PHENIKAA UNIVERSITY

FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING

MICROPROCESSOR AND MICROCONTROLLER ENGINEERING FINAL REPORT

Topic: Secure Entryway Management System - SEMS

Students: Dương Doãn Tùng

Nguyễn Trọng Huy Hoàng

Class: K15-AI&RB

Course: Kỹ thuật vi xử lý và vi điều khiển-1-2-22(AIR05)

Instructor: Huỳnh Bá Phúc

HA NOI, 4/2023

TASK & DIVISION OF WORK

Họ và tên	MSSV	Công việc thực hiện	Chữ ký
Dương Doãn Tùng	21010294	Thiết kế bảng mạch và Raspberry Pi	
Nguyễn Trọng Huy Hoàng	21011002	Thiết kế và lắp đặt mô hình và lập trình Arduino.	

The source code can be found at github.com/dtungpka/SEMS

Table of Contents

TASK & DIVISION OF WORK 2
I. ABSTRACT 1
II. INTRODUCTION 1
III. PROBLEM DEFINITION AND PROPOSED SOLUTIONS 2
IV. THEORETICAL AND PRACTICAL EXPLANATIONS 2
V. DESIGN AND IMPLEMENTATION 5
1. Hardware design 5
2. Program structure 5
2.1. Arduino program 5
2.2. Raspberry Pi Program 6
3. Arduino code explaination 6
VI. RESULTS 10
VII. CONCLUSION 11

I. ABSTRACT

The topic of our project is to create a model door that can be controlled by an Arduino and integrated with facial recognition feature using Raspberry Pi. This model door is designed to be opened or closed by using a push button, and users can enter a password to open the door using a control keypad. When the password is entered correctly, the door will open and the LCD screen outside will display the entered password. Additionally, to ensure the safety and security of the door, we have added the facial recognition feature using Raspberry Pi. This will help users to open the door by recognizing their face instead of entering a password every time. To achieve this, we use the IR camera on Raspberry Pi to recognize and transmit information to the Arduino via UART communication protocol.

II. INTRODUCTION

The project consists of two main functions: unlocking the door by password and unlocking the door by face recognition. The user can enter a password using a 4x4 matrix keyboard, and the password will be displayed on a 16x2 LCD screen with asterisks. If the password is correct, the door will open.

The face recognition function uses an IR camera on Raspberry Pi to detect and identify the user's face. If the face matches one of the registered faces in the database, the Raspberry Pi will send a signal to Arduino via UART communication protocol to open the door. The name of the recognized user will also be shown on the LCD screen.

The project was completed by following these steps:

1) designing the circuit diagram and layout of the model door;
2) programming the Arduino and Raspberry Pi using C++ and Python languages; 3) testing and debugging the code and hardware;
3) evaluating the performance and accuracy of the system;
4) writing the project report and preparing the presentation.

The project faced some challenges, such as:

1) finding suitable components and materials for the model door;

2) ensuring reliable communication between Arduino and Raspberry Pi;
3) improving the speed and accuracy of face recognition;
4) dealing with environmental factors such as lighting.

The project provides a practical example of how Arduino and Raspberry Pi can be used to create a smart home device that can improve the security and convenience of users. The project also demonstrates the skills and knowledge of electronics, programming, and engineering that we have learned during our course.

III. PROBLEM DEFINITION AND PROPOSED SOLUTIONS

The problem that our project aims to address is how to design and implement a model door that can be controlled by Arduino and integrated with face recognition feature by Raspberry Pi. This problem is relevant for applications that require high security and convenience for users, such as smart homes, offices, or hotels. The problem is also challenging because it involves combining different hardware and software components, such as sensors, cameras, LCD screens, keyboards, Arduino boards, Raspberry Pi boards, and communication protocols. The problem has not been fully solved by existing solutions, which either rely on only one method of unlocking (such as password or face recognition) or are too complex and expensive to implement. Therefore, our project aims to provide a simple, low-cost, and effective solution that can demonstrate how to use Arduino and Raspberry Pi to control a model door with both password and face recognition features.

