**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN-ĐIỆN TỬ**

**BỘ MÔN ĐIỆN TỬ**

**---------------------------**



**BÁO CÁO BÀI TẬP LỚN**

**MÔN LẬP TRÌNH HỆ THỐNG NHÚNG**

**ĐỀ TÀI:**

**SMART SECURITY DOOR LOCK SYSTEM**

**LỚP L02 – NHÓM 05**

**Giảng viên hướng dẫn: T.S Trương Quang Vinh**

**Nhóm sinh viên thực hiện:**

1. **Nguyễn Nhựt Linh 1812819**
2. **Lê Xuân Lãm 1913907**
3. **Phạm Ngọc Lâm 1812777**
4. **Trịnh Quán Lâm 1910300**
5. **Trương Quang Linh 1711986**

***Thành phố Hồ Chí Minh, ngày 13 tháng 5, 2022***

**LỜI CẢM ƠN**

Ngày nay, với những ứng dụng của khoa học kỹ thuật tiên tiến, thế giới của chúng ta đã và đang một ngày thay đổi, văn minh và hiện đại hơn. 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 là những yếu tố rất cần thiết góp phần cho hoạt động của con người đạt hiệu quả.

Điện tử đang trở thành một ngành khoa học đa nhiệm vụ. Nó đã đáp ứng được những nhu cầu cần thiết trong hoạt động đời sống hằng ngày. Một trong những nhu cầu đó là vấn đề về bảo mật, an ninh. Đây là một trong những nhu cầu thiết yếu nhất của con người để bảo mật cho ngôi nhà của mình, hoặc để bảo mật những tài liệu quan trọng của cá nhân. Vì những lí do như trên nên nhóm chúng em quyết định làm một hệ thống để bảo mật nhà cửa đó là hệ thống khóa cửa thông minh “Door Lock System”

Qua quá trình trao đổi thảo luận của các thành viên trong nhóm nên nhóm chúng em quyết định sẽ thực hiện đề tài “**Door Lock System**”.

Trong quá trình học tập và nghiên cứu đề tài “**Door Lock System**” dưới sự giúp đỡ, chỉ bảo nhiệt tình của thầy Trương Quang Vinh để hoàn thành dự án này.

Với tình cảm chân thành, nhóm chúng em bày tỏ lòng biết ơn đối với thầy đã tận tình giảng dạy, giúp đỡ chúng em trong suốt quá trình học tập, nghiên cứu về kiến thức, tài liệu và phương pháp.

Mặc dù đã có nhiều cố gắng để hoàn thành dự án tốt đẹp nhất nhưng dự án có thể không tránh khỏi những thiếu sót. Nhóm chúng em kính mong thầy, các bạn và mọi người quan tâm đến đề tài có những ý kiến đóng góp, giúp đỡ để đề tài được hoàn thiện hơn.

TP Hồ Chí Minh, tháng 5 năm 2022

**Nhóm sinh viên**

MỤC LỤC

[1. GIỚI THIỆU DỰ ÁN 5](#_Toc104500112)

[1.2. Nhiệm vụ đề tài 5](#_Toc104500113)

[1.3. Phân chia công việc trong nhóm 5](#_Toc104500114)

[2. LÝ THUYẾT 8](#_Toc104500116)

[2.1. Giới thiệu về Tiva C 8](#_Toc104500117)

[2.1.1. Giới thiệu về module TM4C123G 8](#_Toc104500118)

[2.1.2. Đặc điểm của module 8](#_Toc104500119)

[2.2. Giới thiệu về LCD 16x2 8](#_Toc104500120)

[2.3. Giới thiệu về Keypad: 9](#_Toc104500121)

[2.3.1. Giới thiệu: 9](#_Toc104500122)

[2.3.2. Cấu tạo và nguyên lý hoạt động: 10](#_Toc104500124)

[3. ĐẶC TẢ HỆ THỐNG 11](#_Toc104500127)

[3.1. Description 11](#_Toc104500128)

[3.2. External enviroment 11](#_Toc104500129)

[3.3. Block diagram 11](#_Toc104500130)

[3.3.1. Block diagram 11](#_Toc104500131)

[3.3.2. Các vấn đề cơ bản của hệ thống 12](#_Toc104500133)

[4. THIẾT KẾ HỆ THỐNG 13](#_Toc104500134)

[4.1. Phần cứng 13](#_Toc104500135)

[4.2. Phần mềm 14](#_Toc104500136)

[5. THỰC HIỆN VÀ ĐÁNH GIÁ 15](#_Toc104500137)

[5.1. Code 15](#_Toc104500138)

[5.2. Setup, Test (chụp ảnh mạch khi nạp code) 15](#_Toc104500139)

1. GIỚI THIỆU DỰ ÁN
   1. **Tổng quan dự án**

* Hệ thống này là hệ thống bảo mật sẽ yêu cầu người sử dụng nhập mật khẩu để có thể mở cửa với mật khẩu được chủ nhà đặt sẵn từ trước và có thể thay đổi được. Từ đó, người chủ nhà sẽ không cần phải lo có người lạ vào nhà nếu không biết mật khẩu.
* Hệ thống sẽ thông qua 1 con chip xử lý nhận tín hiệu điện từ keypad và kiểm tra xem người đó có nhập đúng mật khẩu hay không. Nếu đúng, tín hiệu điện sẽ kích hoạt rơ le điện làm cho cửa mở và đèn sáng. Nếu không đúng thì hệ thống sẽ mời nhập lại, nếu không đúng 3 lần thì hệ thống sẽ phát loa tín hiệu để chủ nhà biết là có người lạ đang cố gắng đột nhập
  1. Nhiệm vụ đề tài

