

**MEMBER**

Vo Nhat Truong - JK-ENR-HA-11560  
Tran Ngoc Thang - JK-ENR-HA-11564   
Nguyen Bach Thong - JK-ENR-HA-11556

**8051 project**

**DOOR LOCK SYSTEM**

FPT JETKING UNIVERSITY

**MỤC LỤC**

[PHẦN 1: GIỚI THIỆU CHỦ ĐỀ 3](#_Toc178655799)

[1. Giới thiệu tổng quan: 3](#_Toc178655800)

[Tổng quan về hệ thống: 3](#_Toc178655801)

[2. Chức năng và hoạt động: 4](#_Toc178655802)

[*a.* *Các phương thức mở/khóa cửa:* 4](#_Toc178655803)

[*b.* *Đổi mật khẩu* *(Dành cho chủ nhà):* 5](#_Toc178655804)

[*c.* *Quản lý từ xa với nền tảng Internet of Things (IoT) Blynk:* 5](#_Toc178655805)

[PHẦN 2: GIỚI THIỆU THIẾT BỊ VÀ PHẦN MỀM: 6](#_Toc178655806)

[1. THÀNH PHẦN THIẾT BỊ PHẦN CỨNG: 6](#_Toc178655807)

[I. BỘ KIT PRO 8051: 6](#_Toc178655808)

[*II.* *EEPROM 24C02:* 7](#_Toc178655812)

[*III.* *SERVO MOTOR DC:* 7](#_Toc178655813)

[*IV.* *Vi điều khiển ESP32:* 8](#_Toc178655814)

[*V.* *Module RC522:* 9](#_Toc178655815)

[2. Phần mềm và công cụ: 9](#_Toc178655817)

[Phần mềm Keil C: 9](#_Toc178655818)

[Nền tảng IoT Blynk: 10](#_Toc178655820)

[PHẦN 3: QUÁ TRÌNH THIẾT KẾ 11](#_Toc178655823)

[1. Sơ đồ kết nối: 11](#_Toc178655824)

[2. Lắp đặt mô hình: 12](#_Toc178655825)

[3. Flowchart và code: 13](#_Toc178655826)

[PHẦN 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN: 13](#_Toc178655827)

[1. Kết luận: 13](#_Toc178655828)

[2. Ưu điểm: 13](#_Toc178655829)

[3. Nhược điểm: 14](#_Toc178655830)

[4. Hướng phát triển: 14](#_Toc178655831)

**PHẦN 1: GIỚI THIỆU CHỦ ĐỀ**

1. **Giới thiệu tổng quan:**



**DOOR LOCK SYSTEM**

Mô hình hệ thống cửa tự động bao gồm hai chức năng chính: **mở/khóa cửa** và **đổi mật khẩu**, kết hợp với ứng dụng **Blynk** để quản lý và điều khiển từ xa.

### Tổng quan về hệ thống:

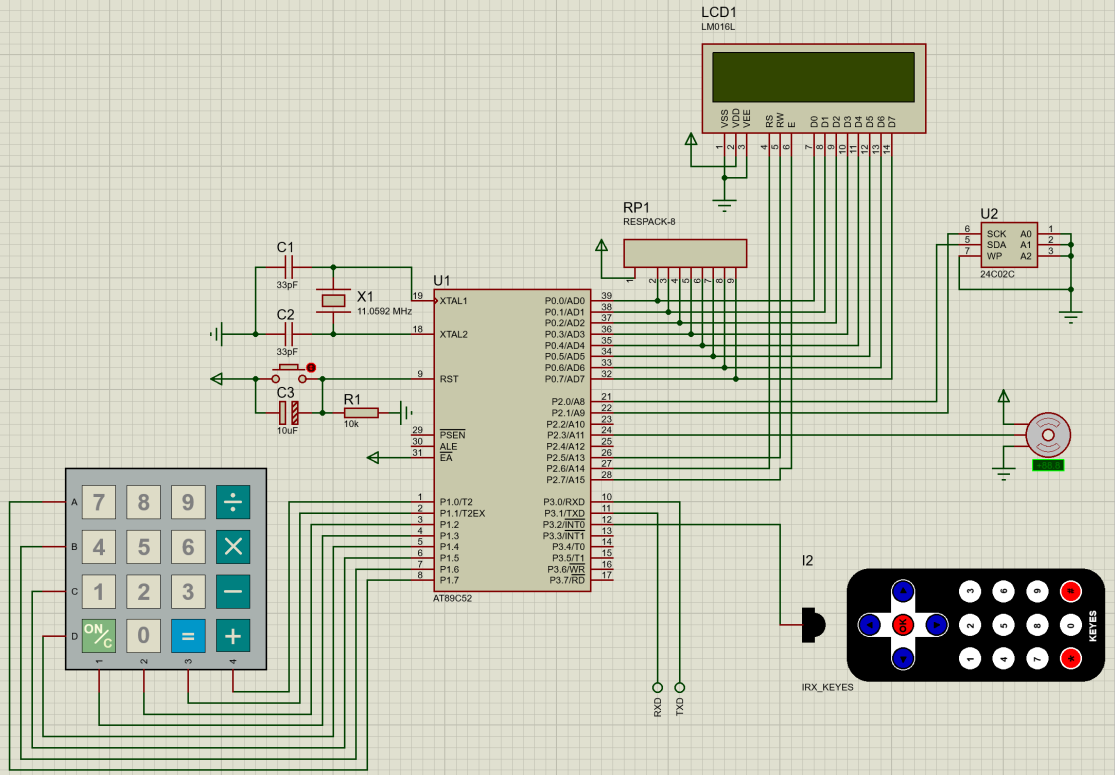
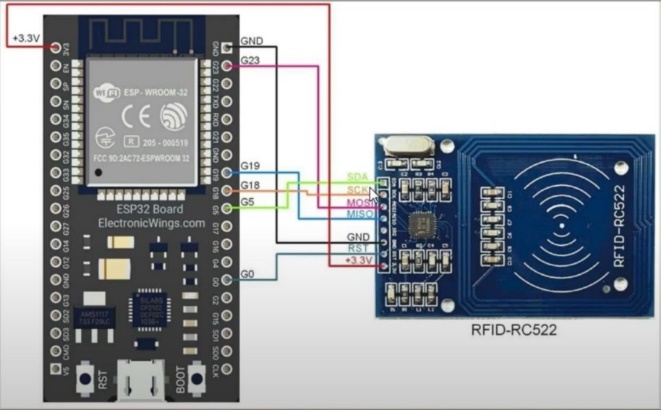
1. **Chức năng mở và khóa cửa**:
   * Hệ thống có thể được mở hoặc khóa thông qua các phương thức an toàn như sử dụng **thẻ từ và bàn phím số**.
   * Cơ chế này đảm bảo tính bảo mật cao, chỉ cho phép những người có quyền truy cập mở cửa.
2. **Chức năng đổi mật khẩu**:

