**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

**BỘ MÔN ĐIỆN TỬ**

---------------o0o---------------

****

**ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ KHÓA SỐ ĐIỆN TỬ**

**GVHD: Thầy NGUYỄN LÝ THIÊN TRƯỜNG**

**SVTH: Vương Minh Đức**

**MSSV: 1711097**

**TP. HỒ CHÍ MINH, THÁNG 07 NĂM 2020**

***LỜI CẢM ƠN***

Đề tài được hoàn thành đúng thời gian thời gian quy định, đầy đủ các phần và đạt được kết quả mong muốn không chỉ là nỗ lực của riêng em mà còn nhờ có sự hướng dẫn tận tình của thầy.

Em xin chân thành cảm ơn

* Thầy NGUYỄN LÝ THIÊN TRƯỜNG là người trực tiếp hướng dẫn, luôn đồng hành và giúp cho em tìm ra được sai sót, nhận biết rõ tình hình và vấn để nhờ đó kịp thời định hướng thiết kế sản phẩm nhanh nhất có thể để đạt được thành quả ngày hôm nay. Một lần nữa em xin gửi lời cám ơn trân trọng nhất đến thầy vì đã tận tuỵ với sự nghiệp trồng người.
* Chân thành cảm ơn các bạn sinh viên khác trong lớp cũng như cùng khoa Điện- điện tử đã hỗ trợ em rất nhiều về mặt phương tiện, sách vở, ý kiến,…

Trong suốt quá trình thực hiện đề tài, mặc dù em đã rất cố gắng, thiếu sót là điều không thể tránh khỏi. Rất mong nhận được sự góp ý, phê bình, chỉ dẫn của các thầy cô.

*Tp. Hồ Chí Minh, ngày 01 tháng 07 năm 2020*

**Sinh viên**

**TÓM TẮT ĐỒ ÁN**

Xã hôi ngày càng phát triển và văn minh. Chúng ta cũng đang từng bước tiến tới công nghệ 4.0. Sự phát triển của kỹ thuật điện tử đã tạo ra hàng loạt những thiết bị với các đặc điểm nổi bật như sự chính xác cao, tốc độ nhanh, gọn nhẹ…góp phần cho hoạt động của con người dần dần hoàn thiện theo thời gian. Điện tử - đang trở thành một ngành khoa học đa nhiệm vụ - luôn phải đáp ứng không chỉ những đòi hỏi của các ngành, các lĩnh vực khác nhau mà còn phục vụ cho các nhu cầu thiết yếu của con người trong đời sống hàng ngày.

Đề tài đồ án “KHÓA SỐ ĐIỆN TỬ” của em là một mạch điện tử sử dụng ma trận phím 4x4 để nhập mật khẩu mở cửa. Mật khẩu gồm 6 số, mạch điện có một màn hình LCD 16x2 để hiện thị mật khẩu và tương tác với người dùng. Người dùng cũng có thể đổi được mật khẩu dễ dàng. Ngoài ra, khi một người nào đó cố tình gõ sai mật khẩu quá 5 lần thì sẽ có còi reo lên, đồng thời mạch điện có một module Sim800A mini, gọi điện cảnh báo cho người dùng.

**MỤC LỤC**

[**1. GIỚI THIỆU 4**](#_Toc45567151)

[**1.1 Tổng quan 4**](#_Toc45567152)

[**1.2 Nhiệm vụ đề tài 4**](#_Toc45567153)

[**2. LÝ THUYẾT 4**](#_Toc45567154)

[**2.1 Tìm hiểu về vi điều khiển 4**](#_Toc45567155)

[**2.1.1 Các thông số cơ bản 7**](#_Toc45567156)

[**2.1.2 Tổ chức bộ nhớ 9**](#_Toc45567157)

[**2.2 Tìm hiểu về module Sim800A mini 10**](#_Toc45567158)

[**2.3 Tìm hiểu về Servo SG90 13**](#_Toc45567159)

[**2.3.1 Mô tả Servo SG90 13**](#_Toc45567160)

[**3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 15**](#_Toc45567161)

[**4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM 19**](#_Toc45567162)

[**5. KẾT QUẢ THỰC HIỆN 40**](#_Toc45567163)

[**6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 42**](#_Toc45567164)

[**6.1 Kết luận 42**](#_Toc45567165)

[**6.2 Hướng phát triển 42**](#_Toc45567166)

[**7. TÀI LIỆU THAM KHẢO 43**](#_Toc45567167)

DANH SÁCH HÌNH MINH HỌA

[**Hình 1.** *Schematic Design..........................................................................................*..17](#_Toc310380287)

**Hình 2**. *Layout Design............................................................................................*.....18

**Hình 3.** *Mô phỏng trên phần mềm Proteus.............................................................*.....40

**Hình 4.** *Kết quả thi công........................................................................................*.....40

**Hình 5.** *Kết quả thực tế........................................................................................*.......41

# GIỚI THIỆU

## Tổng quan

Hiện nay, vấn đề về bảo mật đang là lĩnh vực rất được chú trọng phát triển. Theo sự gia tăng dân số, mật độ dân cư dày đặc, đa dạng vùng miền thì vấn đề an ninh luôn được quan tâm nhất. Vì vậy, mỗi người, hộ gia đình mên tự trang bị cho mình một hệ thống bảo mật, có thể sử dụng khóa số, khóa vân tay, nhận diện khuôn mặt,.. Trong đó, đề tài này em sẽ thiết kế một mạch điện khóa số điện tử.

## Nhiệm vụ đề tài

Thiết kế khóa số điện tử để phục vụ cho mục đích bảo mật, an ninh.

Nội dung 1: Tìm hiểu nguyên lý quét phím ma trận, lý thuyết lập trình vi điều khiển PIC.

Nội dung 2: Tìm hiểu về module sim 800A mini, vi xử lý, cá linh kiện khác cần dùng, …

Nội dung 3: Thiết kế bộ phần cứng mạch, vẽ Schematic và PCB trên Altium.

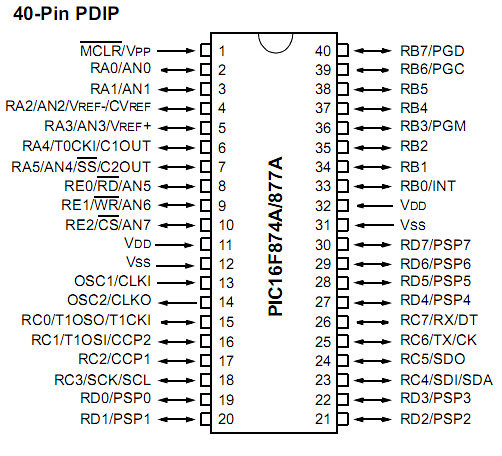
Nội dung 4: Thiết kế phần mèm, sử dụng các giải thuật lập trình, mô phỏng phần mềm trên proteus,…

# LÝ THUYẾT

## Tìm hiểu về vi điều khiển

PIC là viết tắt của “Programable Intellligent Computer”, có thể tạm dịch là “máy tính thông minh khả trình” do hãng General Instructment đặt tên đầu tiên cho vi điều khiển đầu tiên của họ - PIC1650, được thiết kế để dùng làm các thiết bị ngoại vi cho vi điều khiển CP1600. Vi điều khiển này sau đó được nghiên cứu và phát triển thêm và từ đó hình thành nên dòng vi điều khiển PIC ngày nay.

Trong đó, PIC16F877 thuộc về một họ của microcontrollers 8-bit của kiến trúc RISC (vi điều khiển có tập lệnh rút gọn). Để thực hiện đề tài này, nhóm đã sử dụng PIC16F877 bởi nó khá phổ biến, dễ sử dụng và lập trình, rất phù hợp với sinh viên mới bắt đầu làm quen đến lập trình cho vi điều khiển.



