lr2Reference;  
 *late* DatabaseReference \_lr3Reference;  
 *late* DatabaseReference \_nlReference;  
 *late* DatabaseReference \_openReference;  
 *late* DatabaseReference \_closeReference;  
  
 bool doorValue = *false*;  
 bool lr1 = *false*;  
 bool lr2 = *false*;  
 bool lr3= *false*;  
 bool nl = *false*;  
 int \_ledValue = 0;  
 String \_passValue = '';  
 List<String> historyClose = [];  
 List<String> historyOpen = [];  
 DateTime? tO;  
 DateTime? tC;  
 int dem1=0;  
 int dem2=0;  
 *void* listenToPassReference() {  
 \_passReference.onValue.listen((DatabaseEvent event) {  
 *// Truy cập snapshot từ event* DataSnapshot snapshot = event.snapshot;  
  
 *if* (snapshot.value != *null*) {  
 \_passValue = snapshot.value.toString();  
 } *else* {  
 print('Không tìm thấy giá trị password trong cơ sở dữ liệu.');  
 }  
 });  
 }  
 *void* \_addTimeToCloseHistory(String time) {  
 DatabaseReference closeRef = \_database.child('close');  
 closeRef.push().set(time);  
 }  
 *void* \_addTimeOpenToHistory(String time) {  
 DatabaseReference openRef = \_database.child('open');  
 openRef.push().set(time);  
 }  
  
  
  
 @override  
 *void* initState() {  
 *super*.initState();  
 \_ledReference = FirebaseDatabase.*instance*.reference().child('led');  
 \_doorReference = FirebaseDatabase.*instance*.reference().child('door');  
 \_passReference = FirebaseDatabase.*instance*.reference().child('password');  
 \_lr1Reference = FirebaseDatabase.*instance*.reference().child('R1');  
 \_lr2Reference = FirebaseDatabase.*instance*.reference().child('R2');  
 \_lr3Reference = FirebaseDatabase.*instance*.reference().child('R3');  
 \_nlReference = FirebaseDatabase.*instance*.reference().child('NongLanh');  
 \_openReference = FirebaseDatabase.*instance*.reference().child('open');  
 \_closeReference = FirebaseDatabase.*instance*.reference().child('close');  
  
 *\_openStreamController*??= StreamController<List<String>>.broadcast();  
 *\_closeStreamController*??= StreamController<List<String>>.broadcast();  
 *\_closeStreamController*??= StreamController<List<String>>.broadcast();  
 *\_lr1StreamController*??= StreamController<bool>.broadcast();  
 *\_doorStreamController*??= StreamController<bool>.broadcast();  
 *\_lr2StreamController*??= StreamController<bool>.broadcast();  
 *\_lr3StreamController*??= StreamController<bool>.broadcast();  
 *\_nlStreamController*??= StreamController<bool>.broadcast();  
  
  
 *// Lắng nghe sự thay đổi của door* \_doorReference.onValue.listen((DatabaseEvent event) {  
 DataSnapshot snapshots = event.snapshot;  
 doorValue = snapshots.value *as* bool;  
 bool newdoorValue = doorValue;  
 *if* (event.snapshot.value != *null*) {  
 doorValue = event.snapshot.value *as* bool;  
 *// Đưa giá trị mới vào stream  
 \_doorStreamController*.add(doorValue);  
  
 *if* (doorValue) {  
 DateTime currentTimeOpen = DateTime.now();  
 int year = currentTimeOpen.year;  
 int month = currentTimeOpen.month;  
 int day = currentTimeOpen.day;  
 int hour = currentTimeOpen.hour;  
 int minute = currentTimeOpen.minute;  
 int second = currentTimeOpen.second;  
 \_addTimeOpenToHistory("$hour:$minute:$second-$day/$month/$year");  
 }  
  
 *else* {  
 DateTime currentTimeClose = DateTime.now();  
 int year = currentTimeClose.year;  
 int month = currentTimeClose.month;  
 int day = currentTimeClose.day;  
 int hour = currentTimeClose.hour;  
 int minute = currentTimeClose.minute;  
 int second = currentTimeClose.second;  
 \_addTimeToCloseHistory("$hour:$minute:$second-$day/$month/$year");  
 }  
 } *else* {  
 print('Không tìm thấy giá trị door trong cơ sở dữ liệu.');  
 }  
 });  
  
