## Mã nguồn (Source Code)

**\*Code hàm main:**

int main(void)

{

/\* USER CODE BEGIN 1 \*/

uint32\_t TimeCount;

/\* USER CODE END 1 \*/

/\* MCU Configuration \*/

/\* Reset of all peripherals, Initializes the Flash interface and the Systick. \*/ HAL\_Init();

/\* USER CODE BEGIN Init \*/

/\* USER CODE END Init \*/

/\* Configure the system clock \*/ SystemClock\_Config();

/\* USER CODE BEGIN SysInit \*/

/\* USER CODE END SysInit \*/

/\* Initialize all configured peripherals \*/ MX\_GPIO\_Init(); MX\_USART1\_UART\_Init(); MX\_USART3\_UART\_Init();

/\* USER CODE BEGIN 2 \*/

LCD\_Init();

Flash\_Write\_Array(ADDRESS\_DATA\_STORAGE,(uint8\_t\*)password,6);//LUU MK THUONG Flash\_Read\_Array(ADDRESS\_DATA\_STORAGE,(uint8\_t\*)password,6);// doc mk thuong tu bo nho flash Flash\_Write\_Array(ADDRESS\_DATA\_STORAGE1,(uint8\_t\*)master\_pass,6);//LUU MK MASTER Flash\_Read\_Array(ADDRESS\_DATA\_STORAGE1,(uint8\_t\*)master\_pass,6);// DOC MK MASTER HAL\_UART\_Receive\_IT(&huart3, (uint8\_t \*)reply, sizeof(reply)); CurrentNumberFinger=GetNumberOfFinger();

if(CurrentNumberFinger>100)

{

CurrentNumberFinger=1;

}

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

while (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

LCD\_SetPos(0,0); LCD\_String("KHOA CUA NHOM 66");

LCD\_SetPos(0, 1); LCD\_String(" "); key = KEYPAD\_Read();

if (strlen(reply) > 0)

{

}

else

{

char \*pass\_start = strstr(reply, master\_pass); if (pass\_start != NULL)

check\_module\_sim\_to\_Open(); else check\_module\_sim\_to\_Close();

if(key ==0)

{

check\_finger\_to\_Open();

}

else

{

switch (key) {

case '\*':

case 'B':

switch\_to\_selection\_mode(); break;

HandleKeyPress();

case 'C':

case 'A':

case 'D':

default:

}

}

break;

xoa\_van\_tay(); break;

xoa\_tat\_ca\_van\_tay(); break;

mat\_khau(); break;

break;

}

}

/\* USER CODE END 3 \*/

}

**\*Code khai báo:**

#include "main.h" #include <stdio.h> #include <string.h> #include "lcd.h" #define FP\_OK 0x00

#define FP\_ERROR 0xFE #define FP\_NOFINGER 0x02

#define FP\_FINGER\_NOTMATCH 0x0A #define FP\_FINGER\_NOTMATCH 0x0A #define FP\_FINGER\_NOTFOUND 0x09 #define FINGERPRINT\_DATA\_SIZE 9

#define FINGERPRINT\_SIZE 256 // Dung lu?ng c?a m?i m?u vân tay #define FLASH\_START\_ADDR (0x08000000+ 120\*1024)

#define ADDRESS\_DATA\_STORAGE (0x08000000+ 121\*1024) #define ADDRESS\_DATA\_STORAGE1 (0x08000000+ 122\*1024)

UART\_HandleTypeDef huart1; UART\_HandleTypeDef huart3;

uint8\_t FPHeader[6]={0xEF,0x01,0xFF,0xFF,0xFF,0xFF}; uint8\_t FPGetImage[6]={0x01,0x00,0x03,0x01,0x00,0x05};

uint8\_t FPCreateCharFile1[7]={0x01,0x00,0x04,0x02,0x01,0x00,0x08}; uint8\_t FPCreateCharFile2[7]={0x01,0x00,0x04,0x02,0x02,0x00,0x09}; uint8\_t FPCreateTemplate[6]={0x01,0x00,0x03,0x05,0x00,0x09}; uint8\_t FPDeleteAllFinger[6]={0x01,0x00,0x03,0x0D,0x00,0x11};

