Department of Electrical and Computer Engineering
North South University, Dhaka-1229, Bangladesh

Securitas: Security lock system using PIC microcontroller

Sabbir Mollah (1712490042),
Abdullah Al Rafi (1731371642)
Rifat Ahammed (1721486642)

Submitted to: Mr. Rishad Arefin (RsF)
Course: CSE331.2
Fall 2020

Table of Contents

Introduction

Securitas is a multi user security lock system which is controlled by a PIC16 microcontroller. In this system, specific users can open the door by using a unique pin code. There is a servo motor attached to the door, which will let the door open by receiving a correct password. There is also a security alarm for the system, which will buzz if anyone tries to access the system by giving a wrong password. As this is a multi user security system, an admin can monitor the whole system and can add or remove any user from the system.

Objective

The objective of Securitas is to secure a lock system that supports multi users. It is very useful in the cases where multiple people use the same shared space. The primary applications of Securitas are top priority vaults, places with sensitive information and the like. Securitas solves the security issues in the cheapest way, at the same time, it enriches our skills in Microcontroller programming.

Features

(C - Complete, I - Incomplete, P - Partially complete)

[C] Hex Keyboard
[C] Door opening closing
[C] Output to the display
[P] Security Alarm
[I] Add user
[P] ADMIN Control
[I] Add/Remove users
[I] Change Pin

Applications

- High priority vaults in bank
- Sensitive places which have limited access like servers
- Door system in houses
- Multi user access control for any restricted area

Working Procedures

The entire project was created by using the simulation software Proteus 8 and MikroC as the compiler IDE. So, at first , we took PIC16F877A as our microcontroller and then we iteratively added the different components.

We connected a display with the processor so that users can see the current view they are on within the system.

After that we attached a “Keypad SmallCalc” with the D register to achieve hexadecimal input ability. Here, we manually assigned:

- **÷** as A
- **X** as B
- **-** as C
- **+** as D
- **On/C** as F
- **=** as E

To sync with a real door , we used a servo motor ,which will allow the door to open or close.
And for the security Alarm ,we used a buzzer. For more readable design , we used labels.

We stored our whole code in the 14KB memory space of PIC 16F877A.

When the system is being used for the very first time , 1234 is the default pin. There’s also an additional 1 in front of the pin to validate the user. So, if the user exists the first bit will be 1 else it will be 0.

The main display view contains following options:

- **Open:** After entering the correct pin, the servo will be initiated and the door will be opened by rotating the servo at 90 degree. After certain delay it will come back to its original position

- **Change Pin:** This will let the user to change the password

- **Add User:** This section is only for the Admin

- **Remove User:** Admin only section. Admin can remove users upon entering the pin.

- **Information:** This section will display about the developers of the project.

EEPROM Memory Structure

The pincodes of the individual user will be stored in the internal EEPROM. Each user will take 5 bytes in the memory. The first byte will indicate whether this user is active or not. The following four bytes will store the pincode of that user.

Address	Storage
0	Is user 1 active? (1/0)
1	User 1 pin first digit
2	User 1 pin 2nd digit
3	User 1 pin 3rd digit
4	User 1 pin 4th digit
5	Is user 2 active? (1/0)
6	User 2 pin first digit
7	User 2 pin 2nd digit
8	User 2 pin 3rd digit
9	User 2 pin 4th digit
...	...