* Chỉ có **người chủ** mới có thể thực hiện thao tác đổi mật khẩu thông qua **remote**.
* Điều này đảm bảo rằng người dùng khác không thể thay đổi quyền truy cập, tăng cường mức độ kiểm soát và bảo mật cho hệ thống.

1. **Ứng dụng Blynk**:
   * **Blynk** là công cụ kết nối giữa thiết bị phần cứng và điện thoại thông minh, cho phép người dùng điều khiển và theo dõi trạng thái cửa từ bất kỳ đâu.
   * Ứng dụng này cung cấp một giao diện trực quan, dễ sử dụng, cho phép bạn thực hiện các tác vụ như mở/khóa cửa, nhận thông báo khi có người truy cập, và quản lý cài đặt mật khẩu.

1. **Chức năng và hoạt động:**

Hệ thống cửa tự động có 2 chức năng chính: **các phương thức để mở/khóa cửa, đổi mật khẩu (dành cho chủ nhà)** và **quản lý từ xa với nền tảng IoT (Internet Of Things) Blynk.**

1. ***Các phương thức mở/khóa cửa:***
2. **Thẻ RFID (Radio Frequency Identification):** là một loại thẻ sử dụng công nghệ nhận dạng tần số vô tuyến để truyền dữ liệu từ thẻ đến đầu đọc thông qua sóng vô tuyến:
   * + Đầu tiên, viết code để lập trình cho Chip 8051 kết nối với ESP32 [**như sau**](https://github.com/truong92cdv/DoorLockSystem/blob/main/KeilC/uart.c) và 8051 sẽ kết nối với ESP32 thông qua giao thức UART.
     + Sau đó trên chip 8051, nối dây từ chân P3.0/RXD đến chân TXD, chân P3.1/TXD đến chân RXP của ESP32. Kế tiếp kết nối ESP32 với Module RC522 thông qua giao thức SPI.
     + Viết code lập trình cho ESP32 kết nối với RC522 [**như sau**](https://github.com/truong92cdv/DoorLockSystem/blob/main/ESP32/src/main.cpp)**.**
     + Tiếp theo, nối chân P3.2 của 8051 vào Servo Motor *(như Hình 1. Sơ đồ lắp đặt)* và viết code lập trình cho 8051 kết nối với servo Motor [**như sau**](https://github.com/truong92cdv/DoorLockSystem/blob/main/KeilC/servo.h)**.**

* Khi đầu đọc thẻ Module RC522 nhận dữ liệu từ thẻ RFID thì tín hiệu sẽ được truyền tải đến chip 8051, từ đó kích hoạt Servo Motor ở trạng thái “mở cửa”.

1. **Phím bấm (Keypad):** Người dùng có thể nhập mã số trên một bàn phím để mở khóa. Mã số này sẽ được kiểm tra với cơ sở dữ liệu trong hệ thống. Nếu mã số đúng, khóa sẽ được mở:
   * + **Viết Code lập trình khởi tạo mật khẩu trên phím bấm** [**như sau**](https://github.com/truong92cdv/DoorLockSystem/blob/main/KeilC/keypad.c)**.**