uint8\_t FPSearchFinger[11]={0x01,0x00,0x08,0x04,0x01,0x00,0x00,0x00,0x40,0x00,0x4E}; uint8\_t FPGetNumberOfFinger[6]={0x01,0x00,0x03,0x1D,0x00,0x21};

char master\_pass[] ="222222";

char password[] = "111111"; // mat khau mac dinh char inputPassword[7] = ""; // mang de luu mat khau int passwordIndex = 0; // index hien thi trong mat khau int password\_index = 0;

char input\_ID\_finger[7]= ""; int ID\_Index = 0;

uint8\_t IDFromFinger;

uint8\_t CurrentNumberFinger; uint8\_t key =0;

char Data\_Goi[30]="ATD0374944995;\r\n"; char reply[60];

char message22[50]; uint8\_t buffer\_index = 0; int module\_sim = 0;

**\*Code đọc phím nhấn:**

/\* Rows ~ OUTPUT \*/

/\* Row 1 default \*/

#define KEYPAD\_ROW\_1\_PORT GPIOA #define KEYPAD\_ROW\_1\_PIN GPIO\_PIN\_4

/\* Row 2 default \*/

#define KEYPAD\_ROW\_2\_PORT GPIOA #define KEYPAD\_ROW\_2\_PIN GPIO\_PIN\_5

/\* Row 3 default \*/

#define KEYPAD\_ROW\_3\_PORT GPIOA #define KEYPAD\_ROW\_3\_PIN GPIO\_PIN\_6

/\* Row 4 default \*/

#define KEYPAD\_ROW\_4\_PORT GPIOA #define KEYPAD\_ROW\_4\_PIN GPIO\_PIN\_7

/\* Columns ~ INPUT \*/

/\* Column 1 default \*/

#define KEYPAD\_COLUMN\_1\_PORT GPIOA

#define KEYPAD\_COLUMN\_1\_PIN GPIO\_PIN\_0

/\* Column 2 default \*/

#define KEYPAD\_COLUMN\_2\_PORT GPIOA

#define KEYPAD\_COLUMN\_2\_PIN GPIO\_PIN\_1

/\* Column 3 default \*/

#define KEYPAD\_COLUMN\_3\_PORT GPIOA

#define KEYPAD\_COLUMN\_3\_PIN GPIO\_PIN\_2

/\* Column 4 default \*/

#define KEYPAD\_COLUMN\_4\_PORT GPIOA

#define KEYPAD\_COLUMN\_4\_PIN GPIO\_PIN\_3

/\* Row pins are output \*/

#define SET\_ROW\_1\_HIGH HAL\_GPIO\_WritePin(KEYPAD\_ROW\_1\_PORT, KEYPAD\_ROW\_1\_PIN, GPIO\_PIN\_SET) #define SET\_ROW\_2\_HIGH HAL\_GPIO\_WritePin(KEYPAD\_ROW\_2\_PORT, KEYPAD\_ROW\_2\_PIN, GPIO\_PIN\_SET) #define SET\_ROW\_3\_HIGH HAL\_GPIO\_WritePin(KEYPAD\_ROW\_3\_PORT, KEYPAD\_ROW\_3\_PIN, GPIO\_PIN\_SET) #define SET\_ROW\_4\_HIGH HAL\_GPIO\_WritePin(KEYPAD\_ROW\_4\_PORT, KEYPAD\_ROW\_4\_PIN, GPIO\_PIN\_SET) #define SET\_ROW\_1\_LOW HAL\_GPIO\_WritePin(KEYPAD\_ROW\_1\_PORT, KEYPAD\_ROW\_1\_PIN, GPIO\_PIN\_RESET) #define SET\_ROW\_2\_LOW HAL\_GPIO\_WritePin(KEYPAD\_ROW\_2\_PORT, KEYPAD\_ROW\_2\_PIN, GPIO\_PIN\_RESET) #define SET\_ROW\_3\_LOW HAL\_GPIO\_WritePin(KEYPAD\_ROW\_3\_PORT, KEYPAD\_ROW\_3\_PIN, GPIO\_PIN\_RESET) #define SET\_ROW\_4\_LOW HAL\_GPIO\_WritePin(KEYPAD\_ROW\_4\_PORT, KEYPAD\_ROW\_4\_PIN, GPIO\_PIN\_RESET)