 *//pass* \_passReference.onValue.listen((DatabaseEvent event) {  
 DataSnapshot snapshots = event.snapshot;  
 \_passValue = snapshots.value.toString();  
 *if* (event.snapshot.value != *null*) {  
 \_passValue = event.snapshot.value.toString();  
 } *else* {  
 print('Không tìm thấy giá trị password trong cơ sở dữ liệu.');  
 }  
 });  
 *//lr1* \_lr1Reference.onValue.listen((DatabaseEvent event) {  
 DataSnapshot snapshots = event.snapshot;  
 lr1 = snapshots.value *as* bool;  
 *if* (event.snapshot.value != *null*) {  
 lr1 = event.snapshot.value *as* bool;  
 *\_lr1StreamController*.add(lr1);  
 } *else* {  
 print('Không tìm thấy giá trị lr1 trong cơ sở dữ liệu.');  
 }  
 });  
 *//pass* \_lr2Reference.onValue.listen((DatabaseEvent event) {  
 DataSnapshot snapshots = event.snapshot;  
 lr2 = snapshots.value *as* bool;  
 *if* (event.snapshot.value != *null*) {  
 lr2 = event.snapshot.value *as* bool;  
 *\_lr2StreamController*.add(lr2);  
 } *else* {  
 print('Không tìm thấy giá trị lr2 trong cơ sở dữ liệu.');  
 }  
 });  
 *//pass* \_lr3Reference.onValue.listen((DatabaseEvent event) {  
 DataSnapshot snapshots = event.snapshot;  
 lr3 = snapshots.value *as* bool;  
 *if* (event.snapshot.value != *null*) {  
 lr3 = event.snapshot.value *as* bool;  
 *\_lr3StreamController*.add(lr3);  
 } *else* {  
 print('Không tìm thấy giá trị lr3 trong cơ sở dữ liệu.');  
 }  
 });  
 *//pass* \_nlReference.onValue.listen((DatabaseEvent event) {  
 DataSnapshot snapshots = event.snapshot;  
 nl = snapshots.value *as* bool;  
 *if* (event.snapshot.value != *null*) {  
 nl = event.snapshot.value *as* bool;  
 } *else* {  
 print('Không tìm thấy giá trị nl trong cơ sở dữ liệu.');  
 }  
 });  
 *//history open* \_openReference.onValue.listen((event) {  
 DataSnapshot snapshot = event.snapshot;  
 *if* (snapshot.value != *null*) {  
 Map<*dynamic*, *dynamic*> values = snapshot.value *as* Map<*dynamic*, *dynamic*>;  
 *if* (values != *null*) {  
 List<String> newHistoryOpen = [];  
 values.forEach((key, value) {  
 *if* (value *is* String) {  
 String time = value;  
 newHistoryOpen.add(time);  
 } *else* {  
 print('Giá trị không phải là kiểu String ${value}');  
 }  
 });  
 historyOpen.clear();  
 *// Thêm các phần tử mới vào danh sách* historyOpen.addAll(newHistoryOpen);  
  
 *// Cập nhật Stream để thông báo rằng có dữ liệu mới  
 \_openStreamController*.add(historyOpen);  
 *// \_historyStreamController.add(historyOpen);* setState(() {}); *// Cập nhật UI khi dữ liệu đã được tải* } *else* {  
 print('Giá trị snapshot không phải là một Map<dynamic, dynamic>');  
 }  
 } *else* {  
 print('Giá trị snapshot là null');  
 }  
 });  
 *//history close* \_closeReference.onValue.listen((event) {  
 DataSnapshot snapshot = event.snapshot;  
 *if* (snapshot.value != *null*) {  
 Map<*dynamic*, *dynamic*> values = snapshot.value *as* Map<*dynamic*, *dynamic*>;  
  
 *if* (values != *null*) {  
 List<String> newHistoryClose = [];  
 values.forEach((key, value) {  
 *if* (value *is* String) {  
 String time = value;  
 newHistoryClose.add(time);  
 } *else* {  
 print('Giá trị không phải là kiểu String ${value}');  
 }  
 });;  
 historyClose.clear();  
 *// Thêm các phần tử mới vào danh sách* historyClose.addAll(newHistoryClose);  
  