Qua tham khảo các nhu cầu trên thị trường thì yêu cầu của một bộ phận khách hàng đối với sản phẩm là:

* Giao diện người dùng dễ sử dụng.
* Hệ thống phải có tính bảo mật cao.
* Hệ thống phải làm việc được ngay cả khi mất điện.
* Hệ thống nhỏ gọn.
* Điện áp hoạt động của hệ thống không gây nguy hiểm đến người sử dụng.
  1. Phân chia công việc trong nhóm

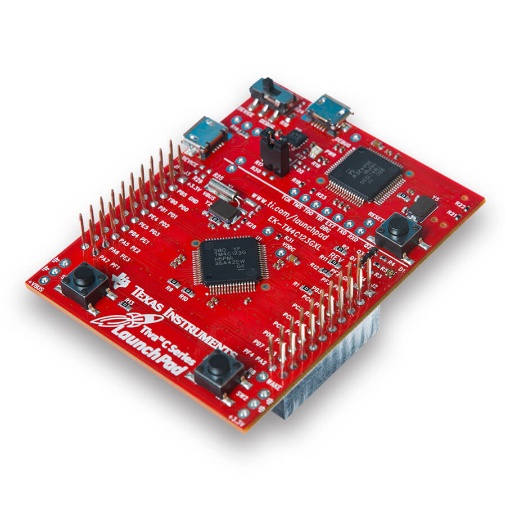
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nhóm: 0 –Door Lock System | | | | Date: 01/3/2022 |
| **Thành viên** | | **Vai trò** | **Đặc điểm** | |
| Lê Xuân Lãm | | Thi công mạch, nhóm trưởng |  | |
| Phạm Ngọc Lâm | | Thiết kế phần cứng |  | |
| Trịnh Quán Lâm | | Thiết kế phần mềm, debug |  | |
| Trương Quang Linh | | Kiểm tra mạch |  | |
| Nguyễn Nhựt Linh | | Viết báo cáo |  | |
| **Nhiệm vụ:** | | | **Thành viên tương ứng** | |
| Triển khai ý tưởng, góp ý thảo luận 🡪 Hoàn thành 20% project | | | Tất cả thành viên | |
| Thiết kế sơ đồ Schemetic | | | Q. Lâm+ X.Lãm | |
| Lập sơ đồ giải thuật, Lập trình phần mềm | | | X.Lãm, N.Linh, Q.Lâm | |
| Lắp mạch thực tế, kiểm tra | | | Tất cả thành viên | |
| Train team chuẩn bị báo cáo | | | All | |
| **Thảo luận nhóm**: Thứ 7 hằng tuần (Google meet & Record) | | | 20h 🡪 21h30 | |
| **Qui tắc nhóm** | 1.Tham gia các buổi thảo luận đầy đủ(Vắng- có lý do rõ ràng).  2. Hoàn thành các công việc được giao trước deadline.  3. Cả nhóm hợp tác cùng hoàn thành project.  4. Lắng nghe và đưa ra ý kiến một cách khách quan. | | | |

* + Kế hoạch dự án

|  |  |  |  |
| --- | --- | --- | --- |
| **DOOR LOCK SYSTEM** | | | |
| Team |  | | |
| Tên sản phẩm | Hệ thống khóa cửa thông minh | | |
| Function | Bảo vệ nhà cửa không bị xâm phạm | | |
| Ước tính thời gian hoàn thành | 3 tháng/ 5h 1 tuần  Start: 1/3/2022 End: 20/5/2022 | | |
| Team member | 1. **Lê Xuân Lãm 1913907** 2. **Phạm Ngọc Lâm 1812777** 3. **Trịnh Quán Lâm 1910300** 4. **Trương Quang Linh 1711986** 5. **Nguyễn Nhựt Linh 1812819** | | |
| Thời gian biểu | | | |
| **Tiến độ** | Tháng đầu | Tháng thứ 2 | Tháng cuối |
| Triển khai ý tưởng, góp ý thảo luận |  |  |  |
| Lập bảng đặc tả hệ thống với 10 mục |  |  |  |
| **Thiết kế phần cứng** |  |  |  |
| Thiết kế sơ đồ Schemetic |  |  |  |
| Tính toán số liệu các linh kiện tối ưu nhất |  |  |  |
| **Thiết kế phần mềm** |  |  |  |
| Lập lưu đồ giải thuật, Nghiêm cứu code |  |  |  |
| Lập trình source code để nạp thử nghiệm và kiểm tra tính thực tế |  |  |  |
| Hoàn thành phần mềm |  |  |  |
| Tiến hành thi công mạch và thử nghiệm đánh giá sản phẩm |  |  |  |
| **Tổng hợp và báo cáo** |  |  |  |
| Lập bảng báo cáo chi tiết |  |  |  |
| Hoàn thành dự án |  |  |  |

