**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁC NHÓM LẬP TRÌNH HỢP NGỮ**

**KHÓA CỬA THÔNG MINH VỚI ARDUINO**

**GIÁO VIÊN HƯỚNG DẪN**

Mai Cường Thọ

Lớp 57CNTT-1

Nguyễn Trọng Kiên 57130859

Lê Xuân Khánh 57131941

Mai Quang Hòa 57136773

Mã Phi Hùng 57131597

Trần Nguyễn Quốc Bảo 57131557

**Năm học: 2016 – 2017**

# LỜI NÓI ĐẦU

Trong thời đại Công nghệ Thông tin phát triển, thời đại của IoT, chính vì vậy không thể thiếu các vật dụng thông minh cho ngôi nhà của bạn. Đầu tiên phải nhắc tới chiếc khóa cửa để bảo vệ căn nhà của bạn. Sau khi học học phần Lập trình hợp ngữ và nhận thấy nhu cầu thực tế trong việc tạo ra một bộ khóa cửa thông minh nên nhóm đã xây dựng và phát triên hệ thống “Khóa thông mình với Arduino”.

Hệ thống khóa cửa thông minh sẽ giúp việc bảo mật hơn tránh trường hợp bị trộm cắt khóa, quên mang chìa khóa,… Bạn chỉ cần nhớ mật khẩu là có thể mở được khóa, ngoài ra bạn có thể nhập mật khẩu qua Smartphone bằng phần mềm mà nhóm xây dựng.

***Nhóm 1 – 57TH1 – Khoa Công nghệ Thông tin – NTU***

# MỤC LỤC

[LỜI NÓI ĐẦU 2](#_Toc486188799)

[MỤC LỤC 3](#_Toc486188800)

[Chương 1: Giới thiệu chung 5](#_Toc486188801)

[I. Thực trạng 5](#_Toc486188802)

[II. Tổng quan về hệ thống khóa cửa tự động 5](#_Toc486188803)

[Chương 2: Hệ thống khóa cửa tự động 5](#_Toc486188804)

[I. Phần thiết kế 5](#_Toc486188805)

[II. Phần lập trình Arduino 6](#_Toc486188806)

[II.1. LCD 6](#_Toc486188807)

[II.2. Lập trình KeyPad 9](#_Toc486188808)

[II.3. Lập trình Servo 13](#_Toc486188809)

[II.4. Lập trình module Bluetooth 14](#_Toc486188810)

[III. Phần lập trình ứng dụng Android 14](#_Toc486188811)

[III.1. Lập trình ứng dụng bằng AppInventor 14](#_Toc486188812)

[III.2. Chức năng và hướng dẫn sử dụng ứng dụng 16](#_Toc486188813)

[Chương 3: Kết luận và hướng phát triển 17](#_Toc486188814)

[I. Kết luận 17](#_Toc486188815)

[II. Hướng phát triển 17](#_Toc486188816)

# Chương 1: Giới thiệu chung

## I. Thực trạng

Đa số khoá kỹ thuật số đang có bán trên thị trường là do Hàn Quốc sản xuất, chủ yếu là loại khoá tay nắm và có giá khá cao.

Khoá sử dụng phương pháp cài đặt mã số (như khoá số của các loại va li hay cặp số) để khoá hoặc mở và người sử dụng có thể cài đặt số bất kỳ. Hệ thống số của khoá được thiết kế như các phím bấm số của điện thoại nên khá tiện lợi khi sử dụng. Bên cạnh loại chỉ có một chức năng khoá bằng mã số, còn có loại kèm theo chức năng khoá bằng chìa. Chìa của loại này cũng đặc biệt hơn các loại thông thường, nó được làm 4 cạnh, khó làm giả như các loại khoá 2 cạnh.

