Software Requirements Specification

for

Door Locker Project

Version 1.0 approved

Prepared by Ahmed & Mostafa

<organization>

<date>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Mostafa Abdelrahman | 26/10 | Started and completed sections 1.1, 1.2, 1.3, 1.4, 2.2. | 1.0 |
| Ahmed Hussien | 29/10 | Completed sections 2.1,2.3,2.4,2.5,2.6 | 2.0 |
| Mostafa Abdelrahman | 30/10 | Completed sections 4.1,4.2,4.3 | 3.0 |
| Ahmed Hussien | 30/10 | Completed sections 3.1,3.2,3.3,3.4 | 4.0 |
| Ahmed Hussien | 3/11 | Updating Sections 4.1,4.2,4.3  & Adding section 4.4 | 4.1 |
| Ahmed Hussien | 4/11 | Completed SRS document | 4.2 |
| Ahmed Hussien | 9/11 | Update 2.7 Assumptions and Dependencies | 5.0 |

# Introduction

## Purpose

The purpose of this document is to state the software requirements for the two AVR atmega16/32 microcontrollers that will be placed in a door locker.

## Document Conventions

No defined standard is used. But code should be readable and maintainable.

## Intended Audience and Reading Suggestions

This document is useful for those who are interested in understanding the software requirements of the door locker product, this shall include developers, software designers and computer and electrical engineering students.

## Product Scope

The software specified in this document is developed using C programming language and the software should provide high quality software whenever is applied on the microcontroller.

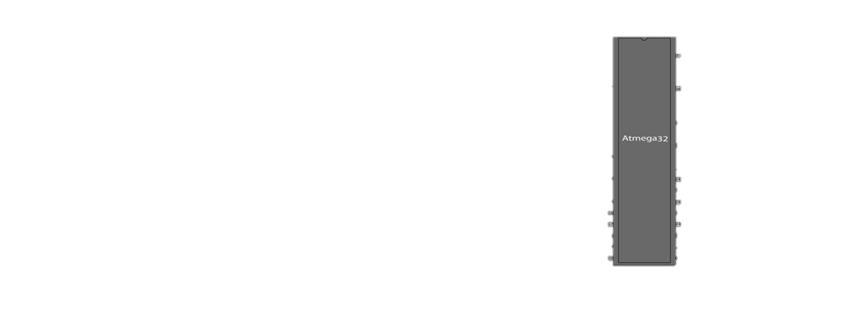
## References

N/A

# Overall Description

## Product Perspective

The overall product being specified in this document is the Door Locker. More importantly is the fact that this Door contains 2ECUs (Electronic Control Unit) requiring software. This software is the primary concern of this document. The upper block diagram shall illustrate the importance of software’s place in this project.



## Product Functions

This product supports:

* 4-Digit Password creation.
* System lock after multiple lock error entry.
* A buzzer
* Easy access to the locker row.
* LCD & Two LEDs For system status
* Password changing

## User Classes and Characteristics

The intended user(s) are those who need an electronic control system with medium security for the Locker or safe Door. It can be used also as a learning project for those who are learning embedded systems.

The system will be used by customer as a safe locker the user will have only access to the ECU-HMI through keypad and LCD which display system response along with two LED’s and the buzzer

With respect to software, no user shall have access once the main program is “burned” into the Atmega32 microcontroller. But code will be available for those who want to learn from it

## Operating Environment

The operating environment is targeted to be inside a safe. The system will run on batteries.

With respect to software the operating environment shall be an embedded system. A primary program shall be written using the C programming language in Atmel studio and compiled with CCS C Compiler. The hardware environment shall be an Atmega32 microcontroller. The software must work in tangent with hardware components such as LEDs, LCD, Keypad, D.C Motor, Servo Motor, Stepper Motor and EEPROM. The embedded software “burned” into the microcontroller must be the “glue” that holds our door locker system together.