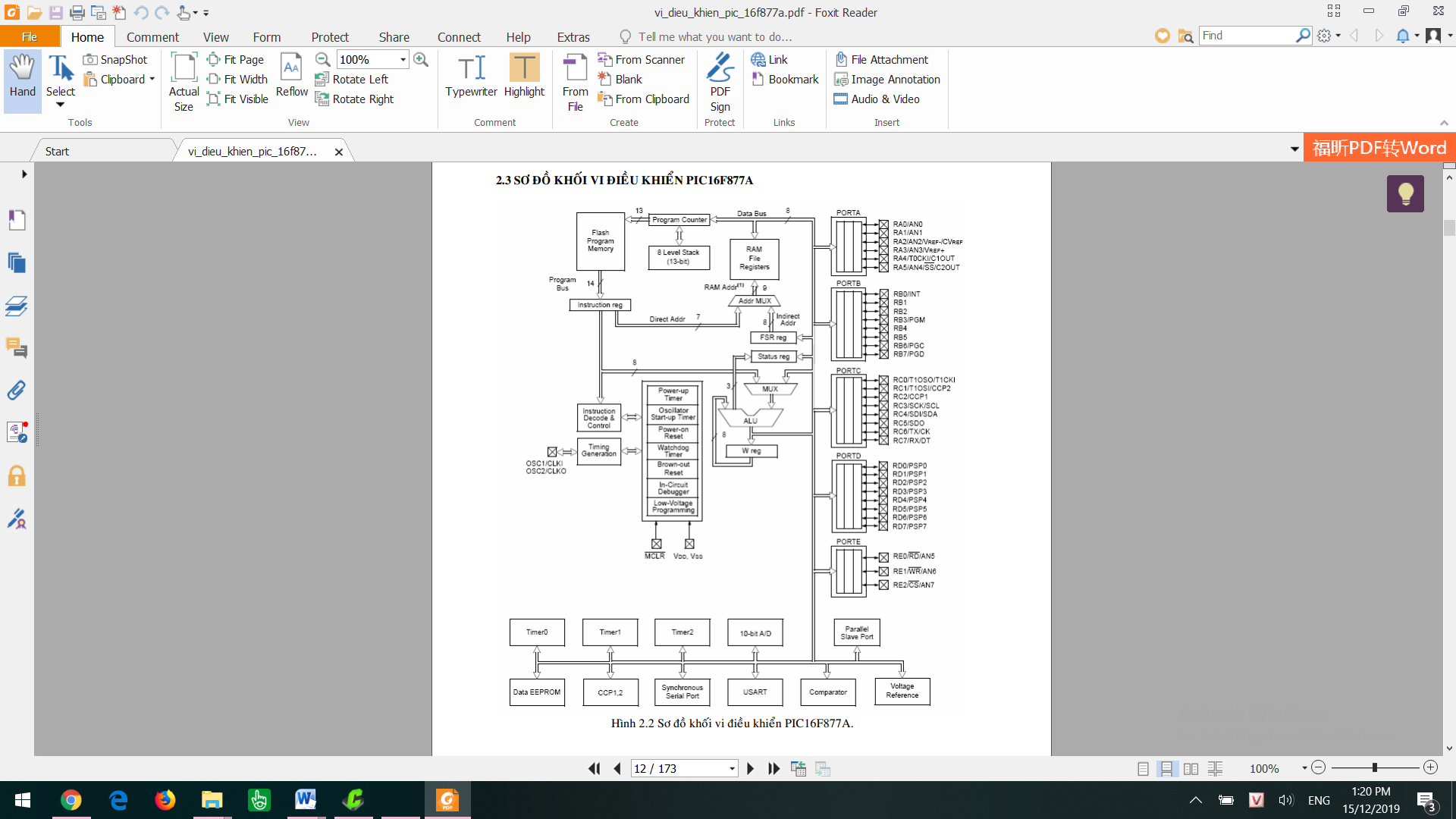
*Sơ đồ chân của vi điều khiển 16F877A*

**Giải thích:**

* OSC1/CLKIN là chân ngả vào của mạch dao động thạch anh. Định tần cho xung nhịp và cũng là một ngả vào của mạch tạo xung nhịp.
* OSC2/CLKOUT là chân ngả ra của mạch dao động thạch anh và là ngả ra của xung nhịp.
* /MCLR (Master Clear) là chân Reset, tác dụng của lệnh Reset là trả chương trình về địa chỉ 0000h. Lệnh tác dụng ở mức áp thấp.
* RA0, RA1, RA2, RA3, là các chân xuất nhập của bến cảng Port A.
* RA4/TOCKI là chân  đa nhiệm, vừa làm chân xuất nhập của Port A và lại là chân lấy xung cho thanh đếm của đồng hồ Timer 0 (ở chân này có cực Drain để hở).
* RB0/INT là chân đa nhiệm, vừa làm chân xuất nhập của Port B và lại là chân phát động theo ngắt ngoài. Các chân của Port B có thể được lập trình để có trở kháng lớn dùng để làm ngả vào, nhập trạng thái ngoài vào PIC.
* RB1, RB2, RB3, là các chân xuất nhập của bến cảng Port B.
* RB4, RB5, là các chân xuất nhập của Port B và phát động ngắt theo sự thay đổi trên các chân này.
* RB6, là 1 chân xuất nhập của Port B và phát động ngắt theo sự thay đổi trên chân này. Nó còn có thể lập trình để dùng chân này phát xung nhịp dùng cho công năng truyền dữ liệu nối tiếp.
* RB7, là 1 chân xuất nhập của Port B và phát động ngắt theo sự thay đổi trên chân này. Nó còn có thể lập trình để dùng chân này trao đổi dữ liệu dùng cho công năng truyền dữ liệu nối tiếp.
* VSS là chân nối masse để lấy dòng (ở đây là dòng điện tử).
* VDD là chân nối vào đường nguồn dương (từ 2V đến 5.5V).

### **2.1.1 Các thông số cơ bản**

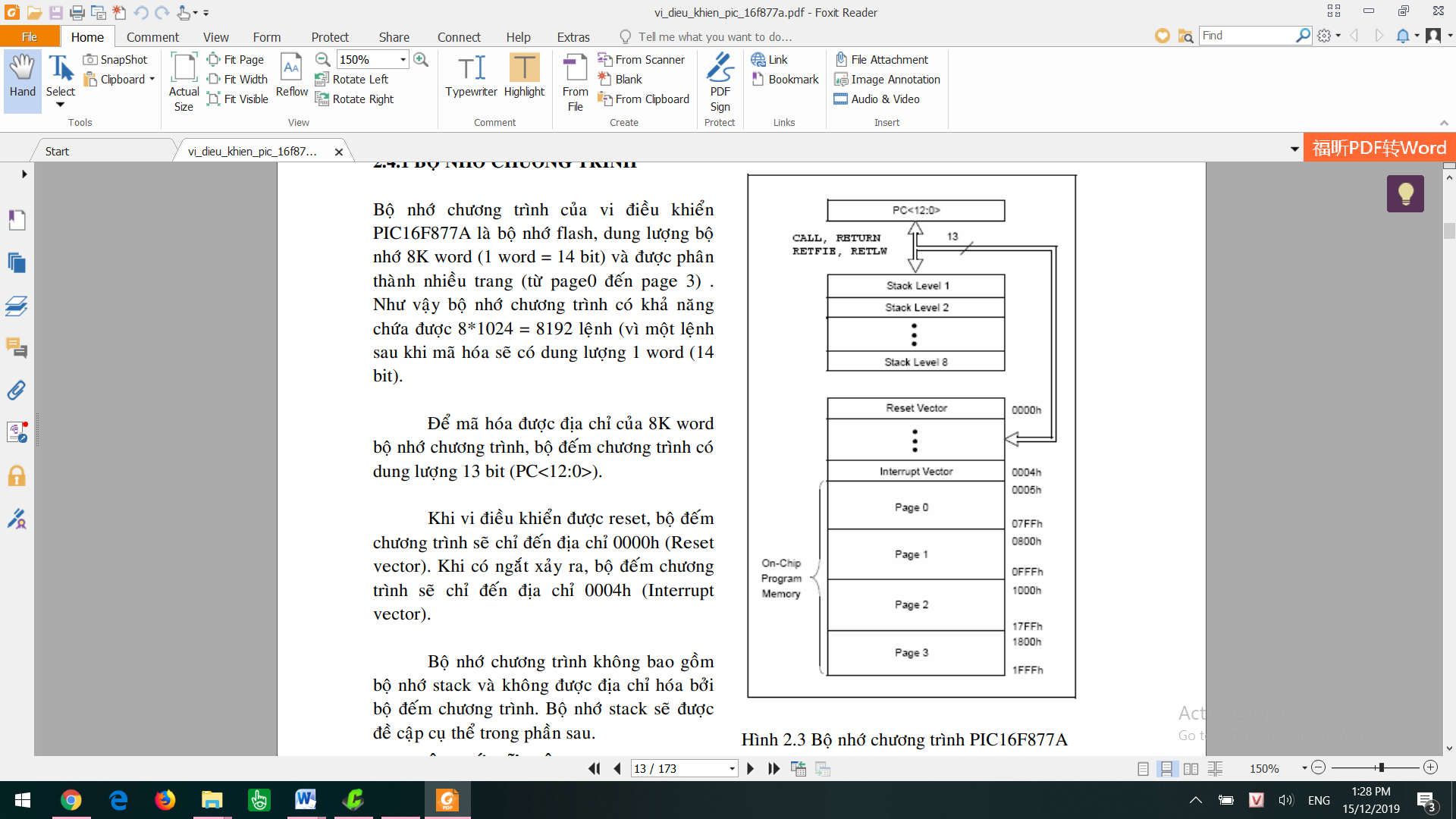
Taäp leänh goàm 35 leänh coù ñoä daøi 14 bit. Moãi leänh ñeàu ñöôïc thöïc thi trong moät chu kì xung clock. Toác ñoä hoaït ñoäng toái ña cho pheùp laø 20 MHz vôùi moät chu kì leänh laø 200ns. Boä nhôù chöông trình 8Kx14 bit, boä nhôù döõ lieäu 368x8 byte RAM vaø boä nhôù döõ lieäu EEPROM vôùi dung löôïng 256x8 byte. Soá PORT I/O laø 5 vôùi 33  
pin I/O.  
  *Caùc ñaëc tính ngoaïi vi bao goàm caùc khoái chöùc naêng sau:*+ Timer0: boä ñeám 8 bit vôùi boä chia taàn soá 8 bit.  
+ Timer1: boä ñeám 16 bit vôùi boä chia taàn soá, coù theå thöïc hieän chöùc naêng ñeám döïa vaøo xung clock ngoaïi vi ngay khi vi ñieàu khieån hoaït ñoäng ôû cheá ñoä sleep.  
+ Timer2: boä ñeám 8 bit vôùi boä chia taàn soá, boä postcaler.  
+ Hai boä Capture/so saùnh/ñieàu cheá ñoä roâng xung.  
+ Caùc chuaån giao tieáp noái tieáp SSP (Synchronous Serial Port), SPI vaø I2C.  
+ Chuaån giao tieáp noái tieáp USART vôùi 9 bit ñòa chæ.  
+ Coång giao tieáp song song PSP (Parallel Slave Port) vôùi caùc chaân ñieàu khieån RD, WR, CS ôû beân ngoaøi.  
 *Caùc ñaëc tính Analog:*  
+ 8 keânh chuyeån ñoåi ADC 10 bit.  
+ Hai boä so saùnh.  
+ Beân caïnh ñoù laø moät vaøi ñaëc tính khaùc cuûa vi ñieàu khieån nhö:  
+ Boä nhôù flash vôùi khaû naêng ghi xoùa ñöôïc 100.000 laàn.  
+ Boä nhôù EEPROM vôùi khaû naêng ghi xoùa ñöôïc 1.000.000 laàn.  
+ Döõ lieäu boä nhôù EEPROM coù theå löu tröõ treân 40 naêm.  
+ Khaû naêng töï naïp chöông trình vôùi söï ñieàu khieån cuûa phaàn meàm.  
+ Naïp ñöôïc chöông trình ngay treân maïch ñieän ICSP (In Circuit Serial Programming) thoâng qua 2 chaân.  
+ Watchdog Timer vôùi boä dao ñoäng trong.  
+ Chöùc naêng baûo maät maõ chöông trình.  
+ Cheá ñoä Sleep.  
+ Coù theå hoaït ñoäng vôùi nhieàu daïng Oscillator khaùc nhau.