 *// Cập nhật Stream để thông báo rằng có dữ liệu mới  
 \_closeStreamController*.add(historyClose);  
 setState(() {}); *// Cập nhật UI khi dữ liệu đã được tải* } *else* {  
 print('Giá trị snapshot không phải là một Map<dynamic, dynamic>');  
 }  
 } *else* {  
 print('Giá trị snapshot là null');  
 }  
 });  
 }  
  
 @override  
 *void* dispose() {  
 *// \_openStreamController.close();  
 // \_closeStreamController.close();  
 \_lr1StreamController*.close();  
 *\_doorStreamController*.close();  
 *\_lr2StreamController*.close();  
 *\_lr3StreamController*.close();  
 *\_nlStreamController*.close();  
 *super*.dispose();  
 }  
  
 *final* passoldController = TextEditingController();  
 *final* passnewdController = TextEditingController();  
 *final* passnewd1Controller = TextEditingController();  
  
 bool containsNonNumericCharacter(String input) {  
 *// Sử dụng biểu thức chính quy để kiểm tra xem chuỗi có chứa ký tự khác số không* RegExp regExp = RegExp(r'[^0-9]');  
 *return* regExp.hasMatch(input);  
 }  
 bool checkPass() {  
 *if* (passnewdController.text.length != 8 || containsNonNumericCharacter(passnewdController.text)) *return false*;  
  
 *if* (\_passValue != passoldController.text) *return false*;  
 *if* (passnewd1Controller.text != passnewdController.text) *return false*;  
  
 *return true*;  
 }  
  
 @override  
 *void* openshowdialog(BuildContext context) {  
 showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* AlertDialog(  
 title: Text('Đổi mật khẩu'),  
 content: Container(  
 height: 300,  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Card(  
 elevation: 5,  
 child: Container(  
 padding: EdgeInsets.all(10),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.end,  
 children: [  
 TextField(  
 decoration: InputDecoration(labelText: 'Mật khẩu cũ'),  
 controller: passoldController,  
 ),  
 TextField(  
 decoration: InputDecoration(labelText: 'Mật khẩu mới'),  
 controller: passnewdController,  
 obscureText: *true*,  
 ),  
 TextField(  
 decoration: InputDecoration(labelText: 'Nhập lại mật khẩu mới'),  
 controller: passnewd1Controller,  
 obscureText: *true*,  
 ),  
 ],  
 ),  
 ),  
 ),  
   
 SizedBox(height: 20),  
 ElevatedButton(  
 onPressed: () {  
 *if*(checkPass()) {  
 String newPassValue = passnewd1Controller.text;  
 \_passReference.set(newPassValue);  
 \_ledReference.set(1);  
 *// setState(() {  
 // // \_ledValue = newLedValue;  
 // \_passValue = newPassValue;  
 // });* showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* AlertDialog(  
 title: Text('Đổi mật khẩu thành công'),  
 content: Text('Mật khẩu đã được cập nhập!'),  
 actions: [  
 TextButton(  
 onPressed: () {  
  
 Navigator.*pop*(context);  
 },  
 child: Text('Đóng'),  
 ),  
 ],  
 );  
 },  
 );  
 passoldController.clear();  
 passnewdController.clear();  
 passnewd1Controller.clear();  
 } *else* {  
 showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* AlertDialog(  
 title: Text('Lỗi'),  
 content: Text('Mật khẩu cũ hoặc Mật khẩu mới không trùng khớp. Vui lòng thử lại'),  
 actions: [  
 TextButton(  
 onPressed: () {  
  
 Navigator.*pop*(context);  
 },  
 child: Text('Đóng'),  
 ),  
 ],  
 );  
 },  
 );  
  
 passoldController.clear();  
 passnewdController.clear();  
 passnewd1Controller.clear();  
 }  
 },  
 child: Text('Đổi mật khẩu'),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
 );  
 }  
 *void* \_openDoor() {  
 \_database.update({'door': *true*});  
 }  
 @override  
 Widget timeDoorOpen(BuildContext context) {  
 *if* (historyOpen.isEmpty) {  
 *return* Text('Không có dữ liệu');  
 } *else* {  
 List<Widget> widgetList = historyClose.map((time) {  
 *return* ListTile(  
 title: Text('Thời gian đóng: ${time}'),  
 );  
 }).toList();  
  