IV. THEORETICAL AND PRACTICAL EXPLANATIONS

The system consists of two main components: an Arduino board and a Raspberry Pi board. The Arduino board is responsible for controlling the door lock mechanism, displaying text on the LCD screen, reading input from the keypad and the button, and communicating with the Raspberry Pi board via UART protocol. The Raspberry Pi board is responsible for performing face recognition, checking password validity, and handling the admin panel. The Arduino UNO plays an essential part in this project, with keyboard reading, display LCD text and control door using Timer1.

The Arduino board is connected to a 4x4 matrix keypad, a 16x2 LCD screen, a button, 2 LED and a stepper motor. The keypad allows the user to enter a password, which will be displayed on the LCD screen with asterisks. The stepper motor acts as a door hinge. The button can be used to open the door from inside.

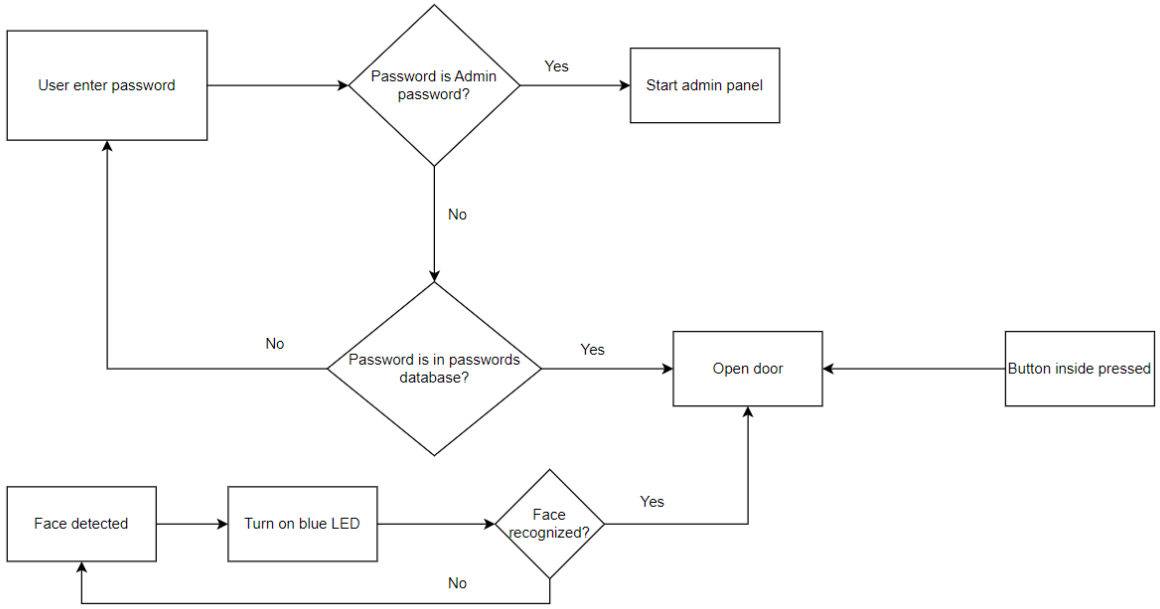

Diagram 1. 3 Ways to open the door

The Raspberry Pi board is connected to an IR camera that can capture images of the user's face. The Raspberry Pi board uses OpenCV library to detect and identify faces using a pre-trained classifier. The classifier compares the captured face with the registered faces in the database and returns the name of the matched user. The Raspberry Pi board also has an admin panel that allows the user to register new faces, delete existing faces, or change passwords, which can be accessed from the keypad.

The communication between the Arduino board and the Raspberry Pi board is done using UART protocol. UART stands for Universal Asynchronous Receiver/Transmitter, which is a serial communication method that uses two wires: TX (transmit) and RX (receive). The Arduino board sends data to the Raspberry Pi board using its TX pin, and receives data from the Raspberry Pi board using its RX pin. The data is sent and received in bytes, which are encoded using ASCII characters.

The system works as follows:

- When the system is powered on, it enters password mode by default. The LCD screen displays "*Enter Password:*" and waits for user input.

