

**INTRODUCTION TO EMBEDDED SYSTEMS**

**MAJOR PROJECT**

**DIGITAL SAFE**

**Student name:**

**Student Number:**

**Lecturer name:**

**Lab instructor:**

**Subject name:**

**Executive Summary:**

In this modern era all the banks, libraries and many smart homes require the safety locks where each person can access its belongings by entering password and make them safe. By this project we have proposed the economical, user friendly and low cost solution for digital safe by using a simple AVR32A microcontroller chip. The main achievement of this project is that the every person can set its own password of its safe and a large number of safes can be configured with the same chip and in the end the status of the entered number and the accessed lock is monitored real time. This project defines the main parameters and can be updated more by adding some other user friendly functionalities.

**Table of contents:**

1 Introduction.................................................................................................4

2 Background..................................................................................................4

3 Assembly code implementation..................................................................5

4 Block Diagram .............................................................................................7

5 Circuit Diagram............................................................................................8

6 Comparison of deliverables ........................................................................8

7 C Code implementaion................................................................................9

8 Result and discussion.................................................................................10

9 Conclusion.................................................................................................12

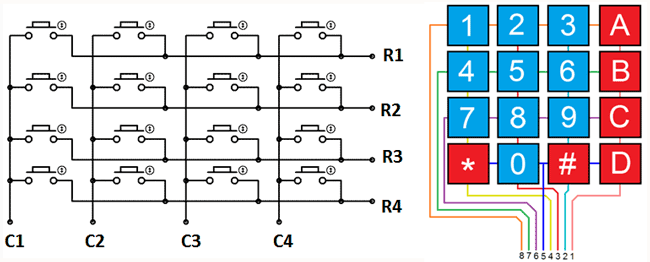
10 References.................................................................................................12

**Introduction:**

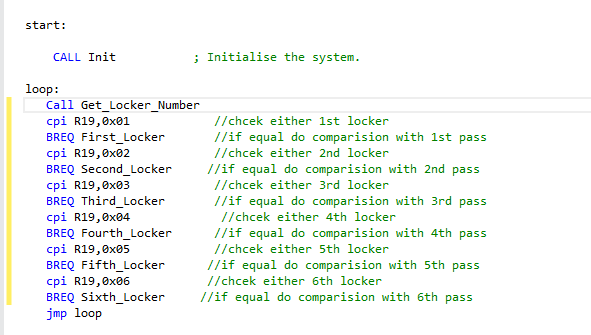
The basic aim of this project is to make a complex real world application using AVR microcontroller. In this project we have the AVR ATMEGA32A microcontroller as heart of this project. The programming of this controller is done in C and Assembly language. As input the Hex keypad and Dip switches are used. The Hex keypad is used for entering the password. On the other hand, the Dip switches are used for entering locker number for each user. For output the Leds are used. 4 leds are used for showing the current number entered on the keypad, while 3 leds are used to show the current locker number and 1 led for showing the current status of safe opened or not.

**Background:**

In this project we have used the 4x4 keypad. The arrangement of the buttons on the keypad is shown below.



The basic procedure of reading the button pressed on the keypad is that we declare rows as output and columns as input. First of all we send 0 on the first row and check all the columns as either any button is pressed or not. Then we send 0 to the next row and check all columns. In this way all rows are send 0 one by one and columns are scanned. After detecting the key pressed we show the button pressed on lower 4 pins of PORTB. The PORT B is declare as Output and half PORTD is declared as input and half is declared as Output.

**Assembly code explanation:**

Overall the working of the project is that the after initialization of the all the respective ports as inputs and outputs the code first check the locker number by reading the button pressed on dip switches after reading the locker number, the sssuser enters the number on the keypad. The number entered by user on keypad is saved in the register and this value is compared with the first number of that locker password. The locker password is already saved in the system. Similarly the Eight numbers are taken from the user and compared on by one with the predefined password. If both passwords are same then the safe opens and status leds glows. If password is wrong then the user cannot enter any number for next 8 seconds. This time can be increased. But for this project we have used time out as 8 seconds. There are many subroutines used in this but major are these:

**Matching:**

This subroutines matches the numbers entered from the keypad with the predefined password.

**Get\_both\_numbers:**

This subroutines takes the number from the keypad and saves that in register. Then it fetches the respective digit in password and also saves that in register for the comparison of both.

**Show\_Pressed\_number:**

This subroutines shows the number pressed on the keypad on the lower 4 bits of the PORTB.

**Show\_Status\_Led:**

This subroutines shows the status of the safe either it is opened or not. If 8th pin of PORTB is high then safe in on then safe is Open else it is Off.

**Show\_switch\_number:**

This subroutines shows the current locker number on the upper nibble of the PORTB.