## Design and Implementation Constraints

### Hardware Constrains

* The Designer have to use atmega32/16 microcontroller
* LED’s color
* 1K EEPROM to carry password
* LCD must be 4x20
* Microcontrollers are using 8MHz oscillator
* 3 motors (dc withl293d,stepper with uln2003 ,servo)
* UART used for communication between two ECUs with BAUD rate 9600 bits/sec
* CNTRL-ECU must use I2C to communicate with the EEPROM

### Software constrains

There are a lot of software constrains that will be discussed later in this document through the implementation. Mainly constrains are on massages displayed and when to display it. One of this constrains for example is to display the process running on the microcontroller.

For example if opening the door display on LCD “opening the door “.

## User Documentation

Documentation for the software developed for the Door Locker project shall consist of this document, a High level design document and a detailed design document.

## Assumptions and Dependencies

* Opening door lock of DC motor requires motor to rotate with maximum speed on anticlockwise for 1 sec
* Closing door lock of DC motor requires motor to rotate with maximum speed on clockwise for 1 sec
* Opening the door will require the Servo motor to rotate 360 degree cycle in the clockwise direction
* Closing the door will require the Servo motor to rotate 360 degree cycle in the anticlockwise direction
* Opening door also require u to enter a value between 0 and 180 to choose locker u want to open
* Closing door require the servo motor to be in initial position

# External Interface Requirements

## User Interfaces

User will communicate with system using keypad and LCD4x20. User is allowed to do specific functions like changing password, opening and closing the safe door and selecting row which he will open. The system will respond to user’s input with massages on LCD which will tell the user what to do.

## Hardware Interfaces

After burning the program on our two atmega32/16 microcontrollers, one microcontroller is expected to interface motors through different ICs. Other hardware components as described before is expected to interface with one of two microcontrollers like LEDs, LCD, EEPROM and keypad. For more details refer to design report.

## Software Interfaces

Software shall be written in C using Atmel studio. Software interfaces will only exist prior to placement of the primary program onto the microcontroller. The embedded system has no interaction with any other software systems.

## Communications Interfaces

The system consist of two ECUs that is expected to communicate together using UART with a baud rate of 9600 bits/sec. The CNTRL-ECU also will have to communicate with the EEPROM using the I2C communication protocol

# System Features

System will have to provide the features described in the use case diagram, For further details about each functions see the sections below.

We have tried in this coming section to avoid the redundancy in requirements so requirements should apply to all of the system. For example in HMI-ECU first requirement apply to all features also all requirements of feature 1 apply to feature 4 so requirements should be think of system wise not feature wise.

## Setup system password.

### Description and Priority

The user should be able to setup the system password when the system starts for the first time. The password consists of 4 characters followed by ‘=’ to enter password. User should be able to clear the screen if he miss clicked wrong character and enter password again. The system should ask the user also to re-enter the password to ensure he know the correct password. After the password is set the system should ask the user to restart system. After system restart, system ask user to enter password to open the safe.

### Stimulus/Response Sequences

As shown in the sequence diagram this should be the successful scenario of setting up the password for the locker system. While confirming password entered is right the green LED flash on indicating the system is unlocked.

If a password is entered wrong the system should display “plz enter password“ again. If the password is entered 3 times wrong the system go to the unsuccessful scenario.

As for the unsuccessful scenario (wrong entrance of the password 3 times) the system should light on red LED, the buzzer beep forever and the system display “the system is locked” until u close the system

On Display 3 options part the system should display the 3 options

“1-open door

2-close door

3-change password”

Then wait for input from user to determine the option that will be executed. More details will come later about each option.

### Functional Requirements

#### HMI-ECU

REQ-1.0: software should be able to interact with user Displaying messages on LCD and taking input from Keypad

