Smart Locker

Qazi Fahim Farhan
1405052

Saad Muhammed Junayed
1405031

January 21, 2018

Contents

1 Introduction **1**

2 Hardware Requirements **2**

3 Software Requirements **2**

4 Flow Chart **3**

5 Block Diagram **5**

6 Circuit Diagram **7**

7 Basic Working Principle **8**
- 7.1 ATMega32 . . . 8
- 7.2 PIR sensor . . . 8
- 7.3 Buzzer . . . 8
- 7.4 LEDs . . . 8
- 7.5 16x2 Alpha-Numeric LCD . . . 8
- 7.6 4x4 Keypad . . . 8

8 Problems That We Faced **8**

9 Acknowledgement **9**

10 Conclusion **9**

1 Introduction

This is a security system project that can detect a thief/bulger and trigger an alarm. Suppose, you have a locker to store something valuable. If you have a key, you can open the locker in, say, 10 seconds. But if there is a bulger/thief, he will have to break the locker. This will take some time, say, 5 minutes.

Therefore, we can detect an intruder by measuring time. This is our main idea. In this project, we have interfaced a microcontroller with a PIR sensor. If a person stands in front of the locker (i.e., the sensor), the uC starts counting time. If the person goes away before a certain time (32 seconds in this case), the alarm wont be triggered otherwise it will. There is also a password system to turn off the alarm.

Click Here to watch the video!