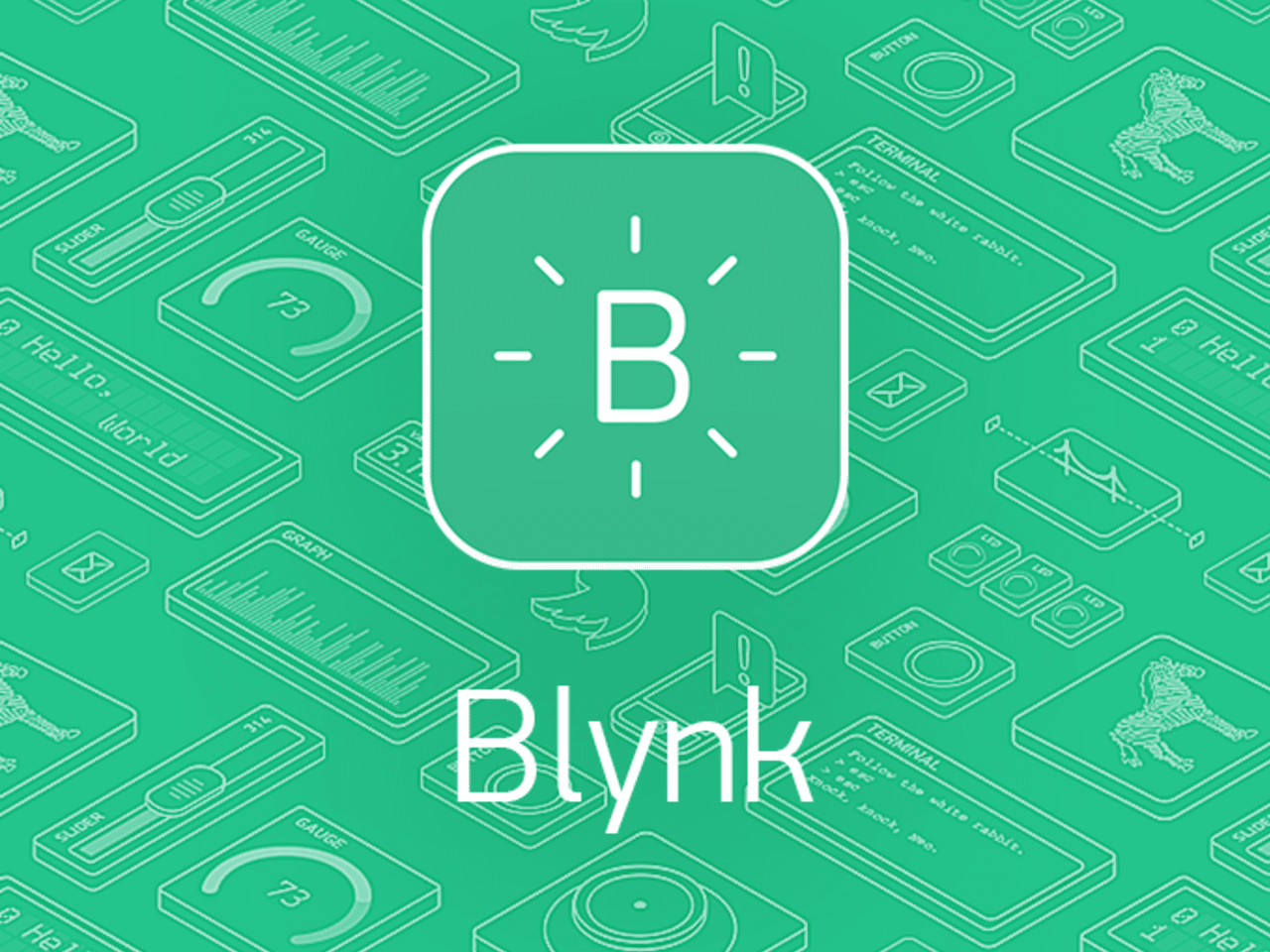
* Khởi tạo mật khẩu trên phím bấm, nếu đúng mật khẩu thì kích hoạt Servo Motor ở trạng thái “mở cửa”.

1. ***Đổi mật khẩu* *(Dành cho chủ nhà):***

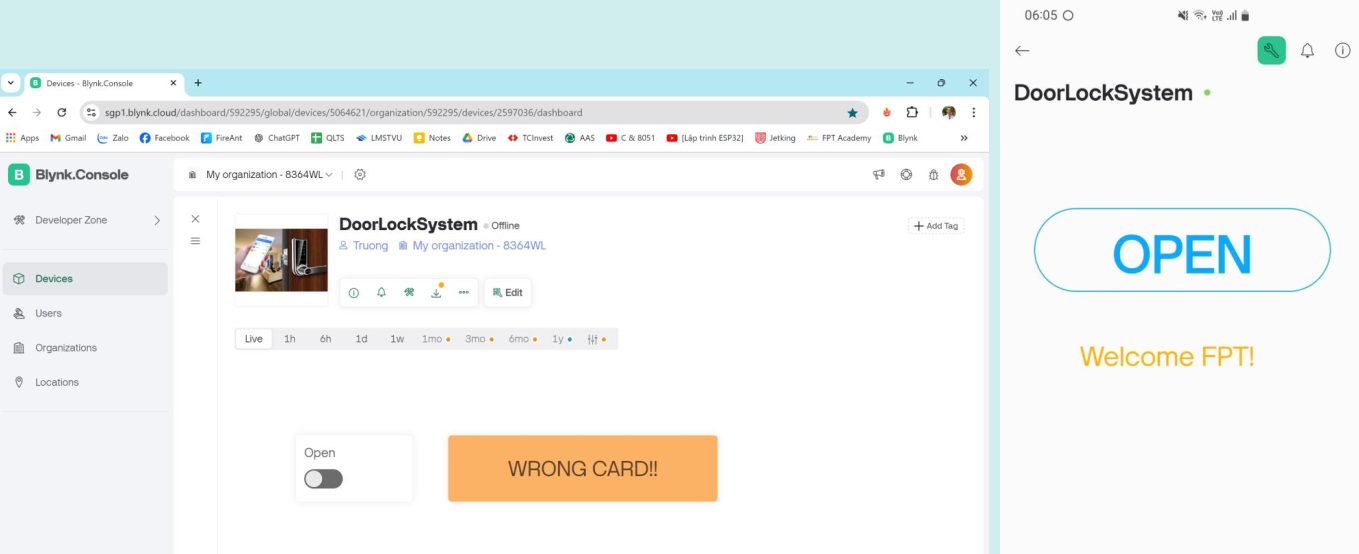
**Trong đồ án này, sử dụng Remote điều khiển là 1 công cụ để thay đổi mật khẩu của cửa, và Remote này sẽ được sử dụng bởi người chủ.**

* + - **Remote hồng ngoại điều khiển:** Đây là phương thức mở khóa từ xa bằng cách sử dụng một thiết bị điều khiển từ xa. Remote thường sử dụng sóng hồng ngoại để gửi tín hiệu đến hệ thống khóa:
    - **Trong đồ án này, sử dụng Remote điều khiển là 1 công cụ để thay đổi mật khẩu của cửa, và Remote này sẽ được sử dụng bởi người chủ.**
    - Trên chip 8051, dùng P3.2 kết nối với Remote điều khiển (*như hình 1*), code lập trình [**như sau**](https://github.com/truong92cdv/DoorLockSystem/blob/main/KeilC/ir.h)**.**
* Để thay đổi mật khẩu thì nhập “\*AB#” trên Remote điều khiển, sau đó nhập mật khẩu mới trên KeyPad.

