

November 12th, 2015

Dr. Andrew Rawicz

School of Engineering Science
Simon Fraser University
Burnaby, BC V5A 1S

Re: Design Specification for Arca by Certus Engineering

Dear Dr. Rawicz,

Enclosed is the Design Specification for our Automated ID Scanner named Arca. Our proposed design is aimed to be used for Identity verification and attendance tracking.

This Design Specification is intended to give you an overview of our design. This includes cost considerations, project schedule, and the technical aspects of our product. This proposal discusses the utilization of the design as well as forecasting perspective challenges we might face in the process of designing this product.

Our company consists of four engineering science students with backgrounds in software/hardware, electronics, real time embedded systems and mechanical design bringing their knowledge together to create Certus Engineering. We also have one consultant and two additional members with a background in business, graphic design and media relations.

Should you require any additional information regarding our product, please do not hesitate to contact via phone at (604)349-8182 or via email at ffa8@sfu.ca or farshad@certusengineering.com.

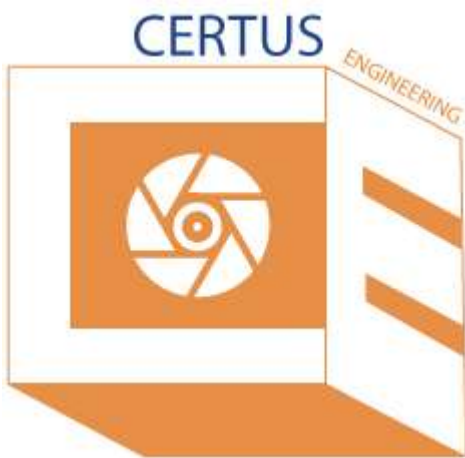
Sincerely,

Farshad Farshadi

A handwritten signature in blue ink, appearing to be "Farshad", is written over a light blue circular stamp.

Chief Executive Officer

Enclosure: Design Specification for an Automated ID Scanner, the Certus Engineering.



Certus Engineering Team:

Chinmay Desai

Farshad Farshadi

Kevin Sabau

Paniz Bertsch

Contact: ffa8@sfu.ca

farshad@certusengineering.com

Prepared For:

Dr. Andrew Rawicz – ENSC 440

Mr. Steve Whitmore – ENSC 305

School of Engineering Science

Simon Fraser University

Submission Date:

November 12, 2015

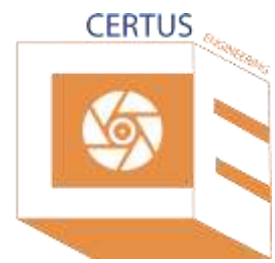
Revision 1.1

Additional Team Members

(Not Enrolled in ENSC 305/440):

Natasha Farshadi

Juri Petrouchtchak



Executive Summary

Arca is a digital identification system and can be customized to use RFID tags, barcode or ID number for identification purposes. Arca is currently in development phase by Certus engineers who are committed to provide a reliable, functional and safe solution to human recognition and security problems. Therefore, detailed project plan and requirements must be documented to insure compliance to standards and customer needs.

This document covers design specification of Arca, digital identification system. Main objective of this document is to provide guidelines in development and implementation of our product and functional specification.

Following topics are discussed in this design document:

- Overall system design
- System requirement with justifications and details
- Tools and data models
- Design methodology and function implementation
- Components' interaction within the system
- Test plan, coverage and deadlines

Table of Contents

1. Introduction	8
1.1 Scope:	8
1.2 Intended User:	8
2. System Overview	9
3. System Specifications & Justification	10
3.1 General Specifications	10
3.2 Electrical Specifications	12
3.3 Mechanical Specifications	13
3.4 Software Specifications	14
3.4.1 General Software	14
3.4.2 Optical Character Recognition	16
3.4 RFID Specifications	20
4.1 Unit Tests:	24
4.2 Integration Testing:	24
4.3 System Testing:	25
4.4 User Acceptance Testing:	25
5. Conclusion	26
6. References	26

List of Figures

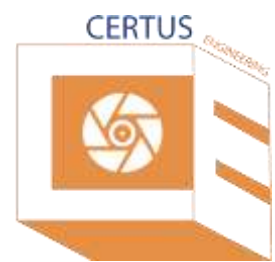
Figure 1.....	10
Figure 2.....	11
Figure 3.....	11
Figure 4.....	12
Figure 5.....	13
Figure 6.....	16
Figure 7.....	16
Figure 8.....	17
Figure 9.....	17
Figure 10.....	19
Figure 11.....	19
Figure 12.....	20
Figure 13.....	21
Figure 14.....	21
Figure 15.....	22
Figure 16.....	24

List of Tables

Table 1	15
Table 2	21
Table 3	24

List of Acronyms

ABS	Acrylonitrile Butadiene Styrene
ASCII	American Standard Code for Information Interchange
CCD	Charged-Couple Device
CDF	Cumulative Distribution Function
CMS	Content Management System
CMOS	Complementary Metal Oxide Semiconductor
CSA	Canadian Standards Unit
CPU	Central Processing Unit
ENSC	Engineering Science
GUI	Graphical User Interface
HW	Hardware
HTML	Hyper Text Markup Language
ID	Identification
kHz	Kilohertz
LED	Light Emitting Diodes
MHz	Megahertz
MS	Microsoft
OCR	Optical Character Recognition
OS	Operating System
PC	Personal Computer
PCB	Printed Circuit Board
PDF	Probability Density Function
PHP	Personal Home Page / Hypertext Preprocessor



RFID Radio Frequency Identification

SFU Simon Fraser University

SQL Structured Query Language

SW Software

UI User Interface

UID User Identification

USB Universal Serial Bus

1. Introduction

Arca is an enclosed system, designed by Certus Engineering, used for personnel verification and attendance tracking for financial or educational institutions. Our goal is to improve accuracy in the authentication process for business enterprises in a cost effective way without compromising on efficiency. This device will read and authenticate barcode, identification number, and the RFID value. Depending on the user's specifications, the device can be set to read any of those three features.

1.1 Scope:

This document is designed to outline a detailed design specification by Certus Engineering for the product, Arca. This will include specifications on software, electronic components, mechanical design, and any other design tools and parameters used to successfully manufacture this product. This document builds on the requirements set forth in the functional specification.

1.2 Intended User:

The intended user of this product will range from students/professors/lab technicians at educational institutions to industry professionals operating a