1. LÝ THUYẾT
   1. Giới thiệu về Tiva C
      1. Giới thiệu về module TM4C123G

* Kit Tiva C TM4C123G là kit có chi phí thấp dùng cho các dòng vi điều khiển họ ARM-Cortex



Hình 2.1: Kit TM4C123G

* + 1. Đặc điểm của module
* Cổng mini USB trên board mạch để kết nối giao tiếp với máy tính.
* Bộ vi điều khiển ARM-cortex kèm với một số tính năng hẹn giờ, bộ đếm, ngắt, chân PWM, CPU, chân I/O và dựa trên xung nhịp giúp tạo ra nhiều tần số và số lệnh hơn trong mỗi chu kì.
  1. Graphical user interface

     Description automatically generatedGiới thiệu về LCD 16x2

Hình 2.4: Màn hình LCD 16\*2

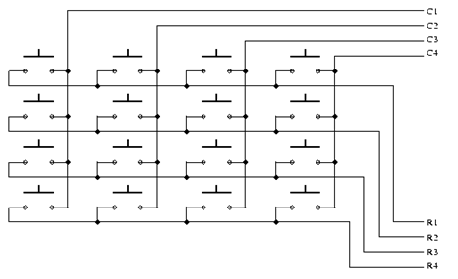
* Chân 1 là VSS tương đương với GND
* Chân 2 là VDD tương đương với VCC
* Chân 3 là VE dùng để chỉnh độ sáng màn hình.
* Chân 4 là Register Select dùng để điều khiển địa chỉ nào sẽ ghi dữ liệu.
* Chân 5 là Read/Write phụ thuộc vào giá trị gửi vào để đọc hay ghi dữ liệu.
* Chân 6 là Enable cho phép ghi vào LCD.
* Chân 7 đến 14 (Data 0 đến Data 7) là 8 chân dữ liệu, mỗi chân sẽ có giá trị HIGH hoặc LOW nếu đang ở chế độ đọc hoặc ghi.
* Chân 15 và 16 là chân Backlight anode và cathode dùng để bật tắt đèn màn hình LCD.
  1. Giới thiệu về Keypad:
     1. Giới thiệu:



Hình 2.3.1: Keypad

* KeyPad là một thiết bị nhập chứa các nút bấm cho phép người dùng nhập các chữ số, chữ cái hoặc ký tự điều khiển. KeyPad không chứa tất cả bảng mã ASCII như keyboard vì thế nó thường được sử dụng trong các ứng dụng chuyên dụng và tương đối đơn giản, ở đó, số lượng nút nhấn thay đổi phụ thuộc vào ứng dụng.
* KeyPad 4×4 là bàn phím gồm 16 nút nhấn, được xếp thành 4 hàng, mỗi hàng gồm 4 phím bấm như hình dưới đây.
  + 1. Cấu tạo và nguyên lý hoạt động:

Cấu tạo:



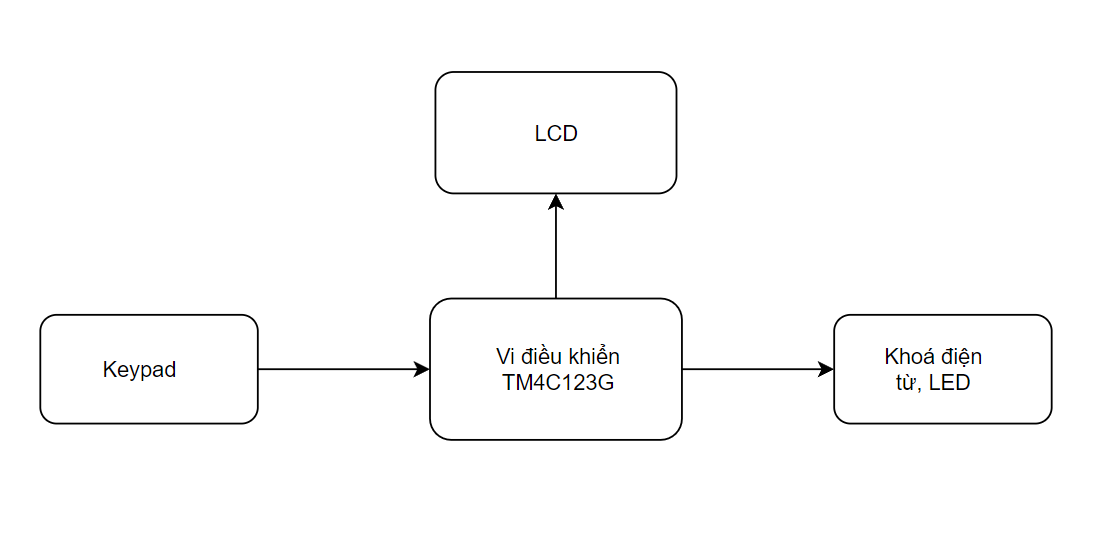
Hình 2.3.2: Cấu tạo của KeyPad 4×4

**Nguyên lý hoạt động:**

* Để làm việc với KeyPad 4×4, người lập trình thường sử dụng giải thuật “quét phím”. Giải thuật này yêu cầu VĐK liên tục đưa các tín hiệu đầu ra ở hàng (hoặc cột) và thu lại đầu vào ở cột (hoặc hàng), nếu phím được bấm, đầu phát tín hiệu sẽ được kết nối với đầu thu, từ đó xác định được phím đã bấm.
* Việc lựa chọn đầu ra/vào hình thành 2 phương pháp quét phím: theo chiều dọc và theo chiều ngang. Trong báo cáo này, tín hiệu xuất ra ở các hàng và thu lại ở các cột.
* Giả sử một nút ‘2’ được nhấn, khi đó đường C và 2 được nối với nhau. Nếu đường C được nối với GND, khi đó, điện áp ở chân số 2 sẽ mang điện áp 0V. Tương tự như thế với các phím cùng hàng C.