1. ***Quản lý từ xa với nền tảng Internet of Things (IoT) Blynk:***



* + - Blynk giúp dễ dàng xây dựng các dự án IoT và điều khiển thiết bị từ xa. Nó cung cấp một ứng dụng di động (có sẵn trên cả iOS và Android) để điều khiển và giám sát các thiết bị IoT từ xa thông qua mạng internet.
* Khi có người sử dụng các phương thức mở cửa, kết quả sẽ được hiển thị trên Blynk. Nhờ vậy, chúng ta sẽ quản lý được từ xa.



* + - Lập trình code cho ESP32 để kết nối với Blynk [**như sau**](https://github.com/truong92cdv/DoorLockSystem/blob/main/ESP32/src/main.cpp)**.**

**PHẦN 2: GIỚI THIỆU THIẾT BỊ VÀ PHẦN MỀM:**

1. **THÀNH PHẦN THIẾT BỊ PHẦN CỨNG:**
2. **BỘ KIT PRO 8051:**

### Kit học tập phát triển 8051 đa chức năng1. ****Cấu tạo của Bộ Kit Pro 8051:****

* **Vi điều khiển 8051:** Trung tâm của bộ kit là vi điều khiển 8051 hoặc một phiên bản tương đương như AT89S52, AT89C51.
* **Bộ nạp chương trình (Programmer):** Giúp nạp mã máy (hex file) vào vi điều khiển.
* **Mạch nguồn:** Cung cấp nguồn điện ổn định cho vi điều khiển và các linh kiện khác.
* **Các module ngoại vi:** Bộ kit thường đi kèm với các module ngoại vi như LED, LCD, bàn phím ma trận, relay, cảm biến nhiệt độ, động cơ servo, và các thiết bị ngoại vi khác để thực hành.

### 2. ****Các thành phần điển hình:****

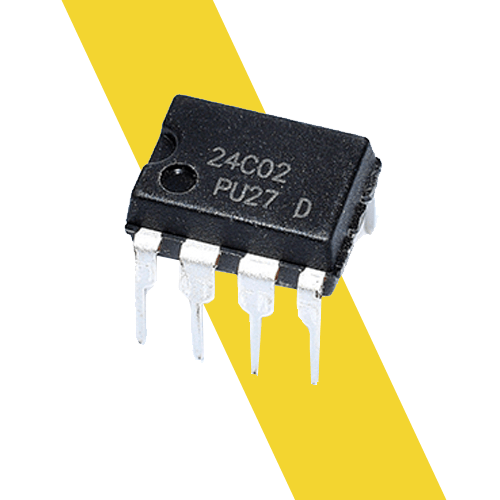
* **LED và Seven Segment Displays:** Để thực hành điều khiển hiển thị số và chữ số.
* **LCD 16x2:** Màn hình hiển thị để hiển thị thông tin từ vi điều khiển.
* **Keypad:** Bàn phím ma trận để nhập dữ liệu hoặc lệnh.
* **Relay và Motor Driver:** Để điều khiển các thiết bị điện hoặc động cơ.
* **Giao tiếp RS232/USB:** Để kết nối và truyền dữ liệu với máy tính.
* **EEPROM ngoài:** Để lưu trữ dữ liệu ngoài bộ nhớ của vi điều khiển.
* **Cổng mở rộng I/O:** Cho phép kết nối thêm các module ngoại vi khác.

### 3. ****Tính năng:****

* **Hỗ trợ học lập trình:** Bộ kit này được thiết kế để hỗ trợ người học lập trình vi điều khiển 8051, giúp họ hiểu rõ cách hoạt động của vi điều khiển, cách giao tiếp với các ngoại vi và cách xây dựng các ứng dụng thực tế.
* **Thực hành trực quan:** Với các module ngoại vi tích hợp sẵn, người dùng có thể thực hiện các bài tập thực hành như điều khiển LED, hiển thị thông tin lên LCD, giao tiếp với máy tính qua cổng serial, điều khiển động cơ, và nhiều ứng dụng khác.
* **Tài liệu và mã nguồn mẫu:** Thường đi kèm với tài liệu hướng dẫn và mã nguồn mẫu, giúp người dùng dễ dàng bắt đầu và phát triển các dự án của riêng mình.

1. ***EEPROM 24C02:***

EEPROM 24C02 là một loại **bộ nhớ EEPROM** (Electrically Erasable Programmable Read-Only Memory) có dung lượng 2 Kbit (256 byte), thường được sử dụng để lưu trữ dữ liệu mà vẫn giữ lại khi mất nguồn điện.

Dưới đây là một số thông tin quan trọng về **EEPROM 24C02**:

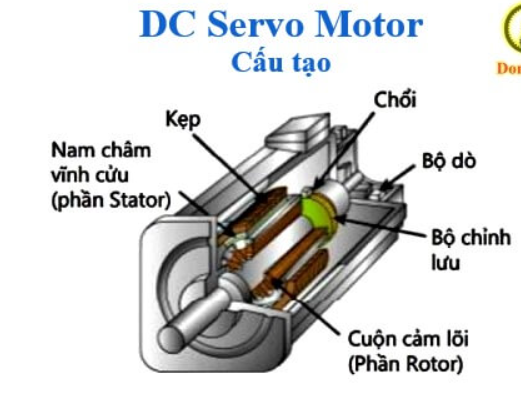
* **Giao tiếp I²C (Inter-Integrated Circuit)**: EEPROM 24C02 sử dụng giao tiếp I²C với hai dây chính là SDA (dữ liệu) và SCL (đồng hồ) để truyền nhận dữ liệu giữa bộ nhớ và vi điều khiển.
* **Dung lượng**: 24C02 có dung lượng 256 byte, thường được chia thành các trang (page) để dễ dàng ghi và đọc dữ liệu. Kích thước mỗi trang thường là 8 byte.
* **Tính năng**:
  + **Lưu trữ dữ liệu vĩnh viễn**: Dữ liệu trên EEPROM không bị mất khi mất nguồn điện, phù hợp để lưu trữ các cấu hình hoặc thông số mà cần được giữ lại sau khi thiết bị tắt.
  + **Ghi/xóa linh hoạt**: Mỗi byte trong EEPROM có thể được ghi/xóa riêng biệt mà không ảnh hưởng đến các byte khác.