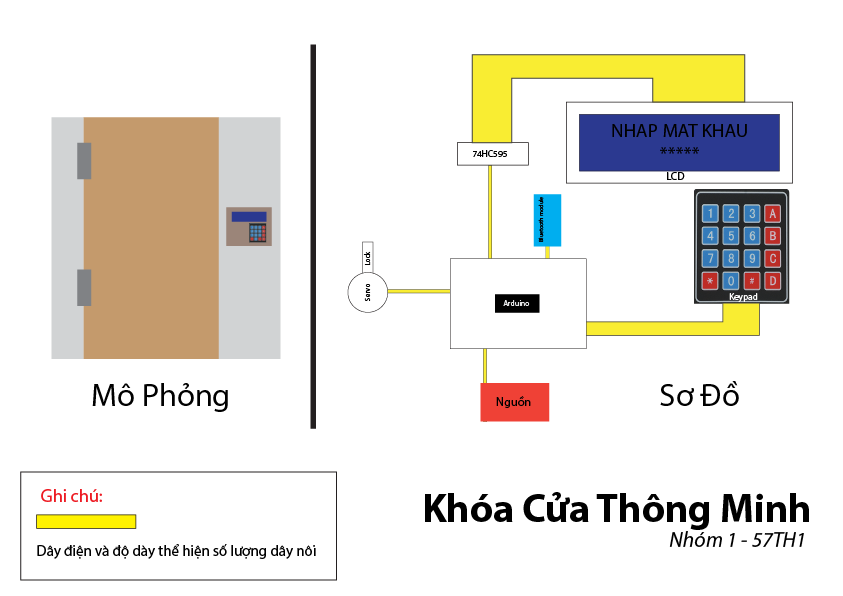
Khoá kỹ thuật số còn có loại mở bằng dấu vân. Loại khoá này có thể đăng ký được 25 hoặc 40 vân tay khác nhau. Như vậy bạn có thể lưu lại rất nhiều vân tay của mọi người trong gia đình vào bộ nhớ của khoá. Khi cho đúng các vân tay có lưu trong bộ nhớ thì cửa sẽ được mở. Phần lớn loại này không sử dụng chìa nữa

## II. Tổng quan về hệ thống khóa cửa tự động

# Chương 2: Hệ thống khóa cửa tự động

## I. Phần thiết kế

Mô tả sơ đồ và mẫu thiết kế sản phẩm như hình bên dưới:



## II. Phần lập trình Arduino

II.1. LCD

1. Hình dáng và kích thước:

Có rất nhiều loại LCD với nhiều hình dáng và kích thước khác nhau, dưới đây là một loại LCD phổ biến



**Hình 2.6:** Hình dáng của LCD

Khi sản xuất LCD, nhà sản xuất đã tích hợp chíp điều khiển (HD44780) bên trong lớp vỏ và chỉ đưa các chân giao tiếp cần thiết. Các chân này được đánh số thứ tự và đặt tên như bên dưới :



b. Chức năng các chân:

1. VSS: tương đương với GND - cực âm
2. VDD: tương đương với VCC - cực dương (5V)
3. Constrast Voltage (Vo): điều khiển độ sáng màn hình
4. Register Select (RS): điều khiển địa chỉ nào sẽ được ghi dữ liệu
5. Read/Write (RW): Bạn sẽ đọc (read mode) hay ghi (write mode) dữ liệu? Nó sẽ phụ thuộc vào bạn gửi giá trị gì vào.
6. Enable pin: Cho phép ghi vào LCD
7. D0 - D7: 8 chân dư liệu, mỗi chân sẽ có giá trị HIGH hoặc LOW nếu bạn đang ở chế độ đọc (read mode) và nó sẽ nhận giá trị HIGH hoặc LOW nếu đang ở chế độ ghi (write mode)
8. Backlight (Backlight Anode (+) và Backlight Cathode (-)): Tắt bật đèn màn hình LCD.

c. Các thanh ghi

Chíp HD44780 có 2 thanh ghi 8 bit quan trọng : Thanh ghi lệnh IR (Instructor Register) và thanh ghi dữ liệu DR (Data Register)

- Thanh ghi IR : Để điều khiển LCD, người dùng phải “ra lệnh” thông qua tám đường bus DB0-DB7. Mỗi lệnh được nhà sản xuất LCD đánh địa chỉ rõ ràng. Người dùng chỉ việc cung cấp địa chỉ lệnh bằng cách nạp vào thanh ghi IR. Nghĩa là, khi ta nạp vào thanh ghi IR một chuỗi 8 bit, chíp HD44780 sẽ tra bảng mã lệnh tại địa chỉ mà IR cung cấp và thực hiện lệnh đó.