  
*Sơ đồ khối vi điều khiển 16F877A*

### **2.1.2 Tổ chức bộ nhớ**

Caáu truùc boä nhôù cuûa vi ñieàu khieån PIC16F877A bao goàm boä nhôù chöông trình (Program memory) vaø boä nhôù döõ lieäu (Data Memory):

Boä nhôù chöông trình cuûa vi ñieàu khieån PIC16F877A laø boä nhôù flash, dung löôïng boä nhôù 8K word (1 word = 14 bit) vaø ñöôïc phaân thaønh nhieàu trang (töø page 0 ñeán page 3). Nhö vaäy boä nhôù chöông trình coù khaû naêng chöùa ñöôïc 8\*1024 = 8192 leänh (vì moät leänh sau khi maõ hoùa seõ coù dung löôïng 1 word (14bit)).



*Boä nhôù chöông trình cuûa PIC16F877A*

## 2.2 Tìm hiểu về module Sim800A mini

**2.2.1 Tìm hiểu chức năng và cấu tạo**

- Sim800A là một module có hai tần số GSM/GPRS. Hiệu suất ổn định, bề ngoài nhỏ gọn, hiệu quả chi phí cao, đáp ứng được nhu cầu của khách hàng.

- Module thích hợp để phát triển các dự án truyền tin qua điện thoại, định vị,...

- Tần số hoạt động của Sim800A: 900/1800MHz GSM/GPRS.

- Nguồn vào: 4.5-5VDC.

- Dòng cấp: 500mA trở lên (nên dùng nguồn cấp >2A).

- Đèn báo: led.

- Có tụ bù, Diode giảm áp.

- Chức năng các chân:

+ RXD: Chân nhận Uart RX.

+ TXD: Chân truyền Uart TX.

+ BAT: Nếu sử dụng pin Lion 3.7VDC thì dùng chân này để cấp nguồn.

+ 5V:Nguồn V+ cấp 4.5~5VDC Module Sim800A

+ GND: Chân Mass, cấp 0V.

**2.2.2 Tìm hiểu về tập lệnh**

**a. Lệnh chung**

Lệnh:      **AT**<CR><LF>

Trả lời:   OK<CR><LF>

Mô tả : Kiểm tra đáp ứng , nếu trả về OK thì Module hoạt động.

Lệnh:      **ATE[x]**<CR><LF>

Trả lời:   OK<CR><LF>

Mô tả : Chế độ echo là chế độ phản hồi dữ liệu truyền đến.

x = 1 bật chế độ echo , x = 0 tắt chế độ echo (ta nên tắt chế độ này) .

Lệnh:      **AT+IPR=[baud rate]**<CR><LF>

Trả lời:   OK<CR><LF>

Mô tả : cài đặt tốc độ giao tiếp dữ liệu với module Sim800A, chỉ cài được các tốc độ sau baud rate : 0 (auto), 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

**b. Lệnh điều khiển cuộc gọi**

Lệnh:      **ATD[Số\_điện\_thoại];**<CR><LF>

Trả lời:   OK<CR><LF>

Mô tả : Lệnh thực hiện cuộc gọi.

Lệnh:      **ATH**<CR><LF>

Trả lời:   OK<CR><LF>

Mô tả : Lệnh thực hiện kết thúc cuộc gọi , hoặc cúp máy khi có cuộc gọi đến.

Lệnh:      **ATA**<CR><LF>.

Trả lời:   OK<CR><LF>.

Mô tả : Lệnh thực hiện chấp nhận khi có cuộc gọi đến.

Chú ý: khi nhân được cuộc gọi đến thì Module Sim 900A sẽ phản hồi về như sau.

     RING.

     RING.

Nếu muốn hiển thị thông tin người gọi đến các bạn thực hiện thêm lệnh sau.

Lệnh:      **AT+CLIP=1**<CR><LF>.

Trả lời:   OK<CR><LF>.

**c. Lệnh điều khiển tin nhắn**

Lệnh:     **AT+CMGF=1**<CR><LF>.

Trả lời:   OK<CR><LF>.

Mô tả : Lệnh đưa SMS về chế độ Text , phải có lệnh này mới gửi nhận tin nhắn dạng Text.

Cấu trúc gửi tin nhắn.

Lệnh:      **AT+CMGS=”Số\_điện \_thoại”**<CR><LF>

Đợi đến khi có ký tự ‘>’ được gửi về.

Gửi nội dụng tin nhắn : “This is a test”

Gửi mã Ctrl+Z  hay  0x1A để kết thúc nội dung và gửi tin nhắn.

Trả lời :  OK<CR><LF>.

Lệnh:      **AT+CMGR=1**<CR><LF>.

Mô tả : Đọc một nhắn vừa gửi đến, lệnh được trả về nội dung tin nhắn, thông tin người gửi, thời gian gửi.

Chú ý: Nếu Module Sim 900A nhận được một tin nhắn bất kỳ, thì nó gửi về cụm “+CMTI” để thông báo.

Trong trường hợp ta muốn hiển thị trực tiếp nội dung tin nhắn(không lưu vào bộ nhớ của sim) bạn gửi lệnh sau.

**AT+CNMI=2,2**<CR><LF>.

Chú ý: sau mỗi lệnh các ta thường thấy <CR><LF>  thực chất nó là hai mã điều khiển.

<CR>  tương ứng 0x0D (hexa).

<LF>  tương ứng 0x0A (hexa)  .

Khi test trên máy tính sau mỗi lệnh ta chỉ cần ấn phím Enter.

Ví dụ  : gõ AT rồi ấn Enter.