1. ĐẶC TẢ HỆ THỐNG
   1. Description

* Nhập mật khẩu từ keypad, mật khẩu nhập vào sẽ được vi điều khiển xử lí và hiển thị lên LCD. Nếu mật khẩu không đúng LCD sẽ hiện thị cho phép nhập lại mật khẩu. Ngược lại nếu mật khẩu chính xác, led sẽ sáng, khóa mở. Sau đó có thể thực hiện đổi mật khẩu của khoá bằng cách giữ nút # 5 giây.
  1. External enviroment
* Có thể ngoài trời hoặc trong nhà (Có thể lấy nguồn điện trực tiếp từ máy tính hoặc bộ tích nguồn 5V).
  1. Block diagram
     1. Block diagram



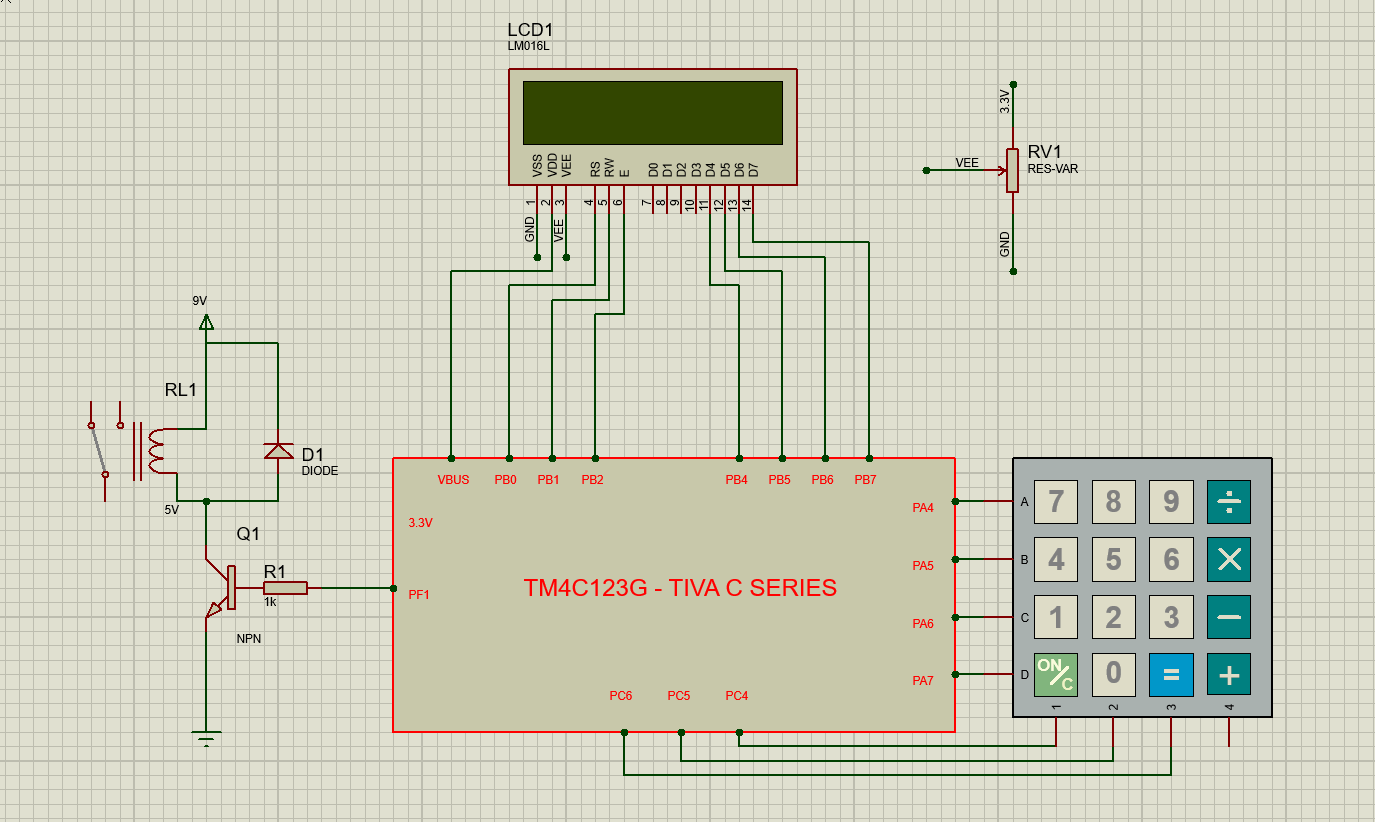
**Mô tả các khối**

* Khối Keypad: Khối tín hiệu vào cho phép nhập kí tự password
* Khối vi điều khiển: Nhận thông tin và xử lí, hiển thị trên LCD ở đề tài này ta sử dụng Tiva C series TM4C123G.
* Khối hiện thị: Sử dụng LCD hiển thị trạng thái của hệ thống và dữ liệu nhập vào.
* Khoá điện từ: Khi đúng mật khẩu gửi tín hiệu mở khoá đồng thời LED sáng, ngược lại mật khẩu không đúng không gửi tín hiệu mở khoá, LED tắt.
  + 1. Các vấn đề cơ bản của hệ thống