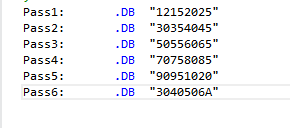
**Get\_Locker\_Number:**

This subroutines takes the status of the locker number by reading the dip switches.

**ReadKP:**

This subroutines reads number entered on the keypad and saves that in a register.

**Passwords:**

****

The passwords for each user are shown above. These can also be changed. The length of each password is 8 digits. The passwords can also contains alphabets like A, B,C,D and all digits from 0 to 9.

**Block Diagram:**

Start

Init

Check Lock No

1

6

5

4

3

2

Read keypad

Read Keypad

Read Keypad

Read Keypad

Read Keypad

Read Keypad

Compare with Pass2

Compare with Pass3

Compare with Pass4

Compare with Pass5

Compare with Pass6

Compare with Pass1

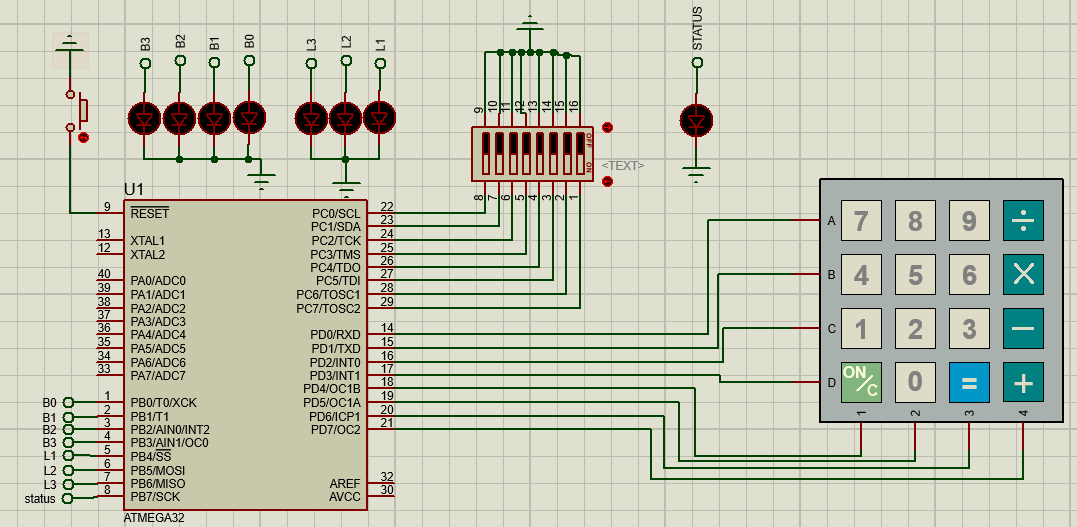
If Password Wrong

If Password Ok

Wait for timeout

Open safe

**Circuit Diagram:**

****

**Circuit Diagram Explanation:**

In this circuit the Keypad is attached on the PORTD and keypad is working as INPUT to the controller. The keypad is used for entering the password. Secondly the Dip switches are attached with the PORTC of the microcontroller. The PORTC is working as the input. The DIP switches are used for entering the Locker number. Finally, the Leds are used for showing the status of numbers pressed by the keypad, the current locker number and the status of the safe opened or not.

**Comparison between original deliverables and current deliverables.**

**Original Deliverable Current deliverable**

Each user has own locker No Each user has own locker No

Locker No should show on Leds Locker No should show on Leds

Current pressed no should show Current pressed no should show

On correct password Safe open on correct password Safe open

On wrong password timeout on wrong password timeout

**C code explanation:**



The logic implemented for C language code is same as for assembly. The only difference between C code and assembly code is the names of functions. The major functions which we have used here are:

void Init(void);

void check\_locker\_number(void);

void show\_status(void);