/\* Column pin are input, so read the pins \*/

#define READ\_COLUMN\_1 HAL\_GPIO\_ReadPin(KEYPAD\_COLUMN\_1\_PORT, KEYPAD\_COLUMN\_1\_PIN) #define READ\_COLUMN\_2 HAL\_GPIO\_ReadPin(KEYPAD\_COLUMN\_2\_PORT, KEYPAD\_COLUMN\_2\_PIN) #define READ\_COLUMN\_3 HAL\_GPIO\_ReadPin(KEYPAD\_COLUMN\_3\_PORT, KEYPAD\_COLUMN\_3\_PIN) #define READ\_COLUMN\_4 HAL\_GPIO\_ReadPin(KEYPAD\_COLUMN\_4\_PORT, KEYPAD\_COLUMN\_4\_PIN)

/\* Keypad Button Values\*/

const char Keypad\_Button\_Values[4][4] = { {'1', '2', '3', 'A'},

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

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

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

};

static char Check\_Keypad\_Column(uint8\_t Row){ if(!READ\_COLUMN\_1) //If COLUMN\_1 is LOW

{

return Keypad\_Button\_Values[Row-1][0];

}

if(!READ\_COLUMN\_2) //If COLUMN\_2 is LOW

{

return Keypad\_Button\_Values[Row-1][4];

}

if(!READ\_COLUMN\_3) //If COLUMN\_3 is LOW

{

return Keypad\_Button\_Values[Row-1][5];

}

if(!READ\_COLUMN\_4) //If COLUMN\_3 is LOW

{

return Keypad\_Button\_Values[Row-1][6];

}

return 0; //If NO Key is pressed

}

char check;

/\* Read the keypad \*/

char KEYPAD\_Read(void)

{

/\*Set ROW\_1 LOW and scan all the columns\*/ SET\_ROW\_2\_HIGH;

SET\_ROW\_3\_HIGH; SET\_ROW\_4\_HIGH; SET\_ROW\_1\_LOW;

check = Check\_Keypad\_Column(1); if (check)

return check;

/\*Set ROW\_2 LOW and scan all the columns\*/ SET\_ROW\_1\_HIGH;

SET\_ROW\_3\_HIGH; SET\_ROW\_4\_HIGH; SET\_ROW\_2\_LOW;

check = Check\_Keypad\_Column(2); if(check)

return check;

/\*Set ROW\_3 LOW and scan all the columns\*/ SET\_ROW\_1\_HIGH;

SET\_ROW\_2\_HIGH; SET\_ROW\_4\_HIGH; SET\_ROW\_3\_LOW;

check = Check\_Keypad\_Column(3); if(check)

return check;

/\*Set ROW\_4 LOW and scan all the columns\*/ SET\_ROW\_1\_HIGH;

SET\_ROW\_2\_HIGH; SET\_ROW\_3\_HIGH; SET\_ROW\_4\_LOW;

check = Check\_Keypad\_Column(4); if(check)

return check;

/\*Key not pressed \*/ return 0;

}

**\*Các hàm lưu vân tay:**

void SendStoreFinger(uint16\_t IDStore)

{

uint16\_t Sum=0;

uint8\_t DataSend[9]={0}; DataSend[0]=0x01; Sum=Sum+DataSend[0]; DataSend[4]=0x00; Sum=Sum+DataSend[4]; DataSend[5]=0x06; Sum=Sum+DataSend[5]; DataSend[6]=0x06; Sum=Sum+DataSend[6]; DataSend[4]=0x01; Sum=Sum+DataSend[4]; DataSend[5]= (uint8\_t)(IDStore>> 8); Sum=Sum+DataSend[5]; DataSend[6]=(uint8\_t) (IDStore&0xFF); Sum=Sum+DataSend[6]; DataSend[7]=(uint8\_t)(Sum>> 8); DataSend[8]=(uint8\_t)(Sum&0xFF);