1. ***SERVO MOTOR DC:***

Servo motor DC là một loại động cơ điện một chiều (DC) được sử dụng để kiểm soát chuyển động chính xác về vị trí, tốc độ và mô-men xoắn.

**Cấu Tạo Của Động Cơ SERVO DC**:

* Động cơ DC servo được cấu tạo từ 2 bộ phận chính là stator và rotor ngoài ra động cơ còn được tích hợp thêm một số bộ phận khác:
* Stator: phần đứng yên cấu tạo bởi nam châm vĩnh cửu



* Rotor: phần chuyển động cấu tạo bởi các cuộn dây quấn quanh lõi thép
* Cổ góp: cấu tạo bằng đồng, để dẫn điện từ chổi than vào cuộn dây
* Chổi tha: dẫn điện vào cổ góp
* Bộ dò: là bộ đọc xung encoder hoặc bộ cảm ứng điện từ
* Động cơ DC servo gồm 2 loại:
* Động cơ DC servo có chổi than:

Là loại động cơ sản xuất với chi phí thấp, cấu trúc đơn giản, dễ điều khiển và cũng dễ bảo trì bảo dưỡng.

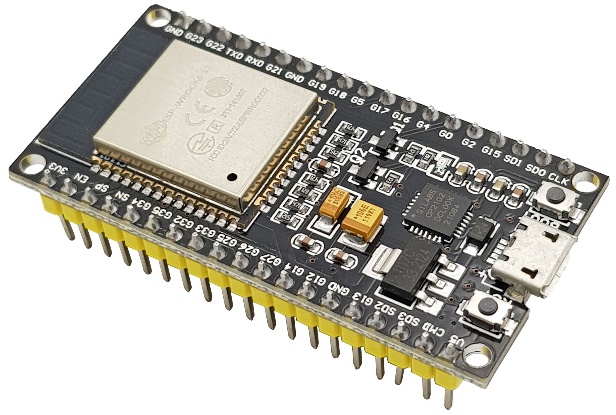
* Động cơ DC servo không chổi than:

Có kích thước nhỏ và trọng lượng nhỏ hơn. Tuy nhiên, loại động cơ này phả ứng rất nhanh, quán tính nhanh nhờ mô men xoắn ổn định hơn động cơ DC servo có chổi than. Nhờ chuyển mạch điện tử linh hoạt theo sóng vuông hoặc chuyển mạch sóng sin động cơ không chổi than hoạt động thông minh hơn.

Động cơ không chổi than có ưu điểm tiết kiệm năng lượng do bức xạ điện từ thấp, ít sinh nhiệt dẫn đến tuổi thọ động cơ cao hơn rất nhiều so với loại có chổi than.  
  
***Nguyên lý hoạt động của động cơ SERVO DC:***

* Dòng điện một chiều thông qua cổ góp và chổi than vào cuồn dây quấn quanh lõi thép tạo ra từ trường ngược vưới từ trường của nam châm vĩnh cửu trên Stator làm quay rotor, để thay đổi tốc độ cần thay đổi điện áp cấp vào cuộn dây
* Bộ phận dò có chức năng phản hồi tốc độ và vị trí của động cơ đến bộ điều khiển, bộ điều khiển so sánh với tín hiệu phản hồi với mứ đặt để điều chỉnh điện áp cấp vào động cơ
* Mạch điều khiển của động cơ được nối với tín hiệu ra. Điều này có nghĩa khi động cơ vận hành thì vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển. Lúc này nếu có bất kỳ lý do nào ngăn cản quá trình chuyển động quay của động cơ, bộ phận cơ cấu hồi tiếp sẽ nhận thấy tín hiệu và mạch điểu khiển sẽ phải tiếp tục điều chỉnh sai lệch cho động cơ đến khi đạt điểm chính xác.

1. ***Vi điều khiển ESP32:***

ESP32 là một vi điều khiển (microcontroller), nổi bật với khả năng tích hợp **Wi-Fi** và **Bluetooth**. Nó được sử dụng rộng rãi trong các dự án IoT (Internet of Things), tự động hóa và các ứng dụng nhúng khác nhờ hiệu suất cao, tiêu thụ năng lượng thấp và tính linh hoạt.

Một số đặc điểm chính của ESP32:

* **Wi-Fi và Bluetooth tích hợp**: Cho phép giao tiếp không dây với các thiết bị và mạng khác.
* **Đa lõi**: Có hai lõi xử lý, giúp tăng khả năng thực hiện các tác vụ song song.
* **Nhiều chân GPIO**: Hỗ trợ kết nối nhiều loại cảm biến, module và thiết bị ngoại vi khác.
* **ADC và DAC**: Tích hợp sẵn các bộ chuyển đổi analog/digital và digital/analog.
* **Tính năng tiết kiệm năng lượng**: Thích hợp cho các ứng dụng yêu cầu tiêu thụ năng lượng thấp, như các hệ thống IoT hoạt động bằng pin.

1. ***Module RC522:***

RC522 là một module RFID (Radio Frequency Identification) được sử dụng để đọc và ghi dữ liệu từ thẻ RFID.

### Các đặc điểm chính của RC522:

* **Tần số hoạt động**: 13.56 MHz, là tần số tiêu chuẩn cho các hệ thống RFID thông thường.
* **Giao tiếp**: Module RC522 có thể giao tiếp với vi điều khiển (như ESP32, Arduino) thông qua giao thức SPI, nhưng nó cũng hỗ trợ I2C và UART.
* **Phạm vi hoạt động**: Khoảng cách đọc thường trong phạm vi từ 0 đến 5 cm, phụ thuộc vào loại thẻ và anten.
* **Khả năng đọc/ghi**: RC522 có thể đọc thẻ RFID chuẩn ISO/IEC 14443 A/MIFARE và ghi dữ liệu lên một số loại thẻ nhất định.