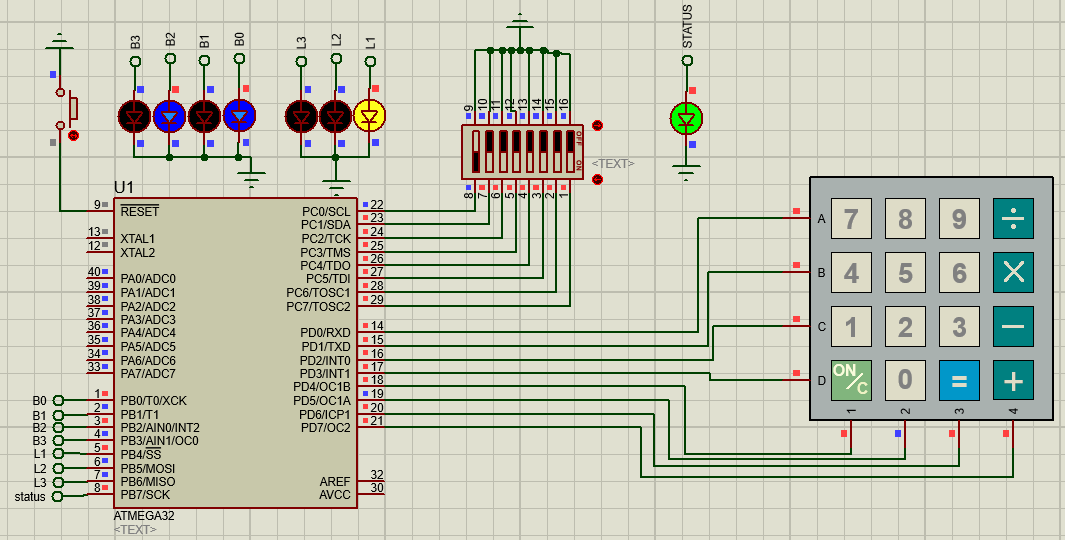
void show\_pressed(void);

void ReadSW(void);

void Disply (void);

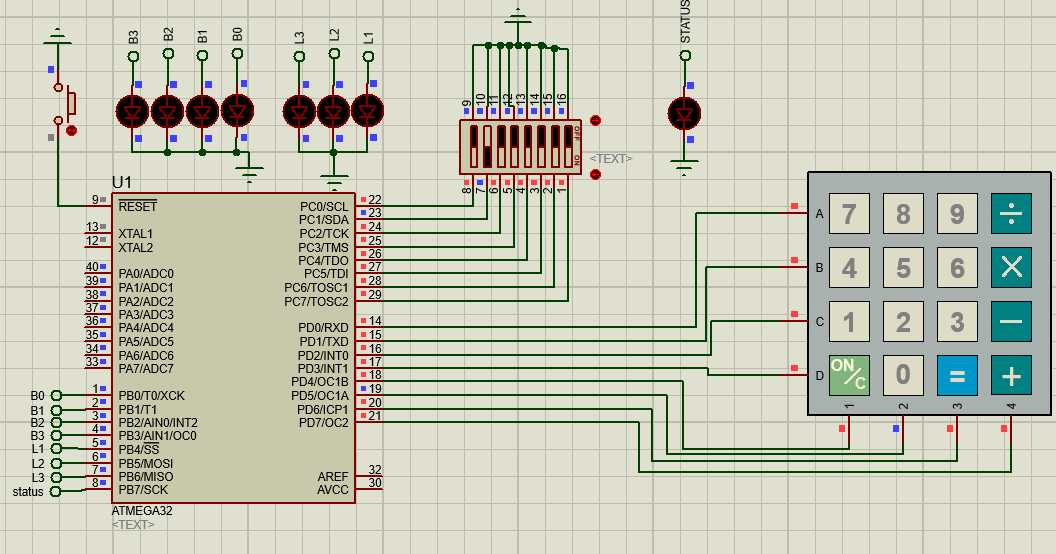
The function of all these functions is same as described above in assembly code section.

**Result and Discussion:**

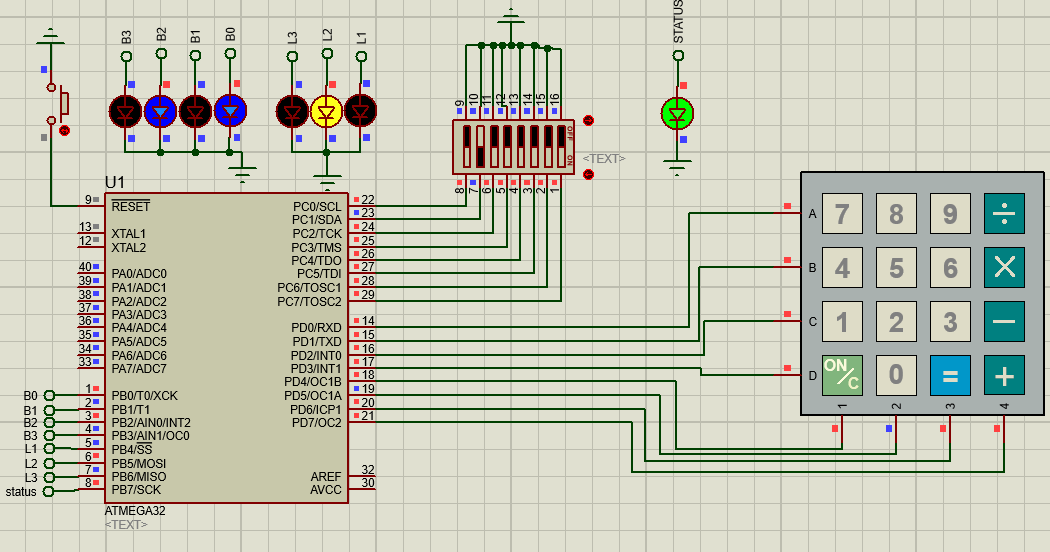
**When password entered for locker 1 is correct:**

The password for Locker1 is saved in system as “12152025”. The yellow Led show the status of locker no in binary as 1. The blue leds shows the current number entered on the Keypad. In this case it is 0101 which becomes 5. The Green Led show the status of the Safe. In this case the entered password is correct so the green Led is glowing and safe is open.