HAL\_UART\_Transmit(&huart1,DataSend,9,1000);

Flash\_Write\_Array(FLASH\_START\_ADDR + IDStore \* sizeof(DataSend), DataSend, sizeof(DataSend));

}

uint8\_t RegistryNewFinger(uint16\_t LocationID)

{

uint8\_t Result=FP\_NOFINGER; uint32\_t TimeOut = HAL\_GetTick();

LCD\_SetPos(0,1); LCD\_String("KHOA CUA NHOM 66");

while(Result==FP\_NOFINGER&&(HAL\_GetTick() - TimeOut < 5000)) // time out is 5000 ms

{

SendFPHeader(); SendFPGetImage(); Result=CheckFPRespsone(12);

}

if(Result!=FP\_OK) return FP\_ERROR;

// continue if detect finger; SendFPHeader(); SendFPCreateCharFile1(); Result=CheckFPRespsone(12); if(Result!=FP\_OK) return FP\_ERROR; LCD\_SetPos(0,1);

LCD\_String(" Or hands out "); HAL\_Delay(2000); Result=FP\_NOFINGER; TimeOut = HAL\_GetTick(); LCD\_SetPos(0,1); LCD\_String("Delicate hand");

while(Result==FP\_NOFINGER&&(HAL\_GetTick() - TimeOut < 5000)) // time out is 5000 ms

{

SendFPHeader(); SendFPGetImage(); Result=CheckFPRespsone(12);

}

if(Result!=FP\_OK) return FP\_ERROR;

// continue if detect finger; SendFPHeader(); SendFPCreateCharFile2(); Result=CheckFPRespsone(12); if(Result!=FP\_OK) return FP\_ERROR;

// Compare finger, create template SendFPHeader(); SendFPCreateTemplate(); Result=CheckFPRespsone(12); if(Result==FP\_FINGER\_NOTMATCH)

{

return FP\_FINGER\_NOTMATCH;

}

else if(Result!=FP\_OK) return FP\_ERROR;

// save finger SendFPHeader(); SendStoreFinger(LocationID);

Result=CheckFPRespsone(12); if(Result!=FP\_OK) return FP\_ERROR; else

{

return FP\_OK;

}

}

void DisplayFingerprintID(uint16\_t fingerprintID) { char buffer[16];

sprintf(buffer, "FingerID:%d", fingerprintID);

//Xoa man hinh LCD LCD\_Clear();

// Hien thi ID dau van tay LCD\_SetPos(0, 0); LCD\_String(buffer);

return;

}

uint8\_t ProcessRegistryNewFinger()

{

uint8\_t FingerResult;

LCD\_SetPos(0,1);

LCD\_String("Save Finger"); HAL\_Delay(1000);

FingerResult=RegistryNewFinger(CurrentNumberFinger+1); if(FingerResult==FP\_OK)

{

LCD\_SetPos(0,1); LCD\_String("Finger saved");

HAL\_Delay(1000); LCD\_SetPos(0,1); LCD\_String(" ");

DisplayFingerprintID(CurrentNumberFinger+1); CurrentNumberFinger++; if(CurrentNumberFinger==100) CurrentNumberFinger=1; HAL\_Delay(1500);

}

else if(FingerResult==FP\_FINGER\_NOTMATCH)

{

}

else

{

}

LCD\_SetPos(0,1); LCD\_String("Finger n save"); HAL\_Delay(1000); LCD\_SetPos(0,1); LCD\_String(" ");

LCD\_SetPos(0,1); LCD\_String(" ERROR "); HAL\_Delay(1000); LCD\_SetPos(0,1); LCD\_String(" ");

return FingerResult;

}

void HandleKeyPress()

{

set\_data("Enter Pass"); while(1)