**2.3 Tìm hiểu về Servo SG90**

**2.3.1 Mô tả Servo SG90**

- Động cơ servo SG90 có thể xoay 180o

- Chủng loại : Động cơ servo SG90

- Kích thước : 22.2  x  11.8  x  32  mm

- Điện áp hoạt động : 5V

- Ứng dụng: Công dụng chính của động cơ servo là đạt được góc quay chính xác trong khoảng từ 90o – 180o. Việc điều khiển này có thể ứng dụng để lái robot, di chuyển các tay máy lên xuống, quay một cảm biến để quét khắp phòng…

**2.3.2 Thông số kỹ thuật**

- Khối lượng : 9g

- Kích thước: 22.2x11.8.32 mm

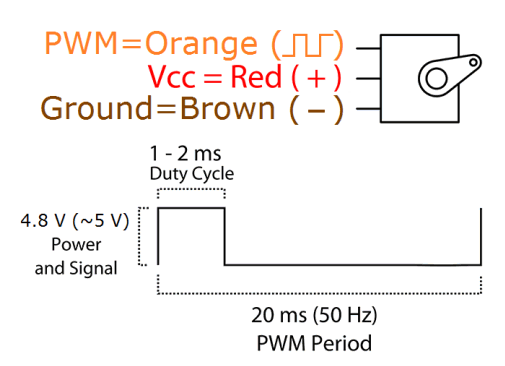
- Momen xoắn: 1.8kg/cm

- Tốc độ hoạt động: 60 độ trong 0.1 giây

- Điện áp hoạt động: 4.8V(~5V)

- Nhiệt độ hoạt động: 0 ºC – 55 ºC

- Kết nối dây màu đỏ với 5V, dây màu nâu với mass, dây màu cam với chân phát xung của vi điều khiển. Ở chân xung cấp một xung từ 1ms-2ms theo để điều khiển góc quay theo ý muốn.



# THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

* **Yêu cầu thiết kế**
  + Khi cần vào nhà, ta phải nhập đúng mật khẩu để mở cửa vào nhà, mật khẩu luôn mặc định gồm 6 số.
  + Người dùng nhập mật khẩu bằng ma trận bàn phím 4x4.
  + Nếu nhập sai mật khẩu quá 5 lần, sẽ có coi hú lên báo động, đồng thời sẽ báo cho người dùng biết có người nhập sai mật khẩu quá 5 lần. Còi sẽ ngưng kêu chỉ khi nhập đúng mật khẩu lại.
  + Người dùng có thể đổi mật khẩu hoặc đổi số điện thoại dễ dàng, nhập đúng mật khẩu hiện tại để đổi mật khẩu hoặc số điện thoại.
  + Hệ thống luôn lưu mật khẩu và số điện thoại vẫn như nguyên như lần gần nhất thay đổi.
  + Số dược nhập từ bàn phím hay cửa đã đóng đều hiện lên màn hình LCD 16x2.
  + Liệt kê các phần cứng sử dụng:
* **Phân tích thiết kế**
  + Mật khẩu lúc đầu của hệ thống sẽ mặc định là “123456”
  + Để báo cho người dùng, ta sử dụng module Sim800A mini giao tiếp Uart.
  + Để hệ thống khi mất điện hoặc reset luôn giữ mật khẩu và số điện thoại như lần thay đổi gần nhất, ta sử dụng EEPROM, lưu dữ liệu vào trong EEPROM, có thể sử dụng EEPROM ngoài hoặc EEPROM của vi điều khiển, ở đây ta sử dụng EEPROM của vi điều khiển vì dữ liệu cần lưu không nhiều.
* Vẽ sơ đồ khối chi tiết

Điện thoại

SERVO SG90

Sim800A



Keypad 4x4

LCD

16x2

Buzzer

**PIC**

**16F877A**

**[[1]](#footnote-1)**

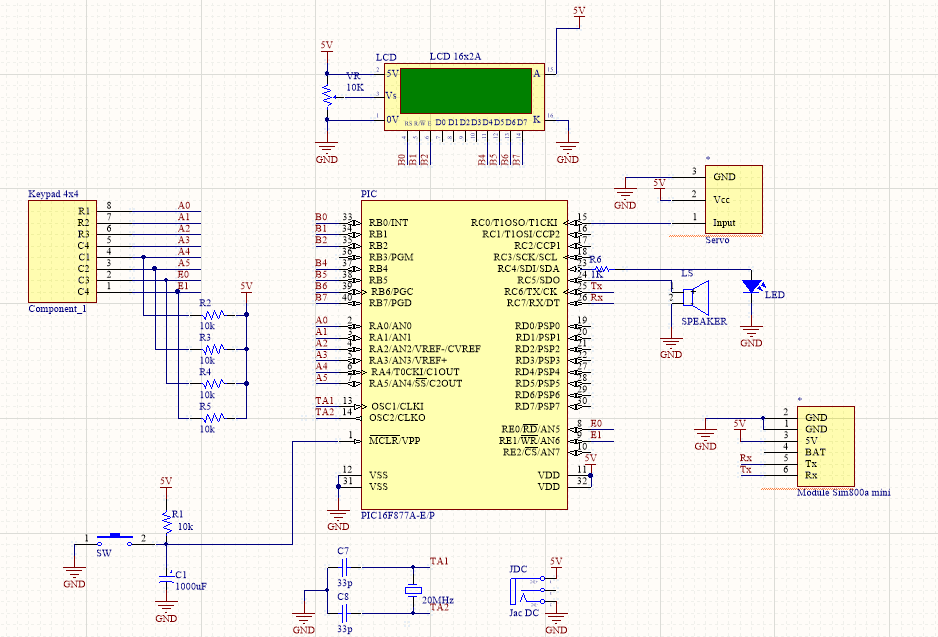
LED

**POWER SUPPLY:**

Adapter 5V – 2A

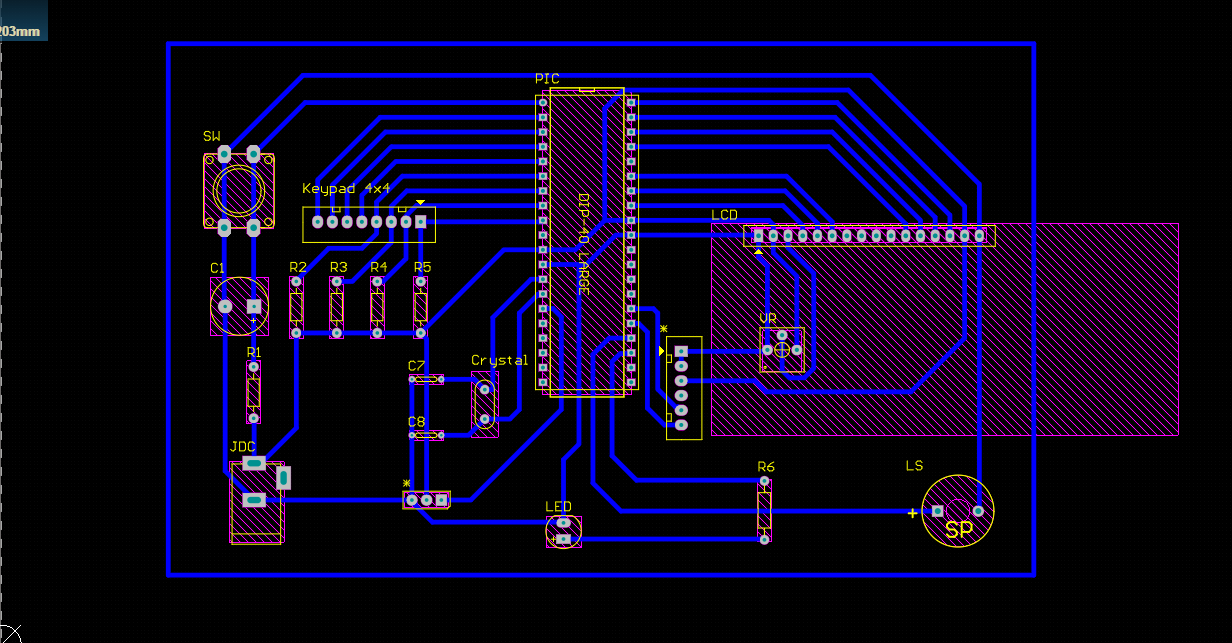
* + Adpater là nguồn nuôi mạch điện.
  + PIC 16F877A là nơi xử lý trung tâm.
  + Keypad 4x4 là khối nhận tính hiệu nhập từ người dùng.
  + LCD 16x2 dùng để hiện thị số được nhập.
  + Buzzer reo khi nhâập sai mật khẩu quá 5 lần.
  + Sim800A nhận tín hiệu từ PIC gọi điện cho người dùng.
  + Điện thoại của người dùng nhận thông báo từ Sim800A.