 *return* SingleChildScrollView(  
 child: Column(  
 children: widgetList,  
 ),  
 );  
 }  
 }  
 @override  
 Widget timeDoorClose(BuildContext context) {  
 *if* (historyClose.isEmpty) {  
 *return* Text('Không có dữ liệu');  
 } *else* {  
 List<Widget> widgetList = historyClose.map((time) {  
 *return* ListTile(  
 title: Text('Thời gian đóng: ${time}'),  
 );  
 }).toList();  
  
 *return* SingleChildScrollView(  
 child: Column(  
 children: widgetList,  
 ),  
 );  
 }  
 }  
 *void* timeD(BuildContext context){  
 showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* AlertDialog(  
 title: Text('Lịch sử đóng mở cửa'),  
 content: Container(  
 height: 500,  
 child: SingleChildScrollView(  
 controller: ScrollController(),  
 child: Column(  
 children: [  
 TextButton(onPressed: (){  
 showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* AlertDialog(  
 title: Text('Lịch sử mở cửa'),  
 content: Container(  
 height: 500,  
 child: SingleChildScrollView(  
 child: Column(  
 children: [  
 timeDoorOpen(context)  
 ],  
 ),  
 ),  
 ),  
 actions: [  
 TextButton(  
 onPressed: () {  
  
 Navigator.*of*(context).pop(); *// Đóng hộp thoại khi nhấn nút* },  
 child: Text('Đóng'),  
 ),  
 ],  
 );  
 }  
 );  
  
 },  
 child: Text('Lịch sử mở cửa') ),  
 TextButton(onPressed: (){  
 showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* AlertDialog(  
 title: Text('Lịch sử đóng cửa'),  
 content: Container(  
 height: 500,  
 child: SingleChildScrollView(  
 child: Column(  
 children: [  
 timeDoorClose(context)  
 ],  
 ),  
 ),  
 ),  
 actions: [  
 TextButton(  
 onPressed: () {  
  
 Navigator.*of*(context).pop(); *// Đóng hộp thoại khi nhấn nút* },  
 child: Text('Đóng'),  
 ),  
 ],  
 );  
 }  
 );  
  
 }, child: Text('Lịch sử đóng cửa')),  
  
 *// Add other functions if needed  
 // timeDoorClose(context),* ],  
 ),  
 ),  
 ),  
 actions: [  
 TextButton(  
 onPressed: () {  
  
 Navigator.*of*(context).pop(); *// Đóng hộp thoại khi nhấn nút* },  
 child: Text('Đóng'),  
 ),  
 ],  
 );  
 }  
 );  
 }  
  
 *void* \_doorManager (BuildContext context) {  
 ValueNotifier<bool> selectedDoorStatus = ValueNotifier<bool>(doorValue);  
  
 showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* AlertDialog(  
 title: Text('Quản lý cửa'),  
 content: Container(  
 height: 300,  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 children: [  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 children: [  
 Text('Trạng thái cửa'),  
 ValueListenableBuilder<bool>(  
 valueListenable: selectedDoorStatus,  
 builder: (context, value, child) {  
 *return* ToggleButtons(  
 children: [  
 Text(  
 'Mở',  
 style: TextStyle(color: value ? Colors.*blue* : Colors.*grey*),  
 ),  
 Text(  
 'Đóng',  
 style: TextStyle(color: value ? Colors.*grey* : Colors.*red*),  
 ),  
 ],  
 isSelected: [value, !value],  
 onPressed: (int index) {  
 selectedDoorStatus.value = index == 0;  
 },  
 );  
 },  
 )  
  
 ],  
 ),  
 *// Text('${historyOpen.toString()} + ${historyOpen.toString()}'),* TextButton(  
 onPressed: () {  
 openshowdialog(context);  
 },  
 child: Text('Đổi mật khẩu')  
 ),  
 TextButton(  
 onPressed: () {  
 timeD(context);  
 },  
 child: Text('Lịch sử đóng mở cửa')  
 ),  
 ],  
 ),  
 ),  
 actions: [  
 TextButton(  
 onPressed: () {  
 \_database.update({'door': selectedDoorStatus.value});  
 *\_doorStreamController*.add(selectedDoorStatus.value);  
  