Project Management

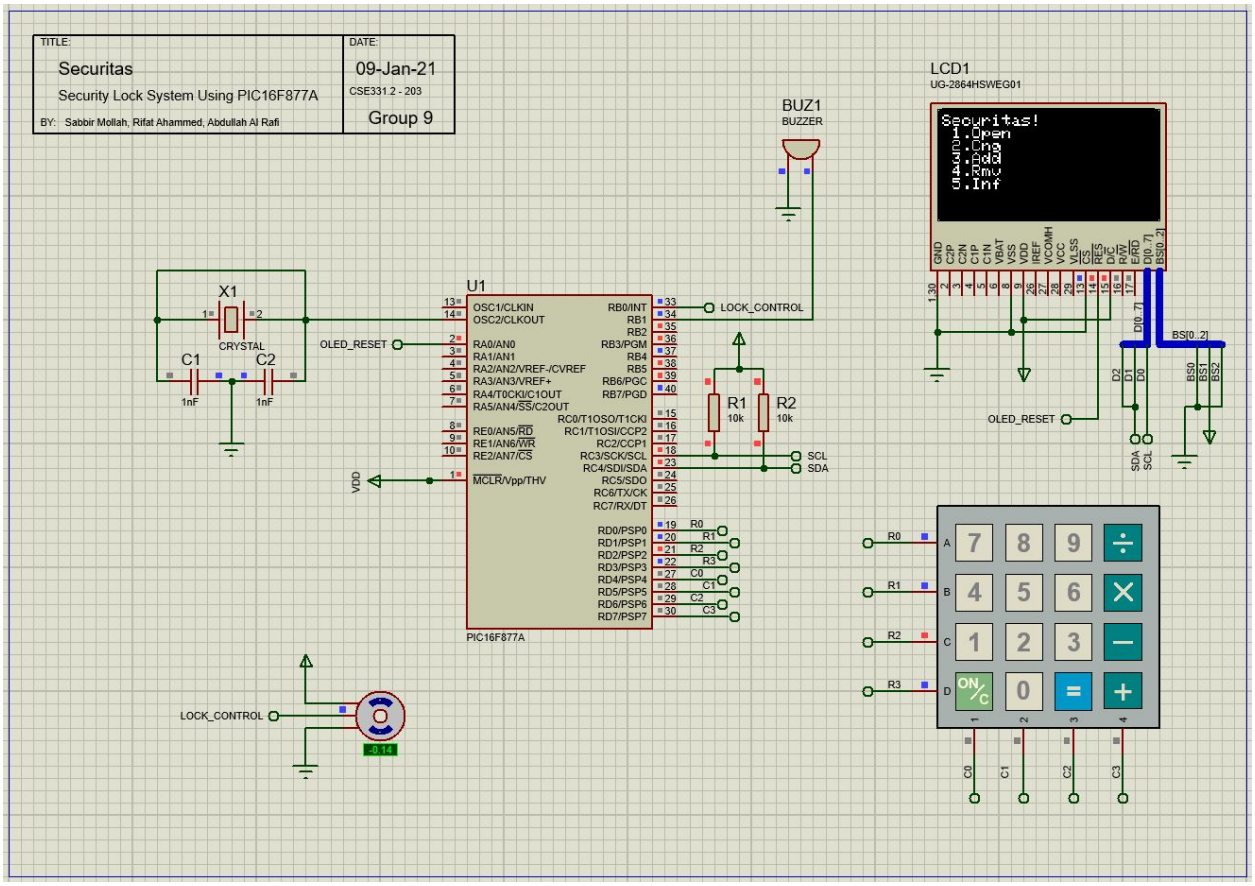
As this is our first time working with microcontrollers, it was an intimidating task to code separately and merge them afterwards. Hence, we avoided using Git or any version controlling systems. We rather set times together and worked by running the simulation on a single system and sharing the code by using the “Live Share” extension in VSCode code editor. This allowed everyone to contribute to the code in real time. This was doable as our team size is relatively slow, and we had to work on a single file.

Schematic Circuit

Components Used

- PIC16F877A
- SSD 1306
- UG-2864HSWEG01
- Keypad SmallCalc
- Motor PWMServo
- Resistor 10k ohm
- Buzzer
- Crystal

Screenshot



Code

The entire code of the project can be found in the following repository: https://github.com/SabbirMollah/Securitas-SecurityLockWithPIC

In our coding convention we have separated the views of the system in different functions. Each of these functions are coupled with another input handling function.

External Libraries

SSD1306.c : This library is used for the functions to operate with the LCD display.

Contribution

Name	**Contribution**	**Difficulty faced**
Sabbir Mollah (1712490942)	Display integration, programming structure, Memory read write, background study.	The display has a non-trivial wiring and programming scheme.
Rifat Ahammed (1721486642)	Door lock design, buzzer, implementation, internet research	Making the rotation at the precise degree , integrating the buzzer in the system.
Abdullah Al Rafi (1731371642)	Keypad design, implementation, internet research	Getting button input without power.

Findings

While doing the project, we faced problems in these sections.

1. **Memory overflow :** For adding too many lines in code, it took more memory to execute the system. So as a result , memory overflow occurs whenever we run the system.
2. **Large LCD display problems:** SSD1306 is slow and sometimes stops the whole system.
3. **Servo Motor :** The servo motor was not rotating exactly at 0 degree or 90 degree. It means, the door will not close or open properly due to this rotation problem.
4. **Buzzer :** We could not integrate the buzzer with the system
5. **Implementing multi user :** Due to the time limitation, we could not execute a multi user facility in our security system.

Conclusion

Throughout the whole making process of this project, we learned a lot of things. However we couldn't manage to complete the project due to our overestimating of the capabilities of a microcontroller.Consequently we faced difficulties in many ways while implementing it, but we learned a lot from those mistakes too. Overall , it was a good journey for all of us. We tried to do something through online and learned a lot of things.
We will subsequently work on this project by redefining the functionalities after having studied the microcontroller more.