VD : Lệnh “hiển thị màn hình” có địa chỉ lệnh là 00001100 (DB7…DB0)

Lệnh “hiển thị màn hình và con trỏ” có mã lệnh là 00001110

- Thanh ghi DR : Thanh ghi DR dùng để chứa dữ liệu 8 bit để ghi vào vùng RAM DDRAM hoặc CGRAM (ở chế độ ghi) hoặc dùng để chứa dữ liệu từ 2 vùng RAM này gởi ra cho MPU (ở chế độ đọc). Nghĩa là, khi MPU ghi thông tin vào DR, mạch nội bên trong chíp sẽ tự động ghi thông tin này vào DDRAM hoặc CGRAM. Hoặc khi thông tin về địa chỉ được ghi vào IR, dữ liệu ở địa chỉ này trong vùng RAM nội của HD44780 sẽ được chuyển ra DR để truyền cho MPU. Bằng cách điều khiển chân RS và R/W chúng ta có thể chuyển qua lại giữ 2 thanh ghi này khi giao tiếp với MPU. Bảng sau đây tóm tắt lại các thiết lập đối với hai chân RS và R/W theo mục đích giao tiếp.

|  |  |  |
| --- | --- | --- |
| RS | R/W | Khi cần |
| 0 | 0 | Ghi vào thanh ghi RS để ra lệnh cho thanh ghi |
| 0 | 1 | Đọc cờ bận ở DB7 và giá trị của bộ đếm địa chỉ ở DB0-DB6 |
| 1 | 0 | Ghi vào thanh ghi DR |
| 1 | 1 | Đọc dữ liệu từ DR |

d. Phần lập trình thiết lập màn hình

void RCK\_PULSE() { RCK\_1; delay\_us(4); RCK\_0; }

void SCK\_PULSE() { SCK\_1; delay\_us(4); SCK\_0; }

void Shift\_595(char data) {

for(int i=0; i<8; i++) {

if(data & (0x80 >> i)) SER\_1; // MSBFIRST

else SER\_0;

SCK\_PULSE();

}

RCK\_PULSE();

}

void LCD\_DATA(char data) {

RCK\_0;

Shift\_595((data & 0xF0) | RS1\_EN1 | BackLight);

Shift\_595((data & 0xF0) | RS1\_EN0 | BackLight);

delay\_us(4);

Shift\_595(((data<<4) & 0xF0) | RS1\_EN1 | BackLight);

Shift\_595(((data<<4) & 0xF0) | RS1\_EN0 | BackLight);

delay\_us(50);

}

void LCD\_CMD(char cmd) {

RCK\_0;

Shift\_595((cmd & 0xF0) | RS0\_EN1 | BackLight);

Shift\_595((cmd & 0xF0) | RS0\_EN0 | BackLight);

delay\_us(4);

Shift\_595(((cmd<<4) & 0xF0) | RS0\_EN1 | BackLight);

Shift\_595(((cmd<<4) & 0xF0) | RS0\_EN0 | BackLight);

delay\_us(50);

}

void LCD\_CMD\_4bit(char cmd) {

RCK\_0;

Shift\_595((cmd & 0xF0) | RS0\_EN1 | BackLight);

Shift\_595((cmd & 0xF0) | RS0\_EN0 | BackLight);

delay\_us(50);

}

void LCD\_INIT(void) {

delay\_ms(100);

Shift\_595(0x00);

LCD\_CMD\_4bit(0x30); delay\_ms(5);

LCD\_CMD\_4bit(0x30); delay\_us(100);

LCD\_CMD\_4bit(0x30); delay\_us(100);

LCD\_CMD\_4bit(0x20); delay\_us(100);

LCD\_CMD(0x28); // 4 bits, 2 line, 5x8 font

LCD\_CMD(0x08); // display off, cursor off, blink off

LCD\_CMD(0x01); // clear display

delay\_ms(3);