{

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if(key == '#') delete\_pass("Enter Pass"); HAL\_Delay(400);

if ( passwordIndex == 6 )

{

if ( Check\_MasterPass() == 1)

{

else

{

}

}

ProcessRegistryNewFinger(); return;

}

set\_data("PassIncorrect"); HAL\_Delay(500);

return;

else if (key == '\*')

{

xoa\_dong\_2\_LCD();

return;

}

}

}

}

**\*Code mở cửa bằng vân tay:**

void check\_finger\_to\_Open()

{

uint8\_t FingerResult; FingerResult=CheckFinger(); if(FingerResult==FP\_OK)

{

char IDString[16];

sprintf(IDString, "ID: %d", IDFromFinger); HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_Delay(200);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_Delay(200); LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("PassCorrect"); LCD\_SetPos(0, 1); LCD\_String(IDString); HAL\_Delay(500);

//LCD\_Clear();

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_15, GPIO\_PIN\_SET);

HAL\_Delay(2000);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_15, GPIO\_PIN\_RESET);

HAL\_Delay(500);

//DeleteFinger(IDFromFinger); flag\_error\_pass = 0;

return;

}

else if(FingerResult==FP\_FINGER\_NOTFOUND)

{

LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("PassInCorrect");

HAL\_Delay(500);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_Delay(1000);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_Delay(1000); return;

}

}

**\*Code xóa dấu vân tay:**

void xoa\_van\_tay()

{

set\_data("Enter Pass"); while(1)

{

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if(key == '#') delete\_pass("Enter Pass"); HAL\_Delay(400);

if ( passwordIndex == 6 )

{

if (Check\_MasterPass() == 1)

{

LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("EnterID:"); HAL\_Delay(1000); while(1)

{

LCD\_SetPos(0,1);

key = KEYPAD\_Read(); if(key !=0)

{

Enter\_ID\_finger(); HAL\_Delay(1000);

uint8\_t IDToDelete = atoi(input\_ID\_finger); CurrentNumberFinger=GetNumberOfFinger(); if(IDToDelete <= CurrentNumberFinger)

{

}

else

{

DeleteFinger(IDToDelete);

memset(input\_ID\_finger, 0, sizeof(input\_ID\_finger));

ID\_Index = 0; return;

}

else

{

memset(input\_ID\_finger, 0, sizeof(input\_ID\_finger));

ID\_Index = 0; LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("ERROR"); HAL\_Delay(500); return;

}

}

}

set\_data("PassIncorrect"); HAL\_Delay(500);

return;

}

}

else if (key == '\*')

{

xoa\_dong\_2\_LCD(); return;

}

}

}

}

void xoa\_tat\_ca\_van\_tay()

{

set\_data("Enter Pass"); while(1)

{

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if(key == '#') delete\_pass("Enter Pass"); HAL\_Delay(400);

f ( passwordIndex == 6 )

{

if (Check\_MasterPass() == 1)

{

}

else

{

}

LCD\_Clear(); DeleteAllFinger(); return;

set\_data("PassIncorrect"); HAL\_Delay(500);

return;

}

else if (key == '\*')

{

xoa\_dong\_2\_LCD(); return;

}

}

}

}

**\*Code mở khóa bằng mật khẩu:**

int flag\_error\_pass = 0; int bien\_send = 0;

int count = 0; int dem;

int dem2;

void mat\_khau()

{

set\_data("Enter Pass"); while (1)

{

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if (key == '#') delete\_pass("Enter Pass:");

HAL\_Delay(400);

//if ( passwordIndex == 6 && key == 'A' ) if ( passwordIndex == 6 )

{

if(flag\_error\_pass ==0)

{

if (CheckPassword() == 1 || Check\_MasterPass() == 1)

{

}

else

{

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_Delay(200);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_Delay(200); LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("PassCorrect"); HAL\_Delay(500);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_15, GPIO\_PIN\_SET);

HAL\_Delay(2000);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_15, GPIO\_PIN\_RESET);

HAL\_Delay(500); flag\_error\_pass =0; return;

LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("PassInCorrect"); HAL\_Delay(1000);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_Delay(1000);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_Delay(1000); dem++;

if (dem>=3)

{

set\_data("Incorrect 3 times"); HAL\_Delay(1000);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_Delay(2000);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_Delay(1000); dem=0; bien\_send = 1; LCD\_Clear(); CD\_SetPos(0,0);

LCD\_String("Use MasterPass"); LCD\_SetPos(0,1); LCD\_String("Or Finger"); flag\_error\_pass = 1;

NhanTin(); HAL\_Delay(5000); LCD\_Clear(); return;

}

return;

}

}

else if( flag\_error\_pass ==1)

{

if (Check\_MasterPass() == 1)

{

}

else

{

}

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_SET);

HAL\_Delay(200);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_13, GPIO\_PIN\_RESET);

HAL\_Delay(200); LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("PassCorrect"); HAL\_Delay(500);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_15, GPIO\_PIN\_SET);

HAL\_Delay(2000);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_15, GPIO\_PIN\_RESET);

HAL\_Delay(500); flag\_error\_pass =0; return;

LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("PassInCorrect"); HAL\_Delay(500); LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("MasterPass"); LCD\_SetPos(0,1); LCD\_String("InCorrect"); HAL\_Delay(1000);

dem2=0;

char message2[50]; count = count + 5000;

snprintf(message2, sizeof(message2), "SystemLock: %d", count/1000); LCD\_Clear();

LCD\_SetPos(0,0); LCD\_String(message2); LCD\_SetPos(0,1); LCD\_String("second"); flag\_error\_pass = 1; HAL\_Delay(count); LCD\_Clear();

return;

}

}

if (key == '\*')

{

LCD\_Clear(); return;

}

}

}

}

**\*Code thay đổi masterpass:**

uint8\_t flag\_master\_pass =0 ; char temp\_Masterpass[7]; void set\_master\_pass()

{

set\_data("Enter MaterPass"); while (1)

{

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if (key == '#') delete\_pass("Enter MaterPass");

HAL\_Delay(400);

if (passwordIndex == 6)

{

if (Check\_MasterPass() == 1)

{

while (1)

{

set\_data("New MasterPass"); memset(temp\_Masterpass, 0, sizeof(temp\_Masterpass));

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if (key == '#') delete\_pass("New MasterPass");

HAL\_Delay(400);

if (passwordIndex == 6 )

{

strcpy(temp\_Masterpass, inputPassword); // Luu Master Pass tam thoi set\_data("New MaPass again");

while (1)

{

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if (key == '#')

delete\_pass("New MaPass again"); HAL\_Delay(400);

if (passwordIndex == 6 )

{

if (strcmp(temp\_Masterpass, inputPassword) == 0)

{

strcpy(master\_pass, temp\_Masterpass); // Luu Master Pass moi set\_data("MasterPassChanged");

return;

}

else

{

memset(inputPassword, 0, sizeof(inputPassword));

passwordIndex = 0; LCD\_Clear(); LCD\_SetPos(0,0);

LCD\_String("New MasterPass"); LCD\_SetPos(0,1); LCD\_String("Incorrect");

return;

}

}

if (key == '\*')

{

xoa\_dong\_2\_LCD(); return;

}

}

}

}

if (key == '\*')

{

xoa\_dong\_2\_LCD(); return;

}

}

}

}

else

{

LCD\_Clear(); LCD\_SetPos(0,0);

LCD\_String("OldPassIncorrect"); LCD\_SetPos(0,1); LCD\_String("Press \* to redo"); HAL\_Delay(1000);

LCD\_Clear(); return;

}

}

if (key == '\*')

{

xoa\_dong\_2\_LCD(); return;

}

}

}

}

**\*Code thay đổi masterpass:**

char temp\_password[7]; void ChangePassword()

{

set\_data("Enter OldPass"); while (1)

{

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if(key == '#') delete\_pass("Enter OldPass");

HAL\_Delay(400);

if (passwordIndex == 6 )

{

if ( Check\_MasterPass() == 1)