- The user can enter a password using the keypad. The password will be displayed on the LCD screen with asterisks, which can be changed in the admin panel.

- The Arduino board sends the password to the Raspberry Pi board using UART protocol.

- The Raspberry Pi board checks if the password is valid by comparing it with the stored passwords in the database.

- If the password is valid, the Raspberry Pi board sends open door command to the Arduino board using UART protocol.

- During the opening and closing phases of the door, the red LED will turn on to indicate its status. Once the door has completed its action, the LED will automatically turn off.

- The Arduino board receives the command and activates the stepper motor, which turns the door and open it. The LCD screen displays "*Access granted*" for 20 seconds, then returns to "*Enter Password:*".

-The user can use the face recognition mode by simply looking at the IR camera. When a face is detected in frame, the blue LED lights up to indicate that a face has been detected.

- The Raspberry Pi board uses OpenCV library to detect and identify faces using a pre-trained classifier.

- If a face is detected and matched with one of the registered faces in the database, the Raspberry Pi board automatically send open door command, followed by the name of the recognized user to the Arduino board using UART protocol.

- The Arduino board then unlocks the door. The LCD screen displays "*Access granted: [name]* " for 20 seconds, then returns to "*Enter Password:*".

- The user can access the admin panel by entering a special password (default is 0301011). This password can be changed by the user in the admin panel.

V. DESIGN AND IMPLEMENTATION

1. Hardware design

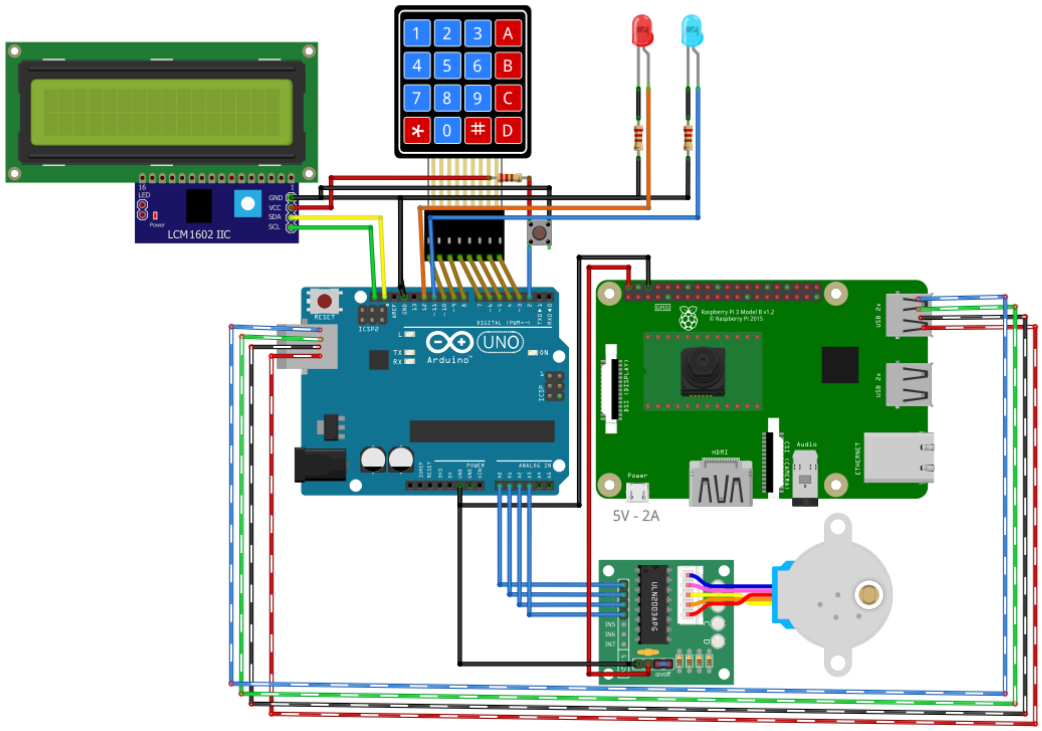

Figure 1. Circuit design

As we mentioned earlier, our circuit consists of 1 Arduino UNO, 1 LCD 16x2 with I2C, 1 4x4 Keypad, 1 28byj-48 step motor with ULN2003 module, 2 LED, 1 Raspberry Pi 4 1GB. Then we designed the circuit in the Figure 1 above.