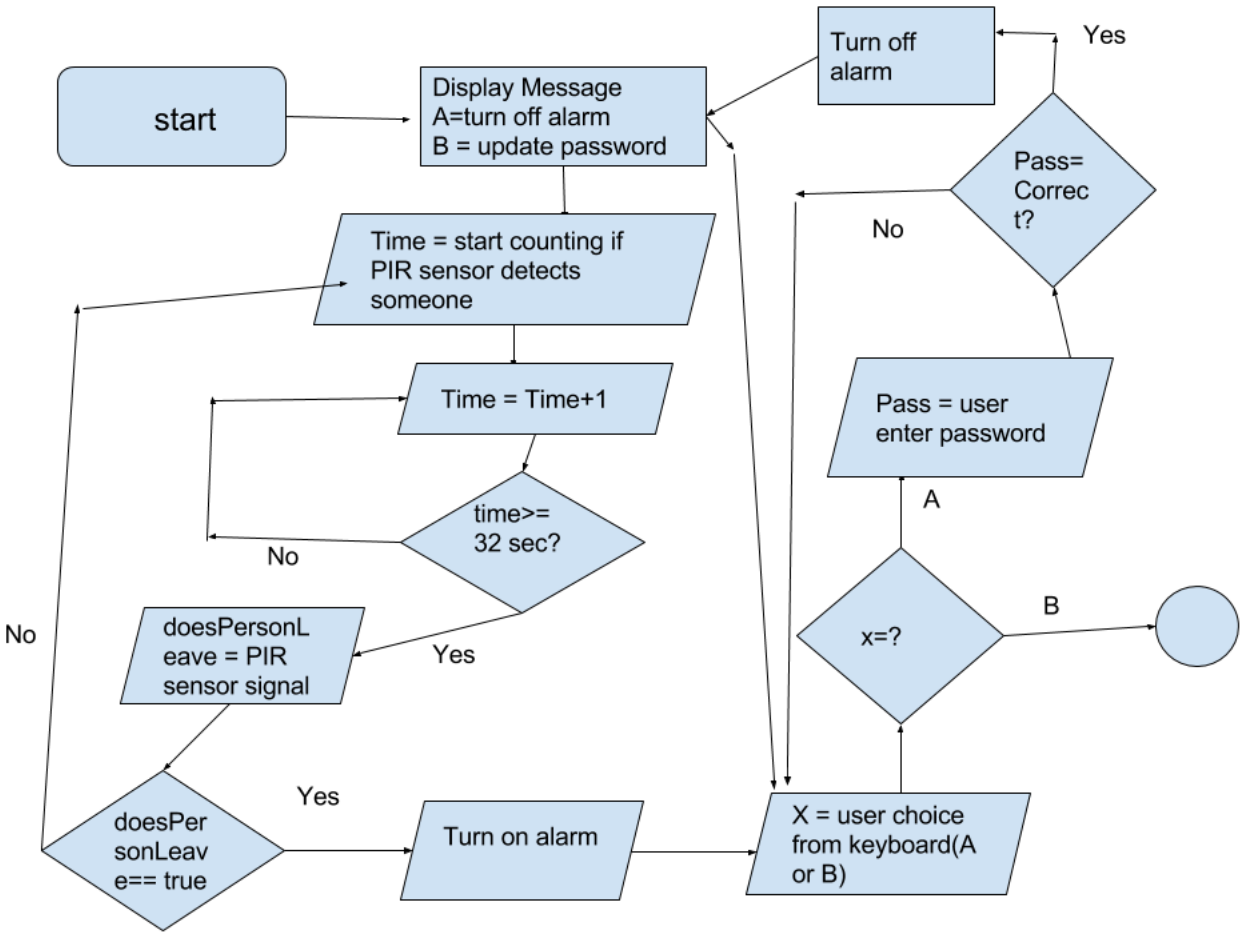
2 Hardware Requirements

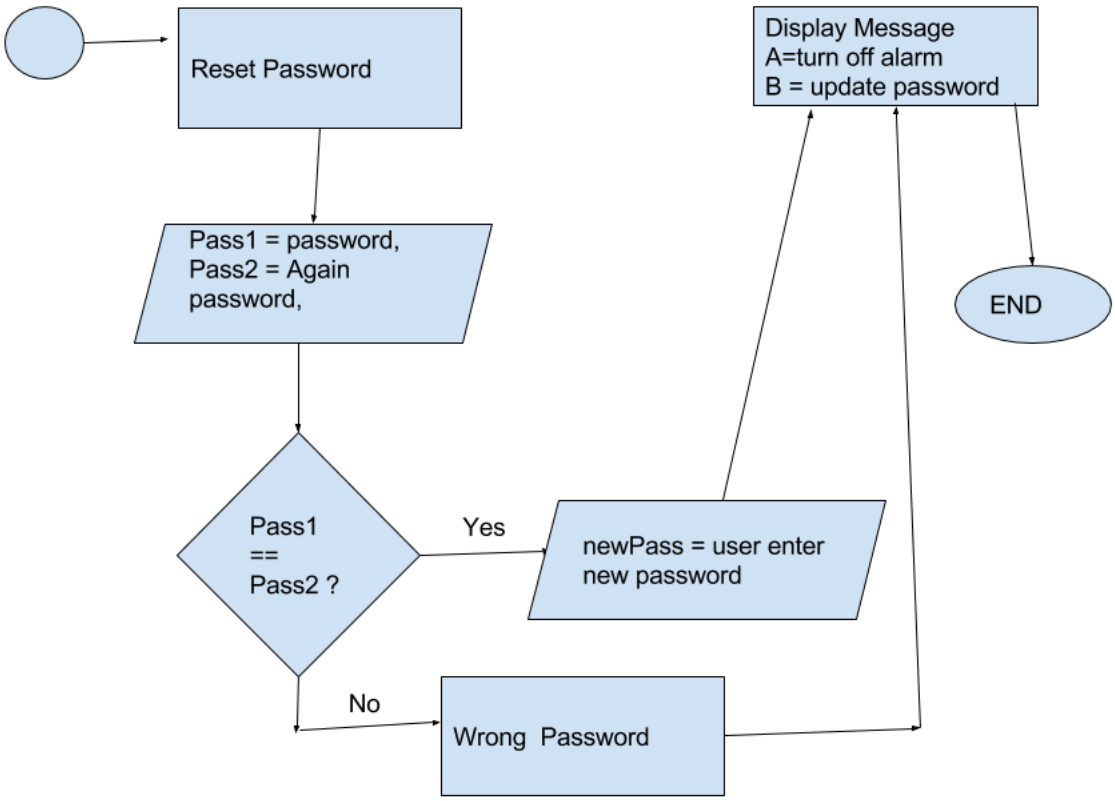
Equipment	Quantity
ATMega32	2
PIR sensor	1
Buzzer	1
Keypad 4x4	1
16x2 LCD	1
Wires	Many