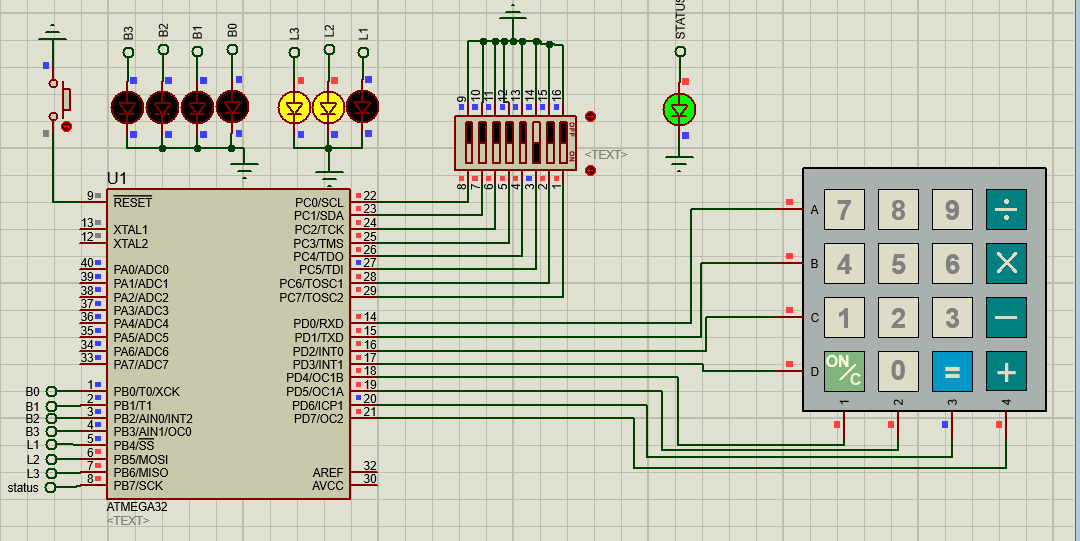
**When password entered is wrong:**



When wrong password is entered then the timeout is started and all leds become off for a specified time.

**When password entered for locker 2 is correct:**

The password for Locker2 is saved in system as “30354045”. The yellow Led show the status of locker no in binary as 2. The blue leds shows the current number entered on the Keypad. In this case it is 0101 which becomes 5. The Green Led show the status of the Safe. In this case the entered password is correct so the green Led is glowing and safe is open.

**When password entered for locker 6 is correct:**

The password for Locker6 is saved in system as “3040506A”. The yellow Led show the status of locker no in binary as 6. The blue leds shows the current number entered on the Keypad. In this case it is 0000 which becomes 5. The Green Led show the status of the Safe. In this case the entered password is correct so the green Led is glowing and safe is open.

**Conclusion:**

Concluding all, by this project we have understand how to make the digital safe device which is very common in our real world. By this project we have also learned how embedded systems is helping in solving the modern problems by efficient and economical way. Overall the system works perfectly fine but it can be modified more by adding some high level things like LCD in place of Leds. Also a motor can be attached in place of status led. Also the buzzer can be attached which can beep in case of wrong code. By doing these modifications the system can be made more robust and useful.

Reference:

1. Microchip, Atmel 8-bit AVR MCU FLASH Microcontroller, "Atmel-8155-8-bit -Microcontroller-AVR-ATmega32A\_Datasheet"
2. Microchip, Atmel 8-bit AVR MCU FLASH Microcontroller, "Atmel-8155-8-bit -Microcontroller-AVR-ATmega32A\_Datasheet", (2020 Reference)
3. Mazidi, Muhammad Ali - "AVR Microcontroller and Embedded Systems: Using Assembly and C", Pearson Custom Electronics Technology, First Edition, 2011,
4. Microchip, Atmel 8-bit AVR MCU FLASH Microcontroller, "Atmel-8155-8-bit -Microcontroller-AVR-ATmega32A\_Datasheet", (2020 Reference)
5. Mazidi, Muhammad Ali - "AVR Microcontroller and Embedded Systems: Using Assembly and C", Pearson Custom Electronics Technology, First Edition, 2011,