* Tính toán và vẽ sơ đồ mạch chi tiết



**Hình 1.** *Schematic Design*

**Hình 1** trên đây là bản thiết kế Schematic hoàn chỉnh bao gồm tất cả các yêu cầu hệ thống nêu ở trên.



**Hình 2**. *Layout Design*

+ Chân (1) là chân Reset, qua 1 nút nhấn có tác dụng là trả chương trình về địa chỉ 0000h. Lệnh tác dụng ở mức áp thấp.

+ Chân (2) đến chân (9) nối với ma trận phím 4x4 để nhận tín hiệ được nhập

+ Chân (11)(32) nối vào đường nguồn dương (5V).

+ Chân (12)(31) nối đất để lấy dòng.

+ Chân (13) là chân ngõ vào của mạch dao động thạch anh; chọn thạch anh 20000MHz; chân (14) là chân ngõ ra của mạch dao động thạch anh và là ngả ra của xung nhịp.

+ Chân (15) nối với chân (OUT) của Servo SG90 để thực hiện mô phỏng chức năng đóng/mở cửa.

+ Chân (23) nối tới Buzzer, buzzer reo lên khi có người nhập sai mật khẩu quá 5 lần.

+ Chân (24) nối với LED – LED này sẽ sáng khi ta cấp nguồn cho mạch.

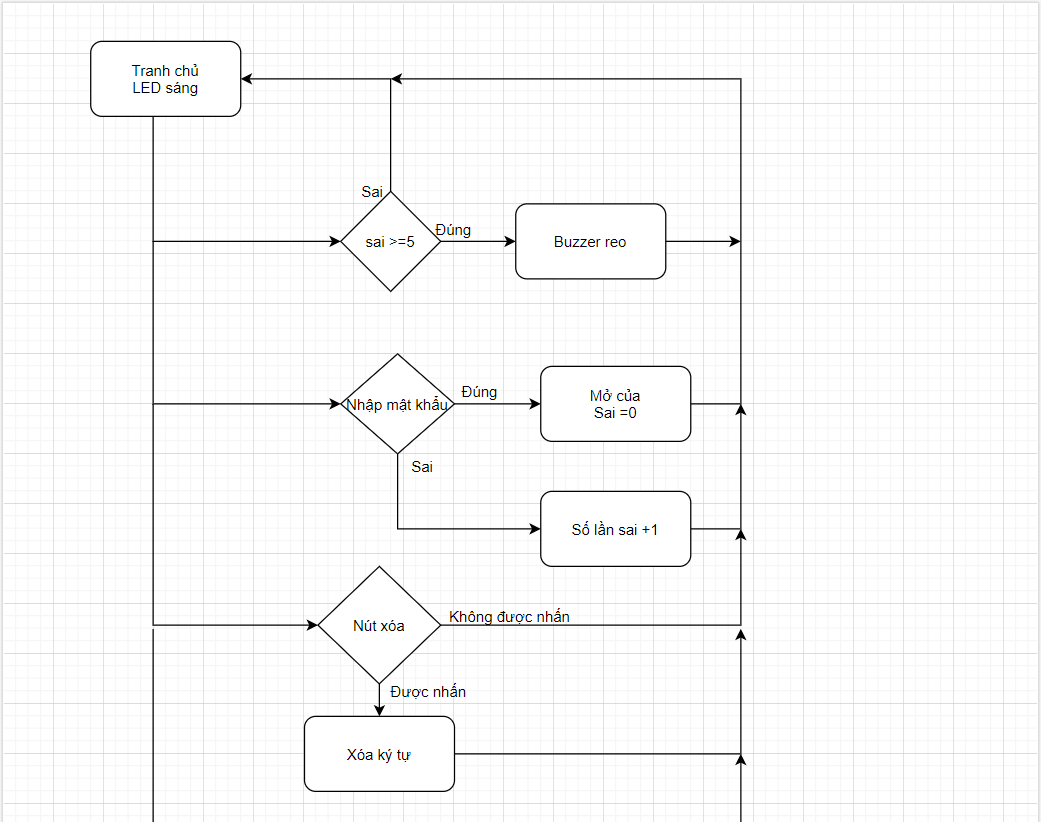
+ Chân (25)(26) gửi tập lệnh cho Module Sim800A thực hiện nhắn, gọi.

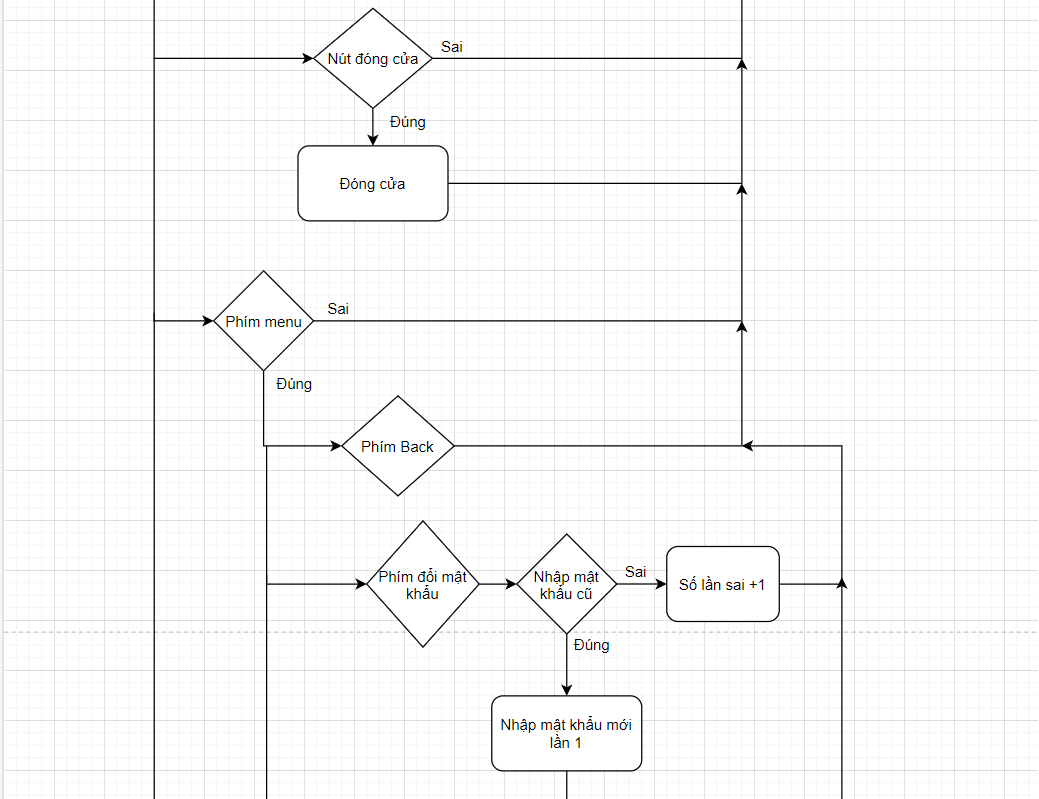
+ Chân (33)(34)(35) nối với chân RS, RW, E của LCD 16x2

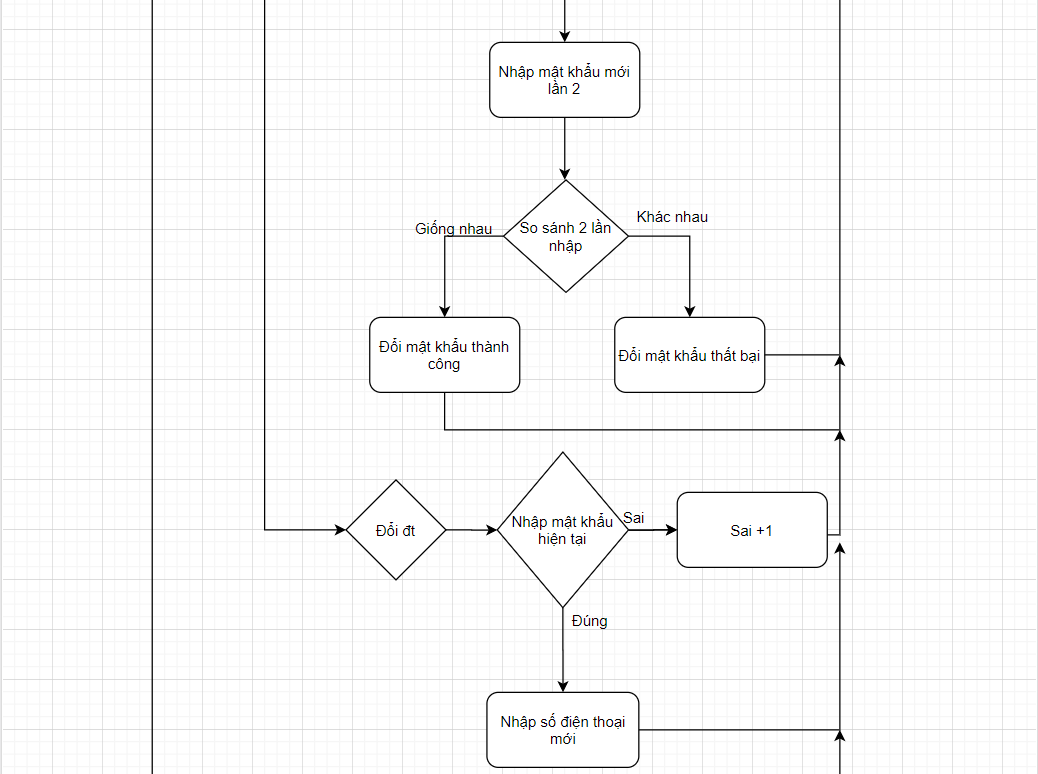
+ Chân (37) đến chân (39) nối với chân D4 đến chân D7 của LCD 16x2, LCD giao tiếp 4 bit