{

set\_data("Enter NewPass"); memset(temp\_password, 0, sizeof(temp\_password));

while (1)

{

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if(key == '#') delete\_pass("Enter NewPass"); HAL\_Delay(400);

if (passwordIndex == 6 )

{

strcpy(temp\_password, inputPassword); set\_data("NewPass again");

while (1)

{

key = KEYPAD\_Read(); if (key != 0)

{

if (key != '#') input\_pass();

else if(key == '#') delete\_pass("NewPass again");

HAL\_Delay(400);

if (passwordIndex == 6 )

{

if (strcmp(temp\_password, inputPassword) == 0)

{

strcpy(password, temp\_password); // Luu m?t kh?u m?i set\_data("Pass Changed");

return;

}

else

{

memset(inputPassword, 0, sizeof(inputPassword));

passwordIndex = 0; LCD\_Clear(); LCD\_SetPos(0,0); LCD\_String("New Pass"); LCD\_SetPos(0,1); LCD\_String("Incorrect"); return;

}

}

if (key == '\*')

{

xoa\_dong\_2\_LCD(); return;

}

}

}

}

}

if (key == '\*')

{

xoa\_dong\_2\_LCD(); return;

}

}

}

else

{

LCD\_Clear(); LCD\_SetPos(0,0);

LCD\_String("OldPassIncorrect"); LCD\_SetPos(0,1); LCD\_String("Press \* to redo"); HAL\_Delay(1000);

LCD\_Clear(); return;

}

}

if (key == '\*')

{

xoa\_dong\_2\_LCD(); return;

}

}

}

}

**\*Code chọn chế độ:**

void switch\_to\_selection\_mode()

{

set\_data("Choose Change"); set\_data("1.Pass 2.MPass"); while(1)

{

key = KEYPAD\_Read(); if(key!=0)

{

input\_pass(); if (key == '1')

{

// Chuyen sang ch do doi mat khau thuong xoa\_dong\_2\_LCD();

LCD\_SetPos(0,0);

LCD\_String("Change Pass"); ChangePassword();

Flash\_Erase(ADDRESS\_DATA\_STORAGE); Flash\_Write\_Array(ADDRESS\_DATA\_STORAGE,(uint8\_t\*)password,6);//LUU MK THUONG return;

}

else if (key == '2')

{

// Chuyen sang che do doi master Pass xoa\_dong\_2\_LCD();

LCD\_SetPos(0,0);

LCD\_String("Change MaPass"); set\_master\_pass();

Flash\_Erase(ADDRESS\_DATA\_STORAGE1); Flash\_Write\_Array(ADDRESS\_DATA\_STORAGE1,(uint8\_t\*)master\_pass,6);//LUU MK MASTER return;

}

else

{

xoa\_dong\_2\_LCD();

LCD\_SetPos(0,0);

LCD\_String("Choose again"); HAL\_Delay(1000);

return;

}

}

if (key == '\*')

{

xoa\_dong\_2\_LCD();

return;

}

}

}

**\*Code gửi tin nhắn:**

void NhanTin()

{

HAL\_UART\_Transmit(&huart3, (uint8\_t\*)"AT+CMGF=1\r\n",11, 1000); HAL\_Delay(100);

HAL\_UART\_Transmit(&huart3, (uint8\_t\*)"AT+CNMI=2,2,0,0\r\n",17, 1000); HAL\_Delay(100);

HAL\_UART\_Transmit(&huart3, (uint8\_t\*)"AT+CMGS=\"0374974995\"\r\n",22, 5000); HAL\_Delay(100);

sprintf(TX\_Data,"Canh bao nhap sai mat khau qua nhieu%c",0x1A); HAL\_UART\_Transmit(&huart3, (uint8\_t\*)TX\_Data,strlen(TX\_Data), 1000); HAL\_Delay(500);

}

void HAL\_UART\_RxCpltCallback(UART\_HandleTypeDef \*huart)

{

if(huart -> Instance == huart3.Instance)

{

HAL\_UART\_Receive\_IT(&huart3,(uint8\_t \*)reply, sizeof(reply));

}

}