Fingerprint Lock Access Design Document



Abstract 3

Connection 6

Functional Requirements 4

How this lay out should work 7

Knowledge Required 3

Non-Functional Requirements 4

Who Should Read This Document? 3

# Abstract

In this document you’ll find the functional requirements and non-functional requirements for the project involving 2 microcontrollers with different architectures, a biometric sensor which is fingerprint sensor, different bus communication like USAR and PSI and an actuator to simulate the open and close motion of a safe box blocking mechanism or a door lock.

# Who Should Read This Document?

Any person that can benefit from this project to use it as leverage for their own implementation.

# Knowledge Required.

Basic understanding of electronics diagrams, basic knowledge of serial bus configurations and functionality for USART and SPI, basic knowledge of C and C++ programming, microcontrollers and FreeRTOS.

# Functional Requirements

Here find the functional requirements needed for the project, the ones marked with the hollow bullets are requirements that can be treated as low priority:

* System must record up to 5 Fingerprints records
* System must open mechanism upon recognized fingerprint pattern
* System must no allow mechanism to open if fingerprint is not recognized
* System must turn on a green led upon access granted
* System must turn on a red led upon access denied
* System must have an enroll sequence triggered by a button, or switch
* System must have an access sequence triggered by a button or a switch
* System should store the enrolled printers in a non-volatile memory

# Non-Functional Requirements

Here find the non-functional requirements gathered for the system

* Fingerprint enrollment must have 80% of reliability
* Open mechanism should grant only access by moving the servo in no more than 3 seconds
* Open mechanism must reset close position in no more than 2 seconds.
* Reliability on data transfer from mcu to mcu
* Robustness on data access
* Error Traceability

# Involved Hardware

In this section you’ll find a list of the involved hardware and references for disclosures of why some elements needed to be removed or marked as optional.

* Microcontrollers
  + STM32F103C8T6
  + Arduino 1
* Peripherals
  + JM-101B Fingerprint Sensor
  + Servo 5 Kg
  + Dip switch
  + Red Led
  + Green Led
  + Blue Led
  + 330 Resistor
  + USB-TTL
  + SPI Micro SD Memory Slot[[1]](#footnote-1)
* Mount
  + Protoboard
  + Dupont Cabled

# POC Connection Diagram

A screenshot of a cell phone

Description automatically generated

The SD Bus is an optional implementation, since for POC, the registry memory of the fingerprint sensor is more than enough.

So Implementation of the embedded file system will be out of scope for this POC (prove of concept).

# How POC Should Work

## Enroll Fingerprint Mode

For the enroll fingerprint behavior the dipswitch channel 7 should on and channel 6 must be disabled so the Arduino mcu will enter in enroll subroutine and enable the sensor to start treading and store the image template on the specific index, on successful index the blue led should blink.

Once user is done with the fingerprint enrollment, dipswitch channel 7 must be disabled, so the other mode can be selected

Here is how the flow goes related to the diagram shown before:

1. Dipswitch channel 7 on and channel 6 off, Arduino reads digital input
2. Start enrolling routine
3. Connection between Arduino and Sensor is serial
4. Upon success blue led ( not included in the diagram yet) will blink and data will be sent via SPI to stm32 for storage on future implementation, for now stm32 will send a 0x31 (1 character) to putty terminal.
5. On wrong enrollment no led will blink and a 0x30 (0 character) will be printed on terminal

## Access by Fingerprint Mode

For the “access by fingerprint” mode which is the main routine for this POC for granting or not access, or opening or not the lock, the dipswitch channel 6 must be enabled and channel 7 must be disabled, if two inputs are high red led will be turn on.

For Access granted the servo will move to a open position ( rotate 180 degrees) maintain this position by 3 seconds and then return to close position in no more than 2 seconds, also the green led will blink, con contraire caise, access denied which is triggered when the fingerprint is either not recognized or not enrolled the red led will blink and no servo motion will be detected.

1. Dipswitch channel 6 on and channel 7 off, Arduino reads digital input
2. Start access routine
3. Connection between Arduino and Sensor is serial
4. Upon known and accepted fingerprint 0xAB is sent to STM32 via SPI , stm32 process this data and after comparing a 0xBA is returned to Arduino to send the pulse to servo and make it turn 180 degrees (open position) and green led blinks
5. On wrong access red led will blink and no data is being sent to stm32
   1. Wrong access can be triggered by a non registered fingerprint or bad fingerprint read.

# Functional Diagrams

TBI

1. This is an optional involved hardware as the JM-101B Sensor contains a memory able to store up to 127 fingerprint samples. So, at this implementation won’t be necessary to implement this hardware unless convenient. [↑](#footnote-ref-1)