2. Program structure

2.1. Arduino program

The program uses the LiquidCrystal_I2C library to display information on an I2C-enabled LCD screen. The user interacts with the system through a 4x4 keypad, which is connected to the Arduino.

The stepper motor has three modes of operation: WAVE, FULL, and HALF. In the WAVE mode, the stepper motor operates in a wave sequence; in the FULL mode, it operates in a full-step sequence; and in the HALF mode, it operates in a half-step sequence.

The system also has a number of global variables that are used throughout the program. These include the row and column pins used by the keypad, the speed of the stepper motor, the time it takes to open the door, the height of the door, and the current step of the door.

Our code makes use of interrupts to detect button presses on the keypad. The INT0_vect interrupt is triggered whenever a button is pressed. If the system is in calibration mode, it counts the number of steps required to open the door. Otherwise, it checks the current state of the door and either opens or closes it accordingly.

The system can also be controlled by the RPI, which sends commands to the Arduino through serial communication. The commands include sending the password entered by the user to the RPI, displaying a message on the LCD screen, granting access to open the door, setting the speed of the stepper motor, setting the time it takes to open the door, setting the number of steps required to open the door, checking the data entered by the user, and calibrating the system.

The system has a number of functions, including openDoor() and closeDoor(), which are called when the system needs to open or close the door. These functions operate the stepper motor in the appropriate sequence to move the door to the desired position. The system also has a timer interrupt that is used to control the speed of the stepper motor.

2.2. Raspberry Pi Program

Our raspberry pi code written in Python make use of face_recognition module to detect and recognize face from the camera. We also using threading to handle UART data transmission from arduino back and forth.

3. Arduino code explaination

In this report, we focus on explaining the arduino code more than the raspberry pi code[1].

We first define a few global variables and enums for our program. Some of them are:

```
enum Command {
	SEND_KEY = 'K', // send pressed key to RPI
	DISPLAY_MESSAGE = 'D', // display message on LCD
	GRANT_ACCESS = 'G', // grant access to open the door
	CLOSE_DOOR = 'L', // close the door
	FACE_DETECTED = 'F', // face detected
	END = '_', //end current command
	READY = 'R', // ready to receive command
	//setting variable
	SET_ACCESS_TIME = 'A',
	SET_SPEED = 'E', // set speed
	SET_OPENTIME = 'T', // set door open time
	CALIBRATING = 'C', // Close the door till button is pressed, then start
counting step take to open the door (button is pressed again)
	SET_CALIBRATING_VALUE = 'V', // set the value of door steps
	GET_CALIBRATING_VALUE = 'S', // get the value of door steps
	CHECK_DATA = 'X', // check if the data is correct
```

[1] The raspberry pi code can be found at the source code repository

```
};
enum DoorState {
	DOOR_OPEN = 'O',
	DOOR_CLOSE = 'C',
	DOOR_OPENING = 'I',
	DOOR_CLOSING = 'L',
};
```

The Command enum define the data to send between arduino and raspberry pi, and DoorState define the state of the door.

```
ISR(INT0_vect) {
	//TBA -> open door
	if (millis() - last_btn_time < DELAY_BETWEEN_BTN) return;
	//if calibrating, move to next step
	if (calibrating)
	{
		calibrate_count++;
	}
	else
	{
		if (doorState == DOOR_CLOSE || doorState == DOOR_CLOSING)
		{
			openDoor();
		}
		else if (doorState == DOOR_OPEN || doorState == DOOR_OPENING)
		{
			closeDoor();
		}
	}

	last_btn_time = millis();
}
```