|  |  |  |
| --- | --- | --- |
| **STT** | **Vấn đề** | **Mô tả** |
| 1 | Ràng buộc | - Giá thành không quá đắt (< 500.000VNĐ)  - Thời gian sử dụng trên 1 năm  - Kích thước nhỏ (< 20x20x10cm) |
| 2 | Chức năng | - Led sáng, mở khóa khi đúng mật khẩu và ngược lại |
| 3 | Real-time | - Hệ thống thuộc loại sort real-time, vì từ lúc người dùng nhật mật khẩu đúng cho tới khi đèn Led được bật, thời gian trễ tối đa cho phép lên đến 0.5 ~ 1 giây |

1. THIẾT KẾ HỆ THỐNG
   1. Phần cứng

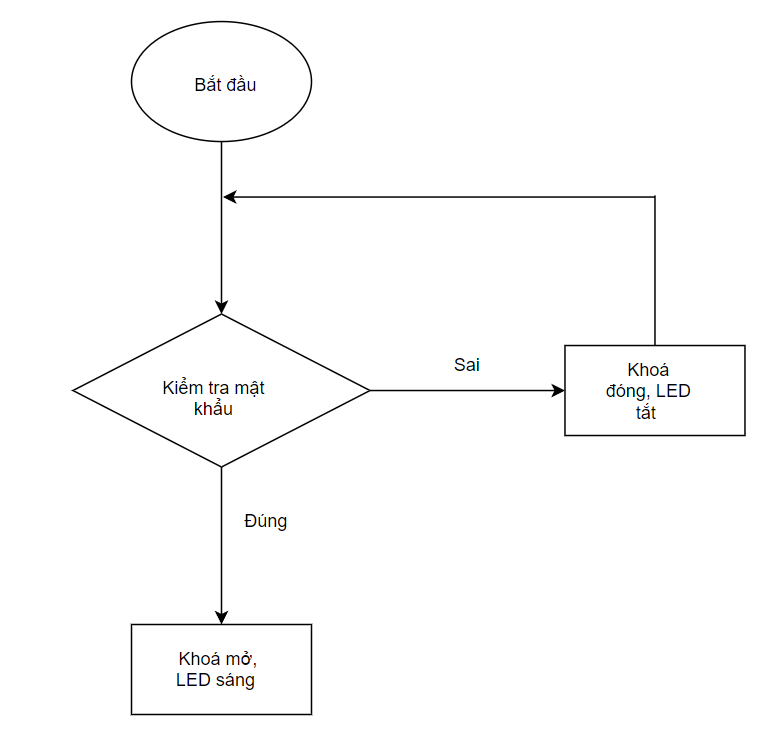
**Schematic** (đính kèm sơ đồ chân Tiva C series TM4C123G)



|  |  |  |
| --- | --- | --- |
| **STT** | **Phần cứng** | **Mô tả** |
| 1 | TM4C123G | Vi xử lý chính của hệ thống |
| 2 | Keypad 3x4 | Cho phép nhập password từ người dùng |
| 3 | LCD | Hiển thị trạng thái hệ thống |
| 4 | Relay 5V | Điều khiển khoá điện từ khi nhận tín hiệu từ vi xử lý |

* 1. Phần mềm

**Lưu đồ giải thuật**



**Giải thích giải thuật**

Bắt đầu người thực nhập mật khẩu trên keypad, hệ thống sẽ hoạt động như sau:

* Mật khẩu đúng: Led sáng cảnh báo khóa đã mở thành công
* Mật khẩu sai: Led không sáng, và trờ về bước yêu cầu nhập mật khẩu

1. THỰC HIỆN VÀ ĐÁNH GIÁ
   1. Code

Main.c:

#include <stdint.h>

#include <stdbool.h>

#include "inc/hw\_types.h"

#include "inc/hw\_memmap.h"

#include "driverlib/sysctl.h"

#include "driverlib/gpio.h"

#include "inc/hw\_gpio.h"

#include "driverlib/interrupt.h"

#include "inc/hw\_ints.h"

#include "LCD.h"

#include "Keypad\_Password.h"

int main()

{

SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ| SYSCTL\_OSC\_MAIN);

has\_lcd\_4bitsetup(); //setup lcd

KeyPad\_Lcd\_Setup();

KeyPad\_Setup();

while(1)

{

KeyPad\_wait();

KeyPad\_PassWord\_Confirm();

}

}

Keypass\_Password.c

#include <stdint.h>

#include <stdbool.h>

#include "inc/hw\_types.h"