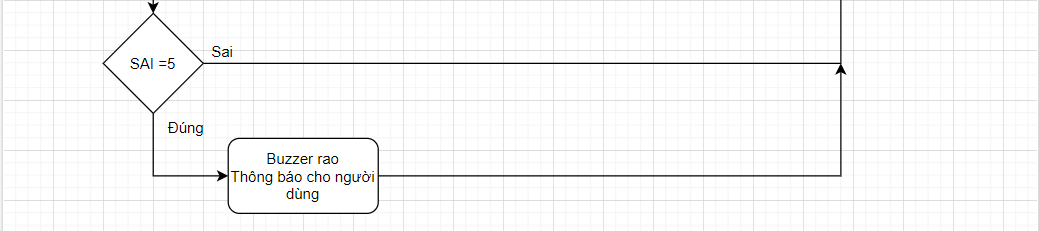
# THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

* Yêu cầu đặt ra cho phần mềm
  + Khi cấp nguồn cho mạch, sẽ có đèn LED sáng lên cho biết hệ thống đã hoạt động.
  + Sử dụng giải thuật quét dòng để biết phím nào được nhấn, đồng thời chống rung phím bằng phần mềm.
  + Có thể đổi được mật khẩu và số điện thoại dễ dàng qua mật khẩu cũ.
  + Khi nhập sai mật khẩu quá 5 lần thì buzzer sẽ reo lên và thông báo cho người dùng biết.
* Vẽ lưu đồ giải thuật chi tiết:









* Chương trình viết trên CCS (code):

#include <khoasov1.h>

#FUSES NOWDT ***//No Watch Dog Timer***

#FUSES NOBROWNOUT ***//No brownout reset***

#FUSES NOLVP //No low voltage prgming, B3(PIC16) or B5(PIC18) used FOR I/O

#use delay(clock=20000000)

#use rs232(baud=9600,xmit=PIN\_C6,rcv=PIN\_C7,stream=GSM)

#define LCD\_RS\_PIN PIN\_B0 ***//Cấu hình chân LCD***

#define LCD\_RW\_PIN PIN\_B1

#define LCD\_ENABLE\_PIN PIN\_B2

#define LCD\_DATA4 PIN\_B4

#define LCD\_DATA5 PIN\_B5

#define LCD\_DATA6 PIN\_B6

#define LCD\_DATA7 PIN\_B7

#include <lcd.c>

const char key[16] ={'1', '2', '3', ' ','4', '5', '6', ' ','7', '8', '9',' ','\*', '0', '#', ' '};

int a=0,sai=0,count=0,dmk=0,solan=0,dsdt=0,doi=0,k=0;

void tranhchu() ***//Chương trình con quay về màn hình nhập mật khẩu***

{

if(k==2)

{

lcd\_putc('\f');

lcd\_putc("Nhap mat khau:");

a=0,count=0,dmk=0,dsdt=0,solan=0,doi=0,k=0;

}

else

{

delay\_ms(2000);

lcd\_putc('\f');

lcd\_putc("Nhap mat khau:");

a=0,count=0,dmk=0,dsdt=0,solan=0,doi=0,k=0;

}

}

void clear() ***//Chương trình con khi tao nhấn phím Back trên menu***

{

a=0,count=0,dmk=0,dsdt=0,solan=0,doi=0,k=0;

}

void mocua() ***//Chương trình con mở cửa***

{

for(int i=0;i<50;i++)

{

output\_high(pin\_c0);

delay\_us(2000);

output\_low(pin\_c0);

delay\_us(18500);

}

}

void dongcua() ***//Chương trình con đóng cửa***

{

for(int i=0;i<50;i++)

{

output\_high(pin\_c0);

delay\_us(1000);

output\_low(pin\_c0);

delay\_us(19000);

}

}

void quetdong(unsigned char row) ***//Quét dòng***

{

output\_high(pin\_a0);

output\_high(pin\_a1);

output\_high(pin\_a2);

output\_high(pin\_a3);

output\_high(pin\_a4);

output\_high(pin\_a5);

output\_high(pin\_e0);

output\_high(pin\_e1);

if(row==0)output\_low(pin\_a0);

if(row==1)output\_low(pin\_a1);

if(row==2)output\_low(pin\_a2);

if(row==3)output\_low(pin\_a3);

}

unsigned char quetphim() ***//Quét phím để nhận biết phím nào được nhấn***

{

unsigned char r=0,c=0xFF;

for (r=0;r<4;r++)

{

quetdong(r);

while (input(pin\_a4)==0) c=r\*4+0;

while (input(pin\_a5)==0) c=r\*4+1;

while (input(pin\_e0)==0) c=r\*4+2;

while (input(pin\_e1)==0) c=r\*4+3;

}

return c;

}

void khoitao() ***//Cấu hình module Sim800A mini***

{

printf("AT");

putc(13);

delay\_ms(500);

printf("ATE0");

putc(13);

delay\_ms(500);

printf("AT&3");

putc(13);

delay\_ms(500);

printf("AT+CSCS=\"GSM\"");

putc(13);

delay\_ms(500);

printf("AT+CMGF=1");

putc(13);

delay\_ms(500);

printf("AT+CNMI=2,1,0,0,0");

putc(13);

delay\_ms(500);

}

void canhbao() ***//Cảnh bảo nhập sai mật khẩu quá 5 lần***

{

lcd\_putc('\f');

lcd\_putc("Ban nhap sai mat");

lcd\_gotoxy(2,2);

lcd\_putc("khau qua 5 lan!");

}

void baodong() ***//Thông báo cho người dùng***

{

printf("ATD%c%c%c%c%c%c%c%c%c%c;",read\_eeprom(0x10),read\_eeprom(0x11),read\_eeprom(0x12),read\_eeprom(0x13),read\_eeprom(0x14),read\_eeprom(0x15),read\_eeprom(0x16),read\_eeprom(0x17),read\_eeprom(0x18),read\_eeprom(0x19));

putc(13);

delay\_ms(15000);

printf("ATH");

putc(13);

}

void main() ***//Chương trình chính***

{

char mk[10]={},new1[6]={},new2[6]={};

unsigned char phim;

//eeprom();

output\_low(pin\_c0);

lcd\_init(); ***//Khởi tạo LCD***

lcd\_putc("Nhap mat khau:");

while(TRUE)

{

if(sai>=5) ***//Nếu đã có người nhập sai 5 lần buzzer sẽ reo liên tục***

{

output\_high(pin\_c5);

}

output\_high(pin\_c4); ***//Đèn led thông báo mạch điện đang hoạt động***

phim=quetphim();

if(phim!=0xFF && phim!=3 && phim!=7 && phim!=11 && phim!=15) ***//Nếu không phải phím A, B, C, D được nhấn***

{

delay\_ms(5); ***//Delay để chống rung phím***

if(phim!=0xFF && phim!=3 && phim!=7 && phim!=11 && phim!=15)

{

if (a==0 && k!=1) ***//Số đầu tiên được nhấn***

{

mk[0]=key[phim];

lcd\_gotoxy(6,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(6,2);

lcd\_putc("\*"); ***//Sau cỡ 300ms số sẽ được ẩn thành dấu \****

k=0;

}

}

if (a==1)

{

mk[1]=key[phim];

lcd\_gotoxy(7,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(7,2);

lcd\_putc("\*");