LCD\_CMD(0x06); // cursor movint direction

LCD\_CMD(0x0C); // display on, cursor off, blink off

}

void LCD\_XY(char x, char y) {

if (y==0) LCD\_CMD(0x80 + x);

else if(y==1) LCD\_CMD(0xC0 + x);

else if(y==2) LCD\_CMD(0x94 + x);

else if(y==3) LCD\_CMD(0xD4 + x);

}

void LCD\_CLEAR(void) { LCD\_CMD(0x01); delay\_ms(2);}

void LCD\_PUTS(char \*str) { while(\*str) LCD\_DATA(\*str++); }

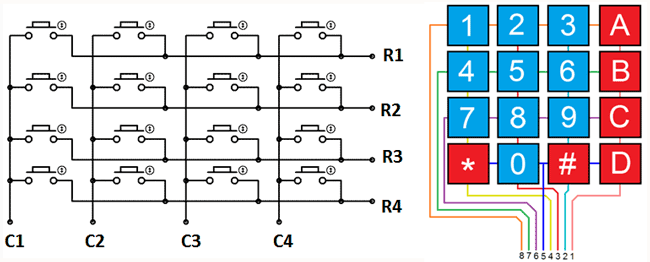
II.2. Lập trình KeyPad

a. Hình dáng và kích thước:

Keypad là một dạng bàn phím gồm nhiều nút bấm để thực hiện chức năng theo lập trình của người sử dụng.

Loại Keypab phổi biến là 4x4 với các phím số từ 0 – 9 và các phím chức năng A B C D \* # như hình bên

b. Sơ đồ và nguyên lý:

[](http://k2.arduino.vn/img/2016/06/29/0/2561_882450-1467179610-0-4x4-matrix-keypad.gif)

Trên đây là hình ảnh sơ đồ nguyên lý của keypad 4x4. Tuy có đến 16 nút nhấn, nghĩa là nếu làm một cách thông thường (dùng chân digital) thì chúng ta phải cần đến 16 chân Arduino để đọc. Nhưng với bàn phím này, chúng ta chỉ cần dùng 8 chân (4 chân hàng ngang (row), và 4 chân cột dọc (column)).

Để kiểm tra một nút có được nhấn hay không? Họ sẽ sử dụng phương pháp quét được mô tả bằng đoạn mã giả như sau:

Với mỗi hàng (R1 đến R4), Chọn ra hàng Ri

Cấp cực âm (0v) cho hàng Ri

Nếu điện áp ở chân Cj bất kì là dương (INPUT PULLUP) => chưa nhấn

Nếu điện áp ở chân Cj bất kì là âm (INPUT PULLUP) => đang nhấn

c. Phần lập trình keypad

const byte rows = 4; //số hàng

const byte columns = 4; //số cột

int holdDelay = 700; //Thời gian trễ để xem là nhấn 1 nút nhằm tránh nhiễu

int n = 3; //

int state = 0; //nếu state =0 ko nhấn,state =1 nhấn thời gian nhỏ , state = 2 nhấn giữ lâu

char key = 0;

char NhapMatKhau[32], LuuMatKhau[32] = {'1','2','3'};

int i;

bool Status = 0;//Status điều khiển trạng thái trong hàm BanPhim

// và ngăn không cho nhảy liên tục làm thay đổi i

//Định nghĩa các giá trị trả về

char keys[rows][columns] =

{

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

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

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

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

};

byte rowPins[rows] = {5, 6, 7, 8}; //Cách nối chân với Arduino

byte columnPins[columns] = {9, 10, 11, 12};

//cài đặt thư viện keypad

Keypad keypad = Keypad(makeKeymap(keys), rowPins, columnPins, rows, columns);

void BanPhim() // Hàm bàn phím

{

char temp = keypad.getKey();

if ((int)keypad.getState() == PRESSED)

{

if (temp != 0)

{

key = temp;

Status = 1;

}

}

if ((int)keypad.getState() == HOLD)

{

state++;

state = constrain(state, 1, n-1);

delay(holdDelay);

}

if ((int)keypad.getState() == RELEASED)

{

key += state;

state = 0;

//Xuất lên Máy tính để xem kết quả

Serial.print(key);

Serial.print(" ");

}

delay(100);

}

//--------------------------------------------------------------------------------//

void NhapMK() // Hàm nhập mật khẩu

{

i = 0;

key = 0;