#include "inc/hw\_memmap.h"

#include "driverlib/sysctl.h"

#include "driverlib/gpio.h"

#include "inc/hw\_gpio.h"

#include "driverlib/interrupt.h"

#include "inc/hw\_ints.h"

#include "Keypad\_Password.h"

#include "LCD.h"

uint8\_t Password[4]="0000"; //password mac dinh

uint8\_t EnteredPassword[4];

uint8\_t col=4; // dung trong ham keypad\_wait open and close the pins of columns in the order

uint8\_t inttertuptvalue=0; //dung de xac dinh pin nao ngat

uint8\_t X;//gia tri hang keypad

uint8\_t Y;//gia tri cot keypad

uint8\_t Enterd\_Password\_Counter=0;// bien dem so ky tu dang duoc nhap

uint8\_t Password\_Confirm\_Counter=0;// bien dem so ky tu trong password

uint8\_t NewPassword\_Counter=0;

bool NewPasswordControl=0;

bool CorrectPasswordWait=0;

/\*

\* the 3\*4 Keypad

\* \*/

uint8\_t MKeyPad[4][3]= {

{'1','2','3'},

{'4','5','6'},

{'7','8','9'},

{'\*','0','#'}

};

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void KeyPad\_Lcd\_Setup()

{

has\_lcd\_erase();

has\_lcd\_write(1,1,"Enter the ");

has\_lcd\_write(2,1,"Password:");

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void KeyPad\_PassWord\_Confirm()

{

uint8\_t b=0;//dung trong vong lap

/\*

\* neu password da duoc reset thi (CounterNewPassword==4) tra gia tri true

\*

\* \*/

if(NewPassword\_Counter==4)

{

NewPasswordControl=0;

NewPassword\_Counter=0; //Reset ve 0 neu password duoc reset nhieu lan

has\_lcd\_erase();

has\_lcd\_write(1,1,"Password has");

has\_lcd\_write(2,1,"been changed");

SysCtlDelay(SysCtlClockGet()/2);

KeyPad\_Lcd\_Setup();//chuan bi nhap password sau khi reset

}

/\*

\* else kiem tra password da nhap dung hay sai

\* bang cach so sanh Password mac dinh voi EnteredPassword

\*

\* \*/

else{

for(b=0;b<4;b++){

if(EnteredPassword[b]==Password[b]){

Password\_Confirm\_Counter++;

}

}

/\* if Entered Password dung

\* in this statement PF1(RELAY or RGB) will be set to 1

\* CorrectPasswordWait se duoc set to 1 khong cho phep nhap password nua ngoai tru reset password (xen ham void KeyPad\_int())

\* \*/

if(Password\_Confirm\_Counter==4)

{

has\_lcd\_erase();

has\_lcd\_write(1,1,"Correct Password");

SysCtlDelay(SysCtlClockGet()/2);

CorrectPasswordWait=1;

has\_lcd\_erase();

has\_lcd\_write(1,1,"Door is open ");

has\_lcd\_write(2,1,"LED is lighting");

GPIOPinWrite(GPIO\_PORTF\_BASE,GPIO\_PIN\_2,4);

}

/\*

\* else if Entered Password sai thi try again

\* \*/

else

{

has\_lcd\_erase();

has\_lcd\_write(1,1,"incorrect Passwo");

has\_lcd\_write(2,1,"Try again");

SysCtlDelay(SysCtlClockGet()/2);

KeyPad\_Lcd\_Setup();

}

Enterd\_Password\_Counter=0;

Password\_Confirm\_Counter=0;

}

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

/\*

\* to calculate 2^x dung de quet keypad

\* \*/

uint8\_t power(uint8\_t x)

{

uint8\_t y=1;

uint8\_t i=0;

for(i=0;i<x;i++)

{

y=y\*2;

}

return y;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void KeyPad\_wait()

{

/\*

\* while the digits of password (the correct password or Enterd password )is lease than 4 digit

\* the operation of set-clear columns pins in order will countinues

\*/

while((Enterd\_Password\_Counter<4)&&(NewPassword\_Counter<4)){

for(col=4;col<=6;col++){

GPIOPinWrite(GPIO\_PORTC\_BASE,GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6,112-power(col));//look to the comment below

/\*

\*(2^4)+(2^5)+(2^6)=112

\*(112-2^COl) operation sets the pins(4-7)to 1(logic) except the pin of COL ( if Col=4 --->PIN4=0(logic) )

\*/

SysCtlDelay(20000);//by testing 20000 MS is a good duration between Entering the digits

}

}

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void NewPasswordWait()

{

/\*

\* to Clear the pin of Relay(or RGB RED) and'CorrectPasswordWait' if the password is changed after Turning on the PIN(PF1)

\*

\* \*/

GPIOPinWrite(GPIO\_PORTF\_BASE,GPIO\_PIN\_1,0);

CorrectPasswordWait=0;

/\*

\* Preparing LCD to New Password

\* \*/

has\_lcd\_erase();

has\_lcd\_write(1,1,"Enter new pass");

has\_lcd\_position(2,1);

NewPasswordControl=1;//To determine the entered char if it belongs to the Password or to a Entered Passowrd

}

/////////////////////////////////////////////////////////////////////////////////////