REQ-2.0: software should be able to follow the given sequence

REQ-3.0: software should be able to communicate with CNTRL-ECU Using UART

#### CNTRL-ECU

REQ-1.0: software should be able to communicate with HMI-ECU using UART

REQ-2.0: software should be able to communicate with EEPROM using I2C

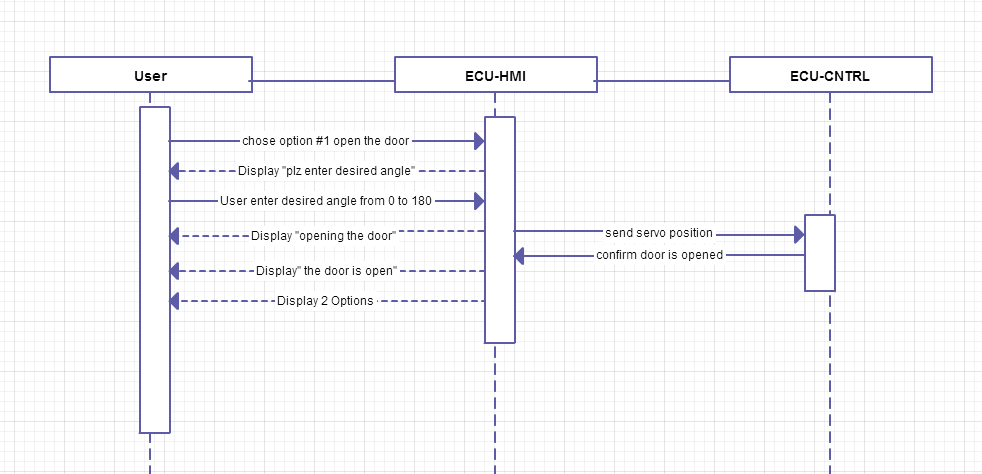
REQ-3.0: software should be able to compare password with one saved on the EEPROM and return true or false value

## Open the door

### Description and Priority

When user enter 1 this option is chosen in this option the CNTRL-ECU Using a DC motor, stepper motor and a servo motor, can let the user easily reach the desired locker. The DC motor is used to open the door of the locker. The servo motor is adjusted to let the door stop just at the desired locker row (Entered by the user) to access. Finally the stepper motor unlocks the desired locker.

### Stimulus/Response Sequences

The successful scenario is shown above. Some errors must be accounted for like if user didn’t enter an angle from 0-180 the system should display “ wrong input “ for one second and then go back to “plz enter desired angle “. It will be good to add clear button so that the user can undo his mistake. Displaying the “door is open should be for one second. Then return to displaying the two options “1- Close the door

2- Change the password “

### Functional Requirements

#### HMI-ECU

REQ-4.0: software should be able to follow the given sequence

REQ-5.0: software should be able to account for wrong inputs by the user

#### CNTRL-ECU

REQ-3.0: software should be able to deal with DC, Servo and Stepper Motor open and close the door

REQ-4.0: software should move the servo to the position given by the user

## Close the door

### Description and Priority

The software should be able to return the motors to its initial positions when this option is chosen and close the door of the safe.

### Stimulus/Response Sequences

This is the successful scenario for closing the door. “Closing the door “ message will appear until the confirmation from CNTRL-ECU. As for “the door is closed” message it will appear for only one second then system should display the 3 options

### Functional Requirements

#### HMI-ECU

REQ-6.0: software should be able to follow the given sequence

#### CNTRL-ECU

REQ-5.0: software should be able to return the motors to its initial positon closing the door

## Change password

### Description and Priority

When change password option is chosen the system allow user to change password saved on the EEPROM.

### Stimulus/Response Sequences

If this option is chosen the system should follow the sequence given in section 4.1.2 system should follow same sequence

### Functional Requirements

#### HMI-ECU

REQ-7.0: software should be able to follow the given sequence when this option is chosen

#### CNTRL-ECU

REQ-6.0: software should be able delete the old password replacing it with new given one.

# Other Nonfunctional Requirements

## Performance Requirements

In our system the performance is very important. With the performance we mean how fast dose the system interact with user, dose the system does its job as described and is it a bug free system.

## Safety Requirements

N/A

## Security Requirements

N/A

## Software Quality Attributes

Software should perform well with a good response time. Characters that interact with user on LCD should not have much delay.

## Business Rules

N/A

# Other Requirements

N/A

Appendix A: Glossary

ECU: electronic control unit

HMI: Human interface

CNTRL: control

UART: Universal asynchronous receiver/transmitter

EEPROM: Electrically Erasable Programmable Read-Only Memory

Appendix B: Analysis Models

N/A

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

Source: http://www.frontiernet.net/~kwiegers/process\_assets/srs\_template.doc