Status = 0;

do

{

if(key != '#')

{

//Serial.print("Status: ");

//Serial.println(Status);

if(Status == 1)

{

NhapMatKhau[i] = key;

LCD\_XY(i, 1); //Xác định đóng và vị trí bắt đầu in// số đầu là vị trí in, số sau là dòng in

LCD\_PUTS((char \*)"\*"); //câu lệnh in

i++;

}

}

Status = 0;

BanPhim();

// Serial.println(i);

}while(key != '#');

key = 0;

}

//---------------------------------------------------------------------------------//

void DoiMK() // Hàm đổi mật khẩu

{

char XacMinhMK[32];

Serial.println("DOI MAT KHAU");

Serial.println("NHAP MAT KHAU CU:");

LCD\_XY(1,0);

LCD\_PUTS((char \*) "DOI MAT KHAU:");

delay(2000);

LCD\_CLEAR();

LCD\_XY(1,0);

LCD\_PUTS((char \*) "MAT KHAU CU:");

NhapMK();

LCD\_CLEAR();

if(KT(LuuMatKhau))

{

Serial.println("MAT KHAU DUNG");

Serial.print("\nNhap mat khau moi: ");

LCD\_XY(1,0);

LCD\_PUTS((char \*) "MAT KHAU DUNG");

delay(2000);

LCD\_CLEAR();

LCD\_XY(1,0);

LCD\_PUTS((char \*) "MAT KHAU MOI:");

NhapMK();

LCD\_CLEAR();

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

XacMinhMK[j] = NhapMatKhau[j];

Serial.print("\nNhap xac minh mat khau moi: ");

LCD\_XY(1,0);

LCD\_PUTS((char \*) "XAC MINH MK:");

NhapMK();

LCD\_CLEAR();

if(KT(XacMinhMK))

{

Serial.println("MAT KHAU XAC MINH CHINH XAC");

LCD\_XY(1,0);

LCD\_PUTS((char \*) "MAT KHAU XAC");

LCD\_XY(0,1);

LCD\_PUTS((char \*) "MINH CHINH XAC");

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

LuuMatKhau[j] = XacMinhMK[j];

}

else

{

Serial.println("MAT KHAU XAC MINH SAI");

LCD\_XY(1,0);

LCD\_PUTS((char \*) "MAT KHAU XAC");

LCD\_XY(0,1);

LCD\_PUTS((char \*) "MINH SAI");

}

}

else

Serial.println("SAI MAT KHAU");

delay(1000);

LCD\_CLEAR();

key = 0;

}

//---------------------------------------------------------------------------------

bool KT(char Luu[]) \\ hàm kiểm tra mật khẩu

{

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

if(NhapMatKhau[j] != Luu[j])

return 0;

return 1;

}

II.3. Lập trình Servo

a. Giới thiệu chung về Servo

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ 0o – 180o. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô mình), có loại thì sở hữu một momen lực bá đạo (vài chục Newton/m), hoặc có loại thì khỏe và nhông sắc chắc chắn,...

Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bầt kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác. Các động cơ servo điều khiển bằng liên lạc vô tuyến được gọi là động cơ servo RC (radio-controlled). Trong thực tế, bản thân động cơ servo không phải được điều khiển bằng vô tuyến, nó chỉ nối với máy thu vô tuyến trên máy bay hay xe hơi. Động cơ servo nhận tín hiệu từ máy thu này.



b. Lập trình Servo:

void DKservo() // hàm điều khiển servo

{

//Serial.println(servoPos);

for(servoPos = 0; servoPos <= 180; servoPos++)

{

servo.write(servoPos);

// Serial.println(servoPos);

delay(100);

}

delay(5000);

for(servoPos; servoPos > 0; servoPos--)

{

servo.write(servoPos);

// Serial.println(servoPos);

delay(100);

}

}

II.4. Lập trình module Bluetooth

void Bluetooth()

{

if(Serial.available() > 0)

{

state1 = Serial.read();

}

else

state1 = 0;

Serial.println(255-state1);

if ((255-state1)==1)

{

Serial.println("MAT KHAU DUNG");

LCD\_XY(1,0);