/\*

\* to take log2 of x

\* \*/

uint8\_t ln(uint8\_t x)

{

uint8\_t y=0;

while(x!=1)

{

x=x/2;

y++;

}

return y;

}

///////////////////////////////////////////////////////////////////////////////

void KeyPad\_int()

{

/\*

\* when on of pins4to7 of PORTA is fulling to 0(interrupting) the Program Counter will Indicate to this function

\*

\*/

inttertuptvalue=GPIOIntStatus(GPIO\_PORTA\_BASE,false); //which pin is interrupted (EX: if pin4---> inttertuptvalue=16)

X=ln(inttertuptvalue)-4;//look below

/\*

\* taking log2 of inttertupvalue will give the pins number

\* and then shifting by -4 will give the real number of Keypads row

\*/

Y=col-4;//shifting (EX 4 to 0 , 7 to 3)will give the real number of Keypads column

/\*

\* if the System waits a password (CorrectPasswordWait==0) the entered charecter will be sent to LCD

\*

\* \*/

if(!CorrectPasswordWait){

has\_lcd\_sendchar(MKeyPad[X][Y]);

}

SysCtlDelay(SysCtlClockGet()/4);//choosen by testing

/\*

\* this 8 lines below are written to detect the holding of #

\* that will allow to change a virtual password

\* \*/

if (MKeyPad[X][Y]=='#'){

SysCtlDelay(20000);

/\*

\* @if the holding of # crosses a specific duration the PC will go to NewPasswordWait() function

\* \*/

if(GPIOPinRead(GPIO\_PORTA\_BASE,inttertuptvalue)==0){

SysCtlDelay(SysCtlClockGet());

while(GPIOPinRead(GPIO\_PORTA\_BASE,inttertuptvalue)==0){

NewPasswordWait();

}

}

/\*

\* @else '#' will be a typical charecter

\* \*/

else if(!CorrectPasswordWait){

PasswordTypeControl();

}

}

/\*

\* if '#' is not has been entered in the first place

\* \*/

else if(!CorrectPasswordWait){

PasswordTypeControl();

}

GPIOIntClear(GPIO\_PORTA\_BASE,GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7);

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void PasswordTypeControl()

{

/\*

\* if MKeyPad belongs to password(NewPasswordControl==1) send it to passwords array

\* else send it to EnterdPasswords array

\* \*/

if(NewPasswordControl){

Password[NewPassword\_Counter]=MKeyPad[X][Y];

NewPassword\_Counter++;

}

else{

EnteredPassword[Enterd\_Password\_Counter]=MKeyPad[X][Y];

Enterd\_Password\_Counter++;

}

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void KeyPad\_Setup()

{

IntMasterEnable();

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);//Enable the peripheral of GPIOPORT F

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOC);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOA);

GPIOPinTypeGPIOOutput(GPIO\_PORTC\_BASE, GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7); //PC4 to PC7 are columns output

GPIOPinTypeGPIOOutput(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3);//look to the comment below

/\*

\* PF1 to PF3 can be used as a controlled pins of RELAY its also connected to RGB LEDs of TIVE C kit

\* \*/

GPIOPinWrite(GPIO\_PORTC\_BASE,GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7,0xFF);//look to the comment below

/\*

\* GPIO\_PORTC pin4 to pin 7 are connected to the columun pins of Keypad

\* they will sets to 1 and Keypad\_wait fun. will clear its one by one in order leaving the others set to 1

\*

\* \*/

GPIOPinTypeGPIOInput(GPIO\_PORTA\_BASE,GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7);// look to the comment below

/\*

\*GPIO PORTA pin4 to 7 are connected to the Rows pins of keypad

\*they will set as input to perceive the sinyal that will come from Keypad

\*

\* \*/

GPIOPadConfigSet(GPIO\_PORTA\_BASE, GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7, GPIO\_STRENGTH\_4MA, GPIO\_PIN\_TYPE\_STD\_WPU);// look to the comment below

/\*

\* GPIO PORTA pin4 to 7 will sets to 1 and trigger to 1-0 fulling edge

\* \*/

GPIOIntRegister(GPIO\_PORTA\_BASE,KeyPad\_int);//Register the Function of interrupting

GPIOIntTypeSet(GPIO\_PORTA\_BASE,GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7,GPIO\_FALLING\_EDGE);//trigger to 1-0 fulling edge

GPIOIntClear(GPIO\_PORTA\_BASE, GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7);

GPIOIntEnable(GPIO\_PORTA\_BASE, GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7);

IntEnable(INT\_GPIOF\_TM4C123);

LCD.c

#include <stdint.h>

#include <stdbool.h>

#include "inc/hw\_types.h"

#include "inc/hw\_memmap.h"

#include "driverlib/sysctl.h"

#include "driverlib/gpio.h"

#include "inc/hw\_gpio.h"

#include "driverlib/pin\_map.h"

#include "LCD.h"

/////////////////////////////////////////////////////////////////////////////////////

void has\_lcd\_switchwrite()

{ SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = GPIO\_LOCK\_KEY;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_CR) |= 0x01;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = 0;

GPIOPinTypeGPIOInput(GPIO\_PORTF\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_4); //pin0 ve pin4 input

GPIOPadConfigSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_4, GPIO\_STRENGTH\_4MA, GPIO\_PIN\_TYPE\_STD\_WPU); // pin0 ve pin4 pull-up

unsigned char a=0x2f;

unsigned char line=1;

unsigned char column=1;

while(1)

{

if(GPIOPinRead(GPIO\_PORTF\_BASE,GPIO\_PIN\_0)==0)

{ while(GPIOPinRead(GPIO\_PORTF\_BASE,GPIO\_PIN\_0)==0){}

has\_lcd\_position(line,column);

has\_lcd\_sendchar(a);

a++;

has\_lcd\_position(line,column);

}

if(GPIOPinRead(GPIO\_PORTF\_BASE,GPIO\_PIN\_4)==0)

{ while(GPIOPinRead(GPIO\_PORTF\_BASE,GPIO\_PIN\_4)==0){}

a=0x2f;// '/' itibaren baslar

if((line==1) && (column==16))

{

column=0;

line=2;

}

if((line==2) && (column==16))

{

column=0;

line=1;

}

has\_lcd\_position(line,++column);

}

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void has\_lcd\_erase()

{

has\_lcd\_sendcommand(0x01);

SysCtlDelay(1000);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void has\_lcd\_write(uint8\_t line , uint8\_t column,char str[])

{

has\_lcd\_position(line,column);

int i=0;

while((str[i])!=0)

{

has\_lcd\_sendchar(str[i]);

i++;

}

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void has\_lcd\_sendchar(uint8\_t ch)

{

GPIOPinWrite(LCDPORT, highpin, ch & 0xf0 );

GPIOPinWrite(LCDPORT, RS, 0x01);

GPIOPinWrite(LCDPORT, E, 0x04);

SysCtlDelay(10);

GPIOPinWrite(LCDPORT, E, 0x00);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT,highpin,( ch & 0x0f)<<4);

GPIOPinWrite(LCDPORT, RS, 0x01);

GPIOPinWrite(LCDPORT, E, 0x04);

SysCtlDelay(10);

GPIOPinWrite(LCDPORT, E, 0x00);

SysCtlDelay(50000);

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void has\_lcd\_sendcommand(uint8\_t x)

{

GPIOPinWrite(LCDPORT, highpin, x & 0xf0 );

GPIOPinWrite(LCDPORT, RS, 0x00);

GPIOPinWrite(LCDPORT, E, 0x04);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT, E, 0x00);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT, highpin,( x & 0x0f)<<4);

GPIOPinWrite(LCDPORT, RS, 0x00);

GPIOPinWrite(LCDPORT, E, 0x04);

SysCtlDelay(100);

GPIOPinWrite(LCDPORT, E, 0x00);

SysCtlDelay(50000);

}

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void has\_lcd\_position(uint8\_t line , uint8\_t column)

{

if(line==1)

{

has\_lcd\_sendcommand(0x80+(column-1));

}

else

{

has\_lcd\_sendcommand(0xc0+(column-1));

}

SysCtlDelay(100);

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void has\_lcd\_4bitsetup()

{

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOB);

GPIOPinTypeGPIOOutput(LCDPORT,0x0ff);//make all pin output

SysCtlDelay(5000000);

GPIOPinWrite(LCDPORT, RW, 0x00 );

SysCtlDelay(5000000);

GPIOPinWrite(LCDPORT, RS, 0x00 );

GPIOPinWrite(LCDPORT, highpin, 0x30 );

GPIOPinWrite(LCDPORT, E, 0x04);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT, E, 0x00);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT, highpin, 0x30 );

GPIOPinWrite(LCDPORT, E, 0x04);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT, E, 0x00);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT,highpin, 0x30 );

GPIOPinWrite(LCDPORT, E, 0x04);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT, E, 0x00);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT, highpin, 0x20 );

GPIOPinWrite(LCDPORT, E, 0x04);

SysCtlDelay(50000);

GPIOPinWrite(LCDPORT, E, 0x00);

has\_lcd\_sendcommand(0x28);//4bit

has\_lcd\_sendcommand(0x06);

has\_lcd\_sendcommand(0x0c);

SysCtlDelay(50000);

}

* 1. Setup, Test (chụp ảnh mạch khi nạp code)



* + 1. Đánh giá chất lượng hệ thống
* Ưu điểm
* Hệ thống hoạt động liên tục, độ chính xác khá cao.
* Chi phí thực hiện rẻ.
* Thời gian xử lí nhanh, nên dữ liệu được cập nhật liên tục.
* Các giải thuật phần mềm tương đối dễ sử dụng.
* Nhược điểm:
* Chỉ dừng lại ở tính chất mô hình, chưa thể phát triển ra thực tế
* Những khó khăn trong quá trình thực hiện.
* Các module có nhiêu cách sử dụng cũng như sử dụng các hạn mức điện áp khác nhau nên rất dễ hư hỏng.
* Gặp khó khăn trong việc giao tiếp với module sim do lần đầu sử dụng.module sim, khi thử nghiệm thực tế thì nhận thấy việc gửi tin nhắn liên tục không ổn định, do đó nhóm quyết định thay thế bằng hình thức gọi điện.
* Chưa ứng dụng được hệ điều hành RTOS vào hệ thống