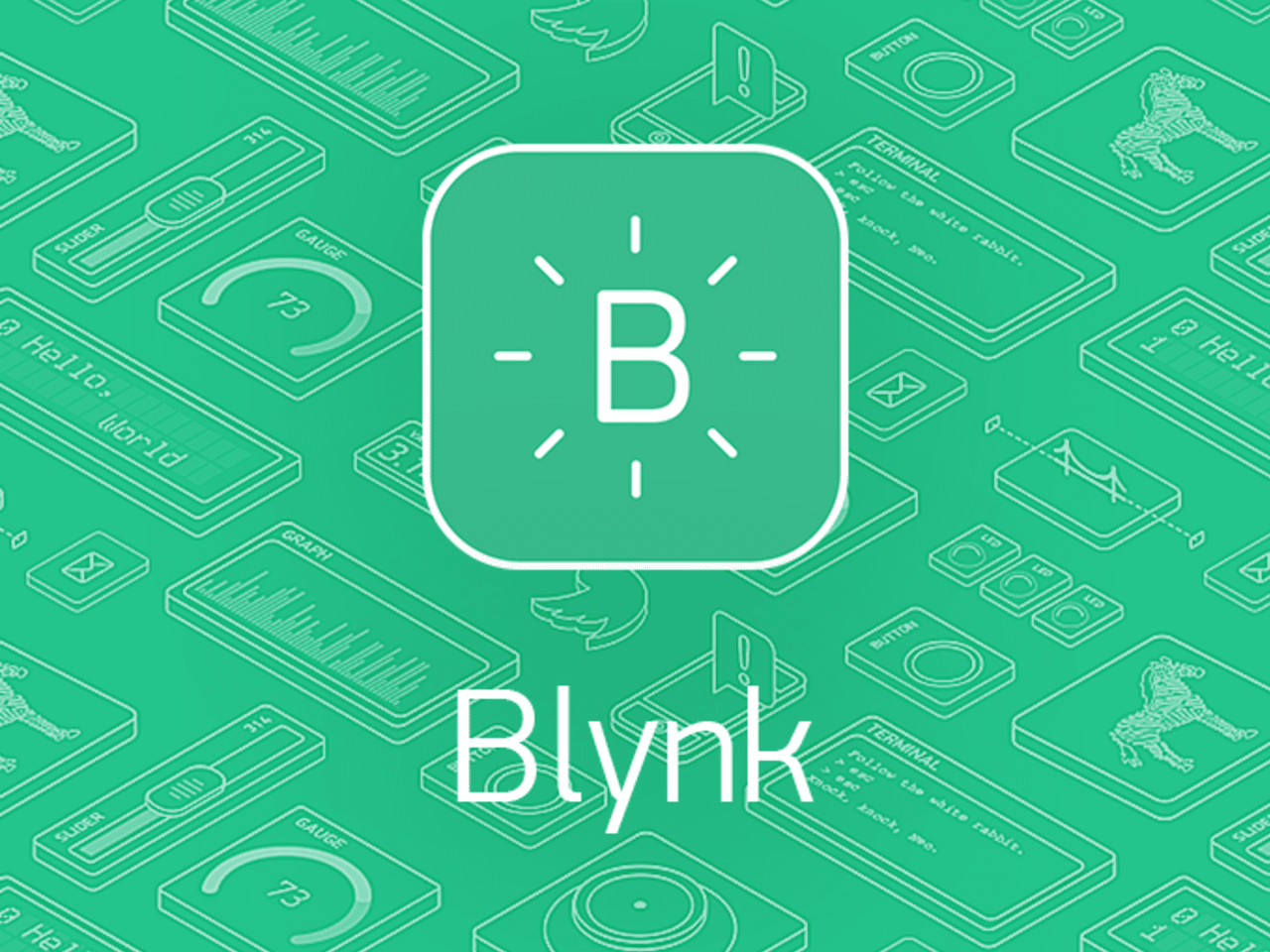
1. **Phần mềm và công cụ:**

* **Phần mềm Keil C:**

Keil C là một môi trường phát triển tích hợp (Integrated Development Environment - IDE) được sử dụng để lập trình các vi điều khiển, đặc biệt là họ vi điều khiển ARM và 8051. Keil cung cấp công cụ để viết mã, biên dịch và debug chương trình cho các vi điều khiển.

### Các tính năng chính của Keil C:

1. **IDE mạnh mẽ**: Hỗ trợ viết mã, biên dịch, và gỡ lỗi trực tiếp trong cùng một môi trường.
2. **Trình biên dịch C/C++**: Cung cấp trình biên dịch C chuẩn và tối ưu hóa cho các bộ vi điều khiển như ARM, 8051, và nhiều dòng khác.
3. **Thư viện phần cứng**: Hỗ trợ nhiều loại vi điều khiển và thư viện tương thích giúp lập trình viên dễ dàng tích hợp và điều khiển phần cứng.
4. **Symantec Debugger**: Cung cấp công cụ gỡ lỗi mạnh mẽ, hỗ trợ xem trực tiếp các thanh ghi, bộ nhớ, và giá trị biến trong quá trình chạy chương trình.
5. **RTX Real-Time Operating System (RTOS)**: Hỗ trợ hệ điều hành thời gian thực, giúp quản lý các tác vụ đồng thời trong hệ thống nhúng.

* **Nền tảng IoT Blynk:**

**Blynk** là một nền tảng IoT (Internet of Things) giúp kết nối và điều khiển các thiết bị thông qua Internet. Blynk cung cấp một giao diện đơn giản và dễ sử dụng để quản lý các thiết bị từ xa, thông qua ứng dụng trên điện thoại hoặc web. Nó đặc biệt hữu ích trong các dự án nhà thông minh, tự động hóa, và các hệ thống điều khiển từ xa như dự án hệ thống khóa cửa tự động mà bạn đang làm.