LCD\_PUTS((char \*) "DUNG MAT KHAU");

delay(1000);

DKservo();

LCD\_CLEAR();

}

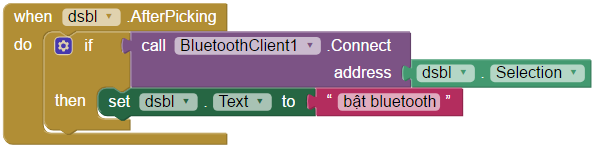
delay(1000);

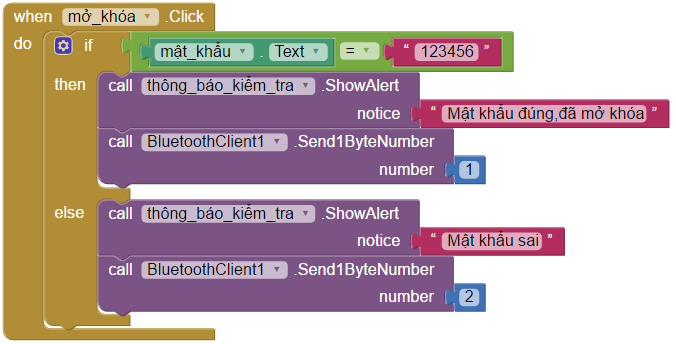
}

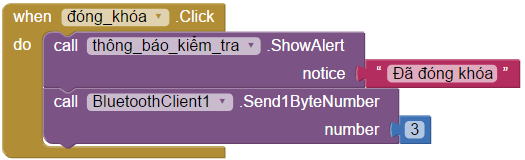
## III. Phần lập trình ứng dụng Android

III.1. Lập trình ứng dụng bằng AppInventor









III.2. Chức năng và hướng dẫn sử dụng ứng dụng

a. Lưu đồ giải thuật:

Khởi Tạo Biến, Mảng, LCD

QUÉT PHÍM

NHẬP PHÍM

KIỂM TRA SỐ LẦN >=3

MỞ CỬA - ĐỔI PASS

b. Hoạt động của sản phẩm

Khi muốn mở cửa thì phải gõ đúng mật khẩu với độ dài 4 ký tự. Nếu gõ đúng thì động cơ sẽ quay thuận, và cửa sẽ mở. Và gõ sai thì màn hình LCD hiển thị chữ nhập sai mật khẩu.

Khi động cơ quay hết hành trình, công tắc hành trình mở xong đóng, khi đó LCD thông báo đã mở xong.

Trong khi mở xong, nếu nhấn khóa thì động cơ sẽ quay và cửa khóa lại, động cơ quay thì nút công tắc hành trình khóa xong sẽ mở, và nếu công tắc hành trình mở xong đóng lại thì động cơ ngừng quay, LCD báo mở xong.

Muốn đổi mật khẩu ta nhấn nút đổi mật khẩu trên bàn phím, khi này LCD báo nhập mật khẩu cũ, và ta phải nhập đúng mật khẩu cũ thì mới đổi được mật khẩu, nếu nhập đúng mật khẩu thì LCD thông báo nhập mật khẩu mới. nhập xong thì nhấn OK, vậy là mật khẩu đã được đổi.

# Chương 3: Kết luận và hướng phát triển

## I. Kết luận

Quá trình xây dựng và phát triển hệ thống khóa tự động mà nhóm thực hiện mới chỉ thể hiện được chức năng cơ bản của khóa thông minh. Một số chức năng còn hạn chế và chưa hoàn hảo. Tuy nhiên nhóm đã cùng nhau xây dựng và phát triển hệ thống khóa hoàn chỉnh.

## II. Hướng phát triển

* Hệ thống khóa sẽ tích hợp thêm chức năng đổi mật khẩu qua thiết bị được kết nối bluetooth, mà linh hoạt hơn trong việc kết nối cũng như mở khóa
* Thích hợp hệ thống nhận dạng khuôn mặt. nhận dạng vân tay để mở khóa
* Hoàn thiện giao diện và ứng dụng đi kèm
* Thêm chức năng cảnh báo khi có kẻ gian cố tình xâm nhập qua báo động và qua điện thoại