3 Software Requirements

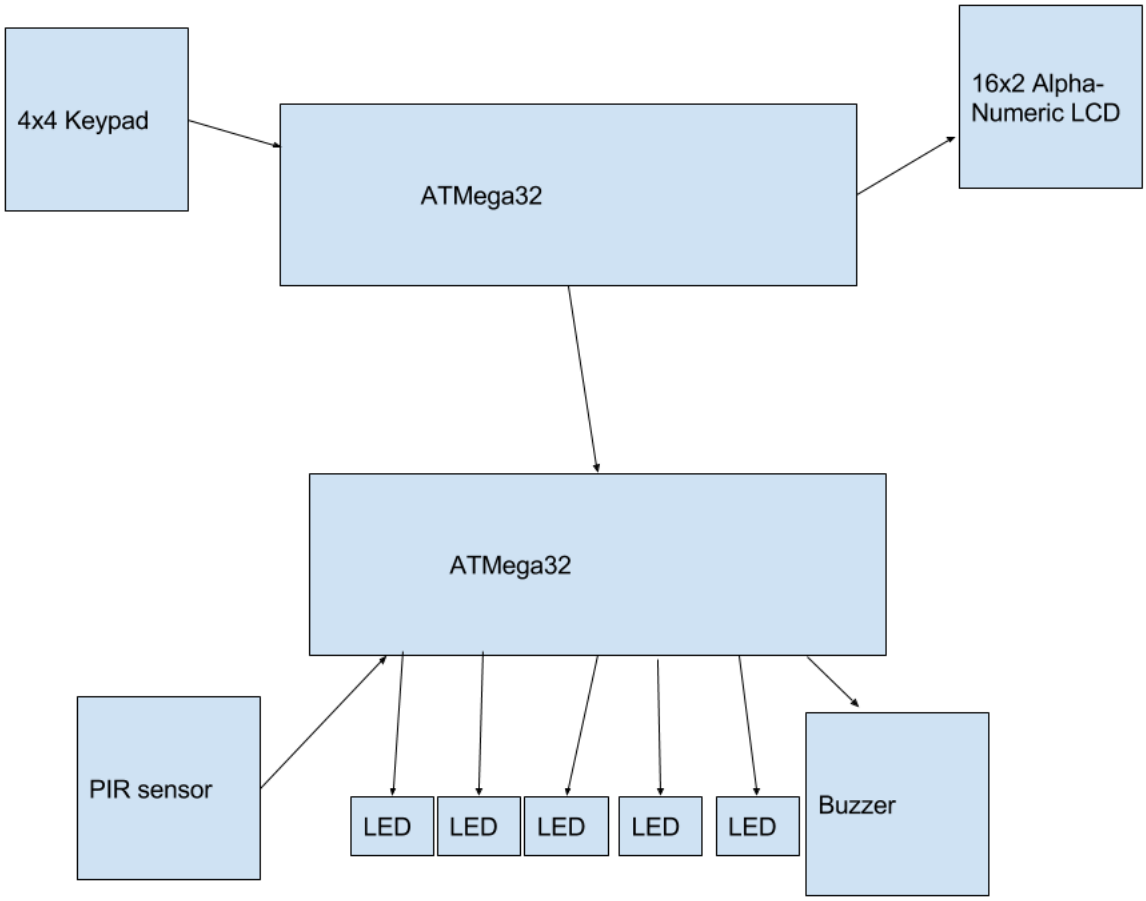
1. avrdude 6.2 (for compiling .c file, generating .hex file and burning code in Ubuntu/Linux Mint)
2. Proteus 8.6 Professional (Circuit Desiging)
3. Any Text Editor/IDE