}

}

if (a==2)

{

mk[2]=key[phim];

lcd\_gotoxy(8,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(8,2);

lcd\_putc("\*");

}

}

if (a==3)

{

mk[3]=key[phim];

lcd\_gotoxy(9,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(9,2);

lcd\_putc("\*");

}

}

if (a==4)

{

mk[4]=key[phim];

lcd\_gotoxy(10,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(10,2);

lcd\_putc("\*");

}

}

if (a==5)

{

mk[5]=key[phim];

lcd\_gotoxy(11,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(11,2);

lcd\_putc("\*");

count=1;

}

}

if(a==6 && dsdt==2)

{

mk[6]=key[phim];

lcd\_gotoxy(12,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(12,2);

lcd\_putc("\*");

}

}

if(a==7 && dsdt==2)

{

mk[7]=key[phim];

lcd\_gotoxy(13,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(13,2);

lcd\_putc("\*");

}

}

if(a==8 && dsdt==2)

{

mk[8]=key[phim];

lcd\_gotoxy(14,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(14,2);

lcd\_putc("\*");

}

}

if(a==9 && dsdt==2)

{

mk[9]=key[phim];

lcd\_gotoxy(15,2);

printf(lcd\_putc,"%c",key[phim]);

if(dsdt!=2)

{

delay\_ms(300);

lcd\_gotoxy(15,2);

lcd\_putc("\*");

}

count=2;

}

a++;

}

}

if (count==1 && phim==15 && dmk==0 && dsdt==0) ***//Kiểm tra mật khẩu được nhập***

{

delay\_ms(5);

if(count==1 && phim==15 && dmk==0 && dsdt==0)

{

if (mk[0]==read\_eeprom(0x00) && mk[1]==read\_eeprom(0x01) && mk[2]==read\_eeprom(0x02) && mk[3]==read\_eeprom(0x03) && mk[4]==read\_eeprom(0x04) && mk[5]==read\_eeprom(0x05)) /Nếu đúng thì mở cửa

{

lcd\_putc('\f');

lcd\_putc("Cua da mo");

lcd\_gotoxy(9,2);

lcd\_putc("Moi vao!");

mocua();

count=0;sai=0;a=0;output\_low(pin\_c5);

}

Else // Sai thì số lần sai tăng lên 1

{

lcd\_putc('\f');

lcd\_putc("Mat khau sai");

lcd\_gotoxy(1,2);

lcd\_putc("Moi ban nhap lai");

sai++;

tranhchu();

}

}

}

if(phim==3) ***//Nút xóa ký tự nếu như ấn nhầm***

{

DELAY\_MS(5);

if(phim==3)

{

if(a==1)

{

lcd\_gotoxy(6,2);

lcd\_putc(" ");

a--;

}

if(a==2)

{

lcd\_gotoxy(7,2);

lcd\_putc(" ");

a--;

}

if(a==3)

{

lcd\_gotoxy(8,2);

lcd\_putc(" ");

a--;

}

if(a==4)

{

lcd\_gotoxy(9,2);

lcd\_putc(" ");

a--;

}

if(a==5)

{

lcd\_gotoxy(10,2);

lcd\_putc(" ");

a--;

}

if(a>=6 && dsdt!=2)

{

lcd\_gotoxy(11,2);

lcd\_putc(" ");

a=5;

}

if(a==6 && dsdt==2)

{

lcd\_gotoxy(11,2);

lcd\_putc(" ");

a--;

}

if(a==7 && dsdt==2)

{

lcd\_gotoxy(12,2);

lcd\_putc(" ");

a--;

}

if(a==8 && dsdt==2)

{

lcd\_gotoxy(13,2);

lcd\_putc(" ");

a--;

}

if(a==9 && dsdt==2)

{

lcd\_gotoxy(14,2);

lcd\_putc(" ");

a--;

}

if(a>=10 && dsdt==2)

{

lcd\_gotoxy(15,2);

lcd\_putc(" ");

a=9;

}

}

}

if (sai==5) ***//Nếu số lần sai là 5 thì sẽ báo động***

{

output\_high(pin\_c5);

sai++;

a=0;

count=0;

canhbao();

khoitao();

baodong();

tranhchu();

}

if(phim==11) ***//Cửa sẽ đóng nếu phím này được nhấn***

{

delay\_ms(5);

if(phim==11)

{

lcd\_putc('\f');

lcd\_putc("Cua da dong");

dongcua();

tranhchu();

}

}

if(phim==7) ***//Phím menu lựa chọn thay đổi mật khẩu hoặc sđt***

{

delay\_ms(5);

if(phim==7)

{

clear();

doi=1;

k=1;

lcd\_putc('\f');

lcd\_gotoxy(1,1);

lcd\_putc("1.Doi mat khau");

lcd\_gotoxy(1,2);

lcd\_putc("2.Doi sdt");

lcd\_gotoxy(11,2);

lcd\_putc("3.Back");

}

}

if(phim==0 && doi==1) //Chọn đổi mật khẩu

{

delay\_ms(5);

if(phim==0 && doi==1)

{

dmk=1;

doi=0;

a=0;

k=0;

lcd\_putc('\f');

lcd\_putc("Mat khau cu:");

}

}

if(phim==1 && doi==1) ***//Chọn thay đổi số điện thoại người dùng***

{

delay\_ms(5);

if(phim==1 && doi==1)

{

dsdt=1;

doi=0;

a=0;

k=0;

lcd\_putc('\f');

lcd\_putc("Mat khau cu:");

}

}

if(phim==2 && doi==1) ***//Phím BACK quay về màn hình chính***

{

delay\_ms(5);

if(phim==2 && doi==1)

{

k=2;

tranhchu();

}

}

if(count==1 && phim==15 && dsdt==1)***//Kiểm tra mật khẩu để đổi sđt***

{

delay\_ms(5);

if(count==1 && phim==15 && dsdt==1)

{

if (mk[0]==read\_eeprom(0x00) && mk[1]==read\_eeprom(0x01) && mk[2]==read\_eeprom(0x02) && mk[3]==read\_eeprom(0x03) && mk[4]==read\_eeprom(0x04) && mk[5]==read\_eeprom(0x05))

{

lcd\_putc('\f'); ***//Đúng thì nhập sđt mới***

lcd\_putc("Nhap sdt moi:");

dsdt=2;a=0;sai=0;output\_low(pin\_c5);

}

Else ***// Sai thì tăng số lần sai và quay về***

{

sai++;

tranhchu();

}

}

}

if(count==2 && phim==15) ***//Nhập đủ 10 số ấn phím enter thì sđt đã được thay đổi***

{

delay\_ms(5);

if(count==2 && phim==15)

{

lcd\_putc('\f');

lcd\_putc("Doi sdt ");

lcd\_gotoxy(7,2);

lcd\_putc("thanh cong");

for(int m=0;m<=9;m++)

{

write\_eeprom(0x10+m,mk[m]); ***//Lưu vào EEPROM***

}

tranhchu();

}

}

if(count==1 && dmk==1 && phim==15 && solan==0 && dsdt==0)

{

delay\_ms(5); ***//Kiểm tra mật khẩu cũ***

if(count==1 && dmk==1 && phim==15 && solan==0 && dsdt==0)

{

if (mk[0]==read\_eeprom(0x00) && mk[1]==read\_eeprom(0x01) && mk[2]==read\_eeprom(0x02) && mk[3]==read\_eeprom(0x03) && mk[4]==read\_eeprom(0x04) && mk[5]==read\_eeprom(0x05))

{

lcd\_putc('\f'); ***//Nhập đúng thì nhập mật khẩu mới lần 1***

lcd\_putc("Mat khau moi");

lcd\_gotoxy(1,2);

lcd\_putc("Lan1:");

solan=1;

count=0;

a=0;sai=0;output\_low(pin\_c5);

}

Else ***//Nhập sai mật khẩu thì tăng số lần sai và quay về***

{

lcd\_putc('\f');

lcd\_putc("Sai mat khau");

sai++;

tranhchu();

}

}

}

if(count==1 && solan==1 && phim==15)

{

delay\_ms(5);

if(count==1 && solan==1 && phim==15)

{

lcd\_putc('\f'); ***//Nhập mật khẩu mới lần 2***

lcd\_gotoxy(1,1);

lcd\_putc("Mat khau moi");

lcd\_gotoxy(1,2);

lcd\_putc("Lan2:");

solan++;

count=0;

a=0;

}

}

if(solan==1 && count==1)

{

for(int n=0;n<=5;n++)

{

new1[n]=mk[n]; ***//Lưu tạm mật khẩu mới lần 1***

}

}

if(solan==2 && count==1)