### Các thành phần chính của Blynk:

1. **Blynk App**: Ứng dụng di động (trên iOS và Android) cho phép người dùng tạo giao diện điều khiển tùy chỉnh để quản lý các thiết bị.
2. **Blynk Server**: Máy chủ Blynk chịu trách nhiệm giao tiếp giữa ứng dụng và các thiết bị IoT. Bạn có thể sử dụng máy chủ Blynk Cloud hoặc thiết lập máy chủ cá nhân.
3. **Blynk Library**: Thư viện phần mềm giúp kết nối thiết bị của bạn với Blynk Server. Các thiết bị như Arduino, ESP8266, ESP32,... đều có thể kết nối dễ dàng thông qua thư viện này.

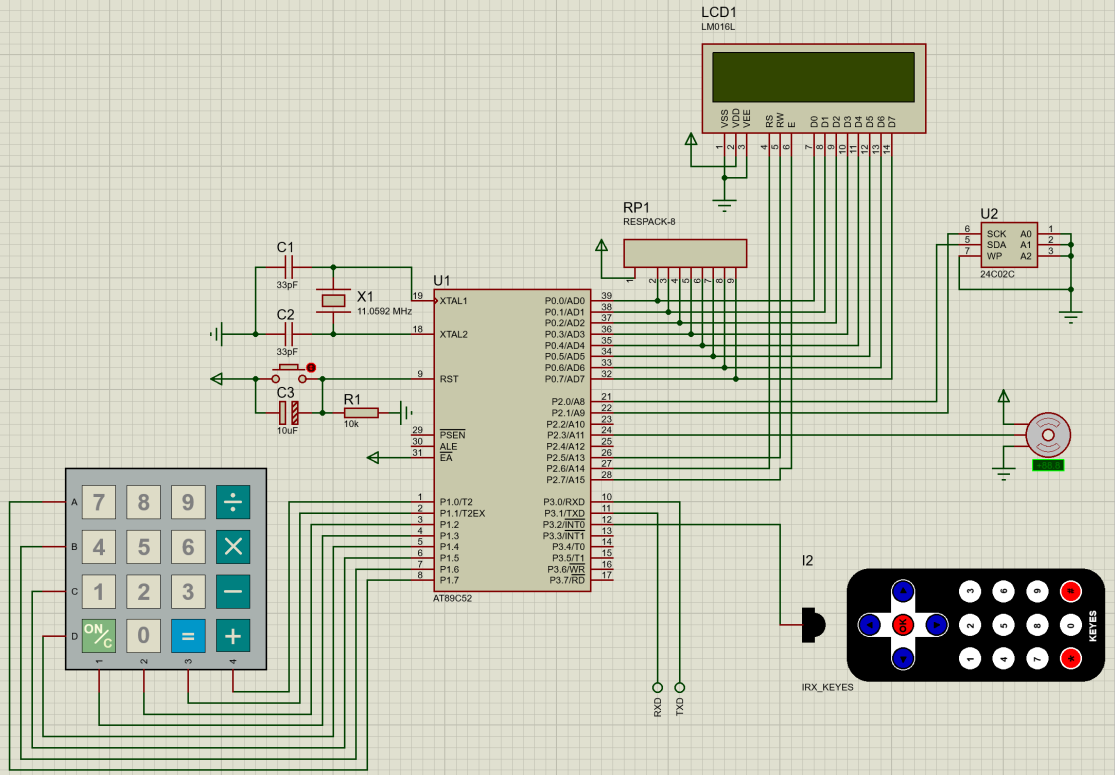
### Tính năng:

1. **Điều khiển từ xa**: Blynk cho phép điều khiển các thiết bị qua Internet, ví dụ bật/tắt thiết bị, khóa/mở cửa.
2. **Tạo giao diện tùy chỉnh**: Bạn có thể kéo thả các widget như nút bấm, thanh trượt, và màn hình hiển thị trong ứng dụng Blynk để tạo một bảng điều khiển phù hợp với nhu cầu của bạn.
3. **Quản lý nhiều thiết bị**: Blynk hỗ trợ kết nối và điều khiển nhiều thiết bị IoT trên một giao diện.
4. **Tích hợp IoT dễ dàng**: Blynk cung cấp API và các công cụ hỗ trợ giúp việc tích hợp thiết bị với nền tảng này trở nên dễ dàng.

Với hệ thống khóa cửa tự động của bạn, Blynk có thể là phương tiện để người chủ có thể dễ dàng quản lý việc khóa/mở cửa hoặc đổi mật khẩu từ xa thông qua ứng dụng trên điện thoại.

**PHẦN 3: QUÁ TRÌNH THIẾT KẾ**

* 1. **Sơ đồ kết nối:**



* + - Sử dụng Chip 8051:

+ P0.0 – P0.7: kết nối với D0 – D7 của màn hình LCD.

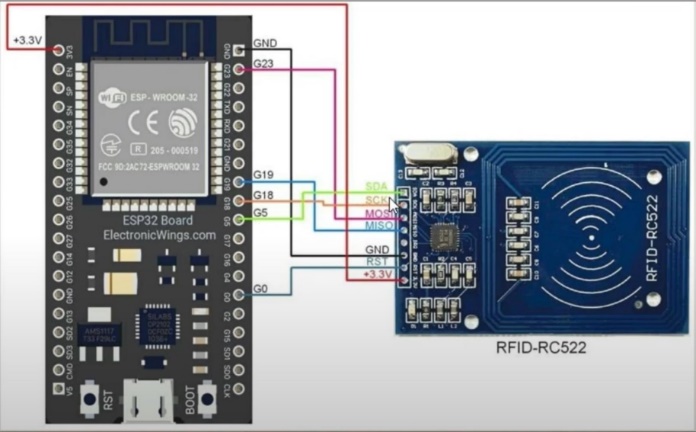
+ P1.0 – P1.7: kết nối với bàn phím.