4 Flow Chart

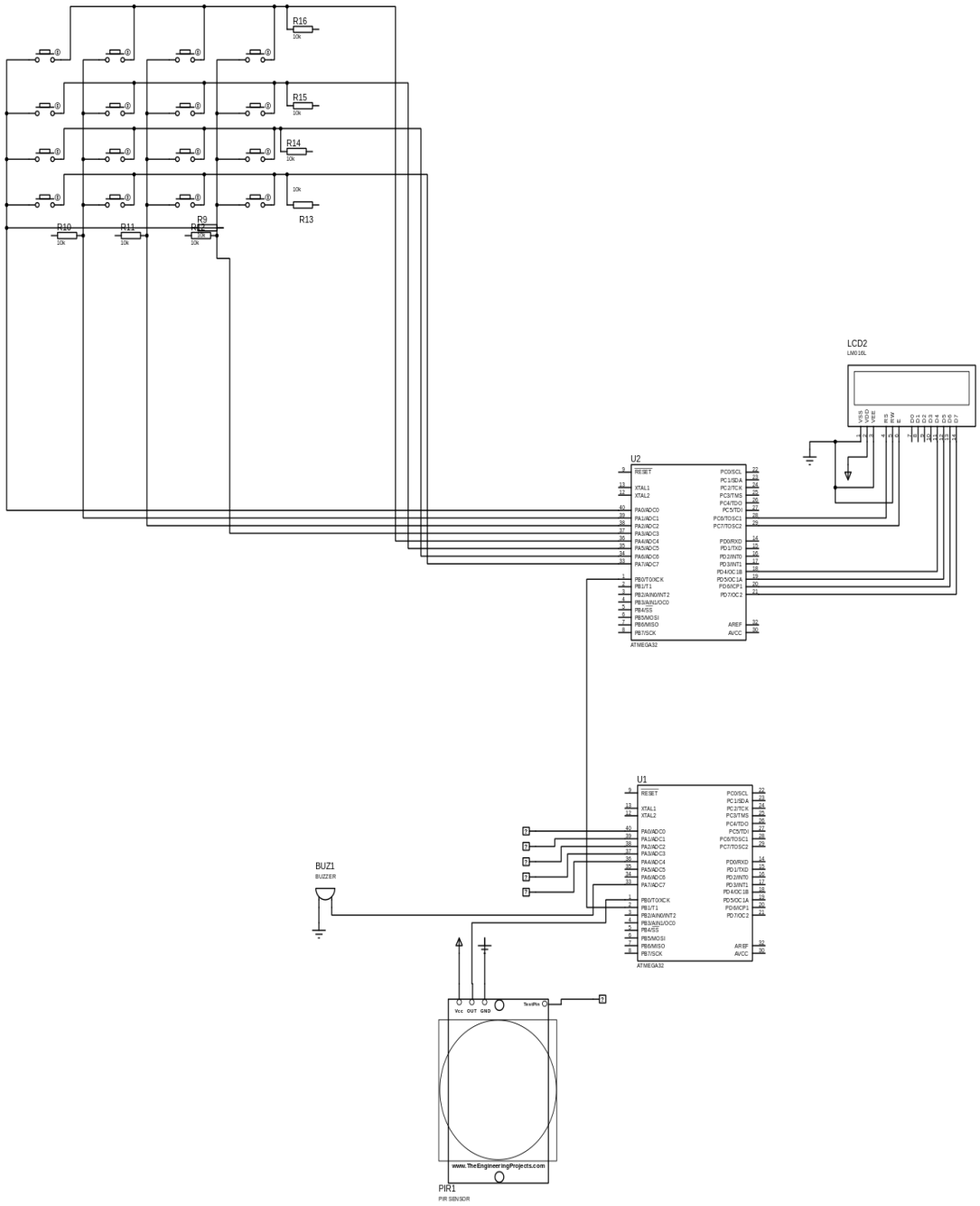

start
Display Message
A=turn off alarm
B = update password
Time = start counting if
PIR sensor detects
someone
Time = Time+1
time>=
32 sec?
No
doesPersonL
eave = PIR
sensor signal
No
Yes
doesPer
sonLeav
e== true
Yes
Turn on alarm
Turn off
alarm
Yes
Pass=
Correc
t?
No
Pass = user
enter password
A
x=?
B
X = user choice
from keyboard(A
or B)


Reset Password
Display Message
A=turn off alarm
B = update password
Pass1 = password,
Pass2 = Again
password,
END
Pass1
==
Pass2 ?
Yes
newPass = user enter
new password
No
Wrong Password

5 Block Diagram



6 Circuit Diagram



7 Basic Working Principle

7.1 ATMega32

We have used two Atmega32. The first one (uC1) deals with the PIR sensor and buzzer. The other one (uC2) deals with keypad and LCD. There is a connection from PB1 of uC1 to PB0 of uC2.

7.2 PIR sensor

The sensor detects human presence. The cathode is connected to VCC(5 V), anode to GND and OUT pin to PB0 of uC1.

7.3 Buzzer

It creates the alarm. The anode is connected to GND and cathode to PA7 of uC1.

7.4 LEDs

We have 5 LEDs to help us visualize the counter. Their cathodes are connected to PA0 t PA4 of uC1 respectively.

7.5 16x2 Alpha-Numeric LCD

The LCD is connected with uC2. Here, p0 goes to GND, p2 to VCC, p3(contrast) to GND, p4 to PC6, p5 to GND, p6 to PC7, p7 to p10 are not connected and p11 to p14 are connected to PD4 to PD7 respectively. P15 to VCC, p16 to GND.

7.6 4x4 Keypad

The 8 pins of the keypad were conneced with PA0 to PA7 of uC2 respectively.

8 Problems That We Faced

At first we had some problems with interfacing the LCD with microcontroller. Nothing was showing in the LCD. Later we removed connections and connected the LCD again and again until somehow it solved. We don't know what happened back then actually.
Secondly, we had some problems with EEPROM for saving password. Whenever we burnt the code, the password would get lost. So, in order to solve it, we made some changes. Whenever the device was powered on, the microcontroller checked if there was garbage value in the memory or not. If there was garbage, the code had been just burnt and so we set a default password. If no garbage was found, it was a password and so we didn't write the default password. That

way, we solved the problem.

Thirdly, we faced some problems with the keypad. So we debugged it by printing corresponding values in the LCD. We then made proper modifications to resolve the problem.

9 Acknowledgement

We took help from the following links:
http://www.electronicwings.com/avr-atmega/atmega16-eeprom
https://electrosome.com/interfacing-lcd-atmega32-microcontroller-atmel-studio/

10 Conclusion

The project helped us to have a better understanding on how to work with microcontrollers. We have learnt a lot of things. Our project can be used to enhance security. But we can make it even better by adding other functionalities such as sending mobile SMS to police, ATMega32 controlled traps etc. Also if you have any ideas, let us know and we'll see what we can do about it.
I hope you liked our project. Thank you!