{

for(int n=0;n<=5;n++)

{

new2[n]=mk[n]; ***//Lưu tạm mật khẩu mới lần 2***

}

}

if(solan==2 && count==1 && phim==15)

{

delay\_ms(5);

if(solan==2 && count==1 && phim==15) //So sánh mật khẩu mới nhập lần 1 và lần 2

{

if(new1[0]==new2[0] && new1[1]==new2[1] && new1[2]==new2[2] && new1[3]==new2[3] && new1[4]==new2[4] && new1[5]==new2[5])

{

lcd\_putc('\f'); ***//Đúng thì đổi mật khẩu thành công***

lcd\_putc("Doi mat khau");

lcd\_gotoxy(7,2);

lcd\_putc("thanh cong");

for(int n=0;n<=5;n++)

{

write\_eeprom(0x00+n,mk[n]);

}

tranhchu();

}

else

{

lcd\_putc('\f'); ***//Nếu 2 lần không khớp thì đổi mật khẩu thất bại***

lcd\_putc("Doi mat khau");

lcd\_gotoxy(9,2);

lcd\_putc("that bai");

tranhchu();

}

}

}

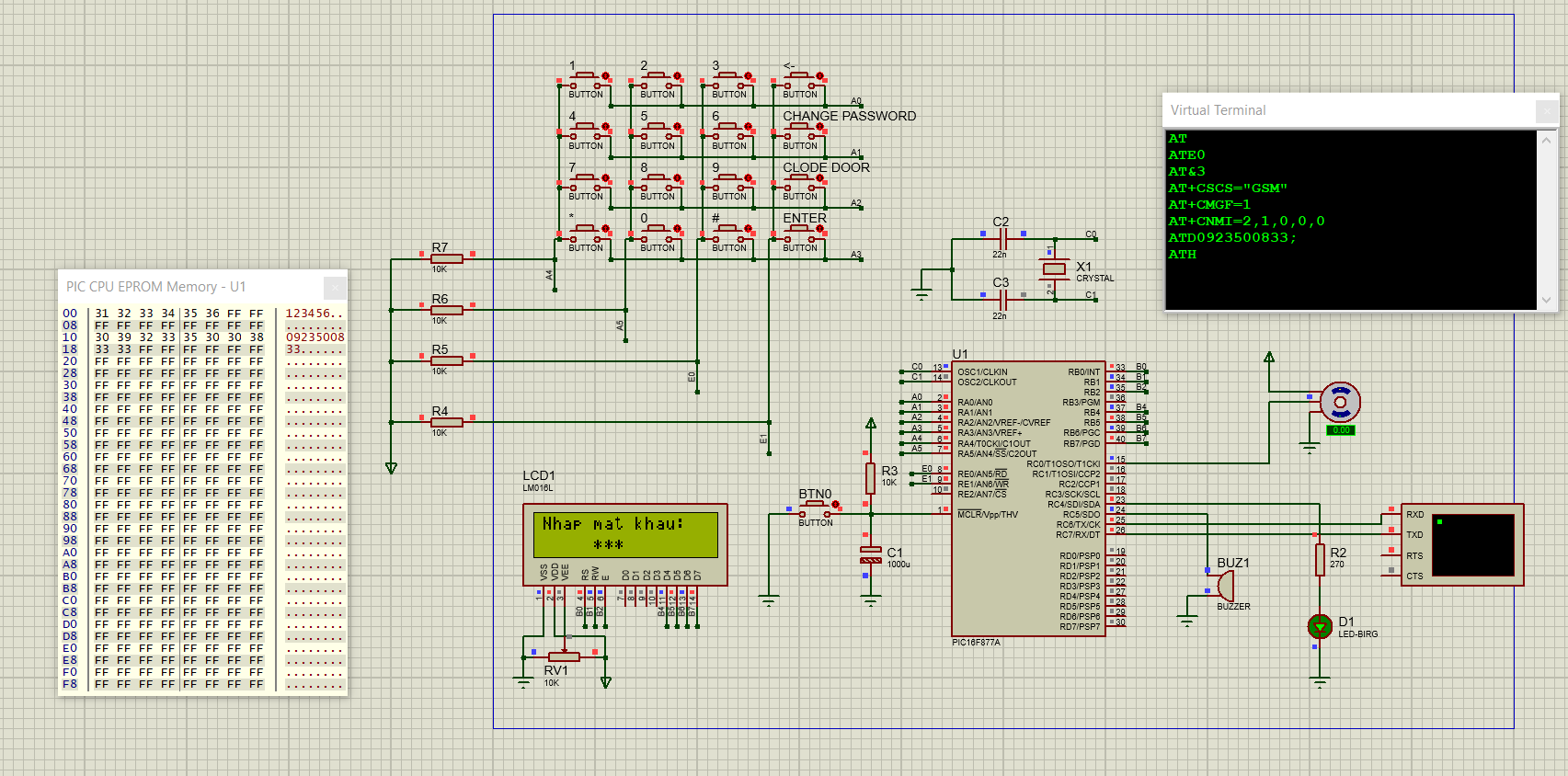
}

}

***//Kết thúc chương trình***

# KẾT QUẢ THỰC HIỆN

* Kết quả mô phỏng



**Hình 3.** *Mô phỏng trên phần mềm Proteus*

* Kết quả thi công



**Hình 4.** *Kết quả thi công*



**Hình 5.** *Kết quả thực tế*

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Qua quá trình thực hiện đề tài, em đúc kết được những kinh nghiệm quý báu, từ đó rút ra những kinh nghiệm và khắc phục những điều thiếu sót.

Trong quá trình thi công mạch in, đã sai sót 3 mạch, lý do vẽ layout chưa kỹ, trong lúc thi công còn khá hập tấp dẫn đến lỗi.

Về những kinh nghiệm đạt được: tìm hiểu được cách hoạt động của module Sim800A mini, cách sử dụng vi điều khiển PIC 16F877A, giải thuật quét phím ma trận bàn phím 4x4, rèn luyện tư duy, nâng cao tay nghề vẽ layout, thi công mạch in, cải thiện kỹ năng khoan và hàn mạch.

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| Thực hiện tốt khả năng báo động: Còi báo reo và nhắn tin (gọi điện) cho người dùng khi phát hiện có người nhập sai mật khẩu quá 5 lần. | So với yêu cầu ban đầu, sản phẩm còn khá đơn giản, cần được phát triển nhiều hơn về sau. |
| Tổng giá trị vật chất của mạch không cao so với công dụng mà nó đem lại (<500k VNĐ). | Quá trình chuẩn bị và thực hiện mạch chưa tốt nên chi phí phải bỏ ra để làm thực sự lớn hơn nhiều. |
| Nhỏ gọn, đơn giản dễ sử dụng; hoạt động tốt; sử dụng nguồn cấp là Adaptor 5V nên tương đối an toàn. | Chịu ảnh hưởng bởi gió và ánh sáng; Có thể bị nhiễu làm cho mạch không hoạt động tốt. |

## Hướng phát triển

Sản phẩm góp phần tích cực vào việc bảo mật khi được gắn tại các vị trí quan trọng như: cửa ra vào, tủ,..

Ngoài việc sử dụng độc lập, ta cũng có thể phát triển mạch hơn nữa, tích hợp trong các thiết bị chống trộm thông minh, góp phần vào sự phát triển không ngừng của xã hội công nghệ hiện nay. Có thể thay bảo mật bằng cách nhập mật khẩu bằng các thiết bị bảo mật hơn như vân tay, nhận diện khuôn mặt,.... Ngoài ra có thể thay module GSM bằng cách thông báo cho người dùng nếu có kết nối mạng internet. Qua đó có thể ứng dụng thêm về IOT.

# TÀI LIỆU THAM KHẢO

1. *Servo SG90 – Cách dùng servo.* Truy cập từ*:* <https://dientu360.com/dong-co-servo-sg90>
2. **Tanatoelectronics.** (2017). *Bạn tự tìm hiểu về PIC: Viết chương trình cho PIC16F84A.* Truy cập từ*:* <https://sites.google.com/site/tanatoelectronic/ban-tu-tim-hieu-ve-pic-viet-chuong-trinh-cho-pic16f84a>
3. **Thạc sĩ Bùi Quốc Bảo.** *Slide bài giảng môn Thiết kế hệ thống nhúng.*
4. **Mlab.***Hướng dẫn sử dụng Module Sim800A Phần 1.*Truy cập từ: <http://mlab.vn/bai-viet-ki-thuat/huong-dan-su-dung-module-gsm-gps-gprs/46225-huong-dan-su-dung-module-sim800a-phan-1.html>

1. [↑](#footnote-ref-1)