The INT0_vect interrupt is triggered whenever a button is pressed. This button is from inside house so we can open the door from inside.

```
ISR(TIMER1_OVF_vect)
{
	//Serial.println(preloader);
	TCNT1 = preloader;
	if (doorStep == 0 && !(calibrating && calibrate_count > 0)) return;
	digitalWrite(DOOR_LED, HIGH);
	PORTC = (PORTC & 0b110000) | (STAGES[mode][stage]);
	if (doorState == DOOR_OPENING)
	{
		if (stage == 0)
			stage = 7;
		else
			stage--;
	}
	else if (doorState == DOOR_CLOSING)
	{
		if (stage == 7)
			stage = 0;
		else
			stage++;
	}
	if (calibrating) doorStep++;
	else doorStep--;
```

```
}
```

The timer1 overflow interrupts is used to control the step motor,the direction is based on the door state.

```
void openDoor() {
	if (doorState == DOOR_OPENING || doorState == DOOR_OPEN) return;
	doorState = DOOR_OPENING;
	if (doorStep == 0)
		doorStep = doorHeight;
	else doorStep = doorHeight - doorStep;
}

void closeDoor() {
	if (doorState == DOOR_CLOSING || doorState == DOOR_CLOSE) return;
	doorState = DOOR_CLOSING;
	if (doorStep == 0)
		doorStep = doorHeight;
	else doorStep = doorHeight - doorStep;
}
```

These 2 function control the door, to open and close it when called.

```
void getCommand() {
cont:
	//Serial.println((char)READY);
	//get serial command from RPI, should be in format: <command><value>
	char command;
	String value = "";
	//read the first char and store it in command
	String tmp = Serial.readStringUntil('\n');
	//Seperate the first char to command
	command = tmp[0];
	//Seperate the rest of the string to value
	value = tmp.substring(1);
	if (command == 0) goto cont;
	//Print back the received command and value
	if ((int)command != END) {
		Serial.print((char)CHECK_DATA);
		Serial.println(value);
	}

#ifdef SERIAL_DEBUG
	Serial.print("Command: ");
	Serial.println(command);
	Serial.print("Value: ");
	Serial.println(value);
#endif
	switch ((int)command)
	{
	case SEND_KEY:
		break;
	case DISPLAY_MESSAGE:
		lcd.clear();
		lcd.setCursor(0, 0);
		lcd.print(value.substring(0, 16));
		lcd.setCursor(0, 1);
		lcd.print(value.substring(16));
		break;
	case GRANT_ACCESS:
		openDoor();
```

```
                break;
        case CLOSE_DOOR:
                closeDoor();
                break;
        case FACE_DETECTED:
                if (value.charAt(0) == 'D')
                        digitalWrite(FACE_LED, HIGH);
                else if (value.charAt(0) == 'U')
                        digitalWrite(FACE_LED, LOW);
                break;
        case SET_ACCESS_TIME:
                doorOpenTime = value.toInt();
                break;
        case SET_SPEED:
                if (value.toInt() > 0)
                        preloader = value.toInt();
                break;
        case SET_OPENTIME:
                if (value.toInt() > 0)
                        doorOpenTime = value.toInt();
                break;
        case CALIBRATING:
                calibrate();
                break;
        case END:
                return;
        case SET_CALIBRATING_VALUE:
                doorHeight = value.toInt();
        case GET_CALIBRATING_VALUE:
                Serial.print("V");
                //doorHeight to string
                Serial.println(doorHeight);
        default:
                break;
        }
        goto cont;
}
```

The getcommand function to handle the UART communication between raspberry pi.

Above are some remarkable fucntion and code that are important. For more information please see the source code.

VI. RESULTS

This is the house model that we created

Figure 2. House model

Figure 3. Enter password screen

Figure 4. Admin panel screen

VII. CONCLUSION

In this project report, we summarizes the main functions, steps, challenges, and outcomes of the project. The project aimed to create a smart door lock system that can be unlocked by password or face recognition using Arduino and Raspberry Pi. The project was successfully completed by following a systematic process of design, programming, testing, evaluation, and documentation. The project also faced some difficulties in finding suitable components, ensuring reliable communication, improving face recognition performance, and dealing with environmental factors. The project showed how Arduino and Raspberry Pi can be integrated to create a useful device for smart home applications. The project also reflected the learning outcomes of the course in terms of electronics, programming, and engineering skills.