 Navigator.*of*(context).pop(); *// Đóng hộp thoại khi nhấn nút* },  
 child: Text('Xác nhận'),  
 ),  
  
 ],  
 );  
 }  
 );  
 }  
 *void* ledManager(BuildContext context) {  
 ValueNotifier<bool> selectedDoorStatus1 = ValueNotifier<bool>(lr1);  
 ValueNotifier<bool> selectedDoorStatus2 = ValueNotifier<bool>(lr2);  
 ValueNotifier<bool> selectedDoorStatus3 = ValueNotifier<bool>(lr3);  
 showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* AlertDialog(  
 title: Text('Trạng thái đèn'),  
 content: Container(  
 height: 500,  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 children: [  
 Text('Đèn phòng 1'),  
 ValueListenableBuilder<bool>(  
 valueListenable: selectedDoorStatus1,  
 builder: (context, value, child) {  
 *return* ToggleButtons(  
 children: [  
 Text(  
 'Mở',  
 style: TextStyle(color: value ? Colors.*blue* : Colors.*grey*),  
 ),  
 Text(  
 'Đóng',  
 style: TextStyle(color: value ? Colors.*grey* : Colors.*red*),  
 ),  
 ],  
 isSelected: [value, !value],  
 onPressed: (int index) {  
 selectedDoorStatus1.value = index == 0;  
 },  
 );  
 },  
 )  
 ],  
 ),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 children: [  
 Text('Đèn phòng 2'),  
 ValueListenableBuilder<bool>(  
 valueListenable: selectedDoorStatus2,  
 builder: (context, value, child) {  
 *return* ToggleButtons(  
 children: [  
 Text(  
 'Mở',  
 style: TextStyle(color: value ? Colors.*blue* : Colors.*grey*),  
 ),  
 Text(  
 'Đóng',  
 style: TextStyle(color: value ? Colors.*grey* : Colors.*red*),  
 ),  
 ],  
 isSelected: [value, !value],  
 onPressed: (int index) {  
 selectedDoorStatus2.value = index == 0;  
 },  
 );  
 },  
 )  
 ],  
 ),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 children: [  
 Text('Đèn phòng 3'),  
 ValueListenableBuilder<bool>(  
 valueListenable: selectedDoorStatus3,  
 builder: (context, value, child) {  
 *return* ToggleButtons(  
 children: [  
 Text(  
 'Mở',  
 style: TextStyle(color: value ? Colors.*blue* : Colors.*grey*),  
 ),  
 Text(  
 'Đóng',  
 style: TextStyle(color: value ? Colors.*grey* : Colors.*red*),  
 ),  
 ],  
 isSelected: [value, !value],  
 onPressed: (int index) {  
 selectedDoorStatus3.value = index == 0;  
 },  
 );  
 },  
 )  
 ],  
 )  
 ],  
 ),  
 ),  
 actions: [  
 TextButton(  
 onPressed: () {  
 \_database.update({'R1': selectedDoorStatus1.value});  
 \_database.update({'R2': selectedDoorStatus2.value});  
 \_database.update({'R3': selectedDoorStatus3.value});  
  
 Navigator.*of*(context).pop(); *// Đóng hộp thoại khi nhấn nút* },  
 child: Text('Xác nhận'),  
 ),  
 ],  
 );  
 }  
 );  
 }  
 *void* nlManager(BuildContext context) {  
 ValueNotifier<bool> selectedDoorStatus = ValueNotifier<bool>(nl);  
 showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 *return* AlertDialog(  
 title: Text('Trạng thái nóng lạnh'),  
 content: Container(  
 height: 500,  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceBetween,  
 children: [  
 Text('Nóng Lạnh'),  
 ValueListenableBuilder<bool>(  
 valueListenable: selectedDoorStatus,  
 builder: (context, value, child) {  
 *return* ToggleButtons(  
 children: [  
 Text(  
 'Mở',  
 style: TextStyle(color: value ? Colors.*blue* : Colors.*grey*),  
 ),  
 Text(  
 'Đóng',  
 style: TextStyle(color: value ? Colors.*grey* : Colors.*red*),  
 ),  
 ],  
 isSelected: [value, !value],  
 onPressed: (int index) {  
 selectedDoorStatus.value = index == 0;  
 },  
 );  
 },  
 )  
 ],  
 )  
 ],  
 ),  
 ),  
 actions: [  
 TextButton(  
 onPressed: () {  
 \_database.update({'NongLanh': selectedDoorStatus.value});  
  