+ P2.0 – P2.1: kết nối với bộ nhớ EEPROM 24C02.

+ P2.3: kết nối với Servo Motor.

+ P2.5 – P2.7: kết nối với các chân RS, RW, E của màn hình LCD.

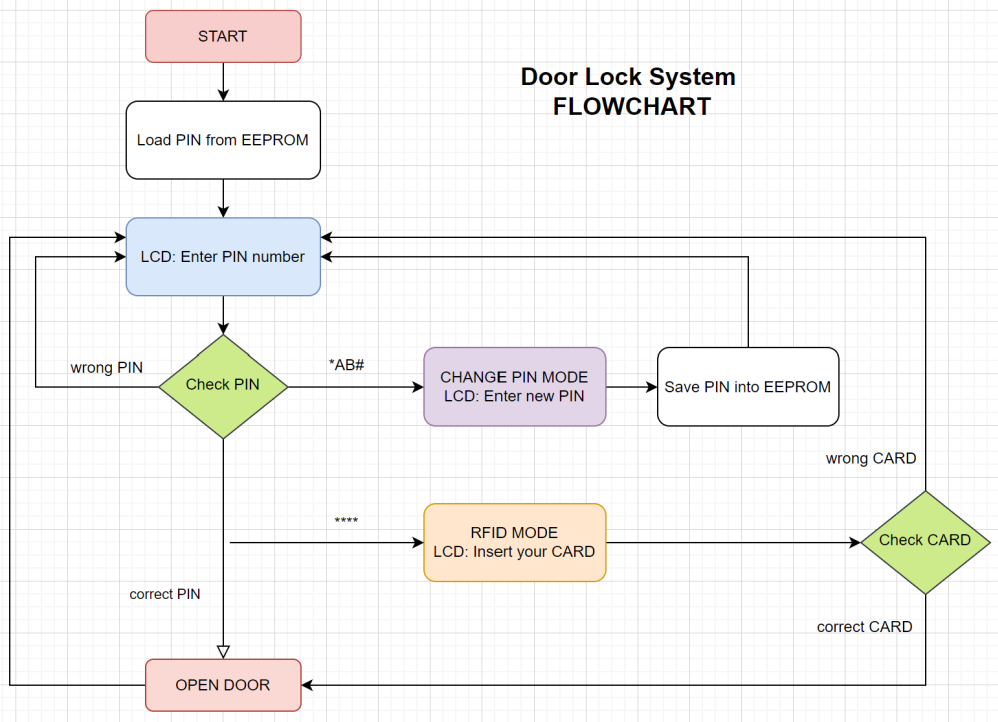
+ P3.2: kết nối với Remote điều khiển.



+ P3.0 và P3.1: kết nối với chân RXD và TXD của vi điều khiển ESP32.

* + - Vi điều khiển ESP32 kết nối với Module RC522.
  1. **Lắp đặt mô hình:**

****

* 1. **Flowchart và code:**
* Code [**tại đây**](https://github.com/truong92cdv/DoorLockSystem/blob/main/KeilC/ir.h).

**PHẦN 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN:**

* 1. **Kết luận:**

Mô hình hệ thống cửa tự động với hai chức năng chính là **mở/khóa cửa** và **đổi mật khẩu**, được tích hợp với nền tảng **Blynk** để quản lý từ xa, mang lại giải pháp toàn diện cho việc bảo mật và kiểm soát ra vào. Với khả năng điều khiển cửa bằng **thẻ từ, bàn phím số** cùng với việc đổi mật khẩu chỉ được thực hiện bởi **người chủ** thông qua **remote**, hệ thống đảm bảo sự tiện lợi và an toàn tối đa.

Việc quản lý và giám sát qua **Blynk** không chỉ giúp tăng cường khả năng kiểm soát từ xa mà còn mang lại sự linh hoạt cho người dùng. Đây là một giải pháp lý tưởng cho các ứng dụng **nhà thông minh** và các môi trường yêu cầu **bảo mật cao**, cung cấp sự an tâm và tiện lợi trong việc quản lý quyền truy cập.

* 1. **Ưu điểm:**
* **Bảo mật cao**: Chức năng đổi mật khẩu chỉ được thực hiện bởi người chủ qua remote, bảo đảm quyền kiểm soát tuyệt đối.
* **Tiện lợi và linh hoạt**: Người dùng có thể dễ dàng điều khiển hệ thống qua nhiều phương thức khác nhau và theo dõi từ xa qua Blynk.
* **An toàn và hiện đại**: Phù hợp với các ứng dụng nhà thông minh, văn phòng, hoặc các môi trường yêu cầu mức độ bảo mật và tiện ích cao.
  1. **Nhược điểm:**
* **Giới hạn về tính năng thay đổi mật khẩu**:
* Việc chỉ cho phép **người chủ** đổi mật khẩu bằng **remote** mang lại tính bảo mật cao, nhưng nếu **remote** bị mất hoặc hỏng, quá trình thay đổi mật khẩu có thể gặp khó khăn và mất nhiều thời gian để khôi phục.
* **Chi phí ban đầu cao**:
* Hệ thống cửa tự động tích hợp các chức năng hiện đại như thẻ từ, bàn phím, remote và sử dụng nền tảng Blynk có thể đòi hỏi chi phí ban đầu cao, bao gồm cả phần cứng và phần mềm.
* **Phụ thuộc vào nguồn điện**:
* Hệ thống cửa tự động thường cần nguồn điện ổn định để hoạt động. Nếu xảy ra mất điện mà không có nguồn điện dự phòng, cửa có thể không thể mở hoặc khóa được.
  1. **Hướng phát triển:**
* Phát triển thêm nhiều cách mở khóa: cảm biến vân tay, nhận diện khuôn mặt.
* Các chức năng mới như:

1. Thêm thẻ, xóa thẻ.
2. Hệ thống tra cứu lịch sử ra/vào cửa.
3. Báo động qua tin nhắn SMS.
4. Tích hợp nhà thông minh.