 *// \_nlStreamController.add(selectedDoorStatus.value);* Navigator.*of*(context).pop(); *// Đóng hộp thoại khi nhấn nút* },  
 child: Text('Xác nhận'),  
 ),  
 ],  
 );  
 }  
 );  
 }  
 @override  
 Widget build(BuildContext context) {  
 *return* Scaffold(  
 appBar: AppBar(  
 title: Text('Smart Home',  
 style: TextStyle(  
 fontWeight: FontWeight.*bold*,  
 color: Colors.*white*,  
 fontSize: 30,  
 )  
 ),  
 backgroundColor: Colors.*blueAccent*,  
 ),  
 body: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 SizedBox(height: 100),  
 Row(  
 children: <Widget>[  
 Padding(padding: EdgeInsets.all(20.0)),  
 ElevatedButton(  
 onPressed: () {  
 \_doorManager(context);  
 *// Navigator.of(context).popAndPushNamed("/pass");* },  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 Icon(  
 Icons.*door\_sliding*,  
 color: Colors.*lightBlue*,  
 size: 50,  
 ),  
 Text('Cửa',  
 style: TextStyle(  
 color: Colors.*black*,  
 ),  
 )  
 ],  
 )  
 ),  
 ElevatedButton(  
 onPressed: () {  
 ledManager(context);  
  
 *// Navigator.of(context).popAndPushNamed("/pass");* },  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 Icon(  
 Icons.*key*,  
 color: Colors.*lightBlue*,  
 size: 50,  
 ),  
 Text('Đèn',  
 style: TextStyle(  
 color: Colors.*black*,  
 ),  
 )  
 ],  
 )  
 ),  
 ElevatedButton(  
 onPressed: () {  
 nlManager(context);  
  
 *// Navigator.of(context).popAndPushNamed("/pass");* },  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 Icon(  
 Icons.*hot\_tub*,  
 color: Colors.*lightBlue*,  
 size: 50,  
 ),  
 Text('Nóng lạnh',  
 style: TextStyle(  
 color: Colors.*black*,  
 ),  
 )  
 ],  
 )  
 ),  
 ],  
 )  
 ],  
 ),  
 );  
 }  
}

# CHƯƠNG IV: KẾT QUẢ THỰC HIỆN VÀ KẾT LUẬN

## Ưu, nhược điểm của sản phẩm

* + - Ưu điểm:
      * Dễ dàng sử dụng.
      * Giá thành rẻ, dễ thi công , dễ sửa chữa , độ an toàn cao.
      * Độ chính xác cao.
    - Nhược điểm:.
      * Bố trí linh kiện chưa được khoa học.

## Kết luận.

Trong quá trình nghiên cứu và xây dựng đề tài em đã vận dụng những kiến thức chuyên môn. nhằm để hoàn thành yêu cầu đặt ra. Qua đó em được dịp cũng cố lại kiến thức cơ bản chuyên ngành. Trên cơ sơ đó chúng em có thể đánh giá lại những gì mình có được sau khi học.

Trong thời gian thực hiện đề tài với sự chỉ bảo và giúp đỡ tận tình của giảng viên hướng dẫn đến nay hệ thống mở khóa cửa thông minh & điều khiển các thiết bị trong nhà đã hoàn thành.Tuy nhiên do thời gian có hạn và trình độ chuyên môn còn hạn chế nên đồ án còn tồn tại những thiếu sót. Nhóm rất mong nhận được những ý kiến và góp ý của thầy để sản phẩm của nhóm được hoàn thiện hơn. Cuối cùng nhóm em xin được cảm ơn thầy đã tạo điều kiện tốt nhất giúp đỡ nhóm em hoàn thành đề tài.

Nhóm 4 xin chân thành cảm ơn!

**Tiếng Việt**

# TÀI LIỆU THAM KHẢO

[1]. https://khuenguyencreator.com/tong-quan-ve-so-do-chan-esp32-va-ngoai-vi/

[2]. <http://arduino.vn/bai-viet/42-arduino-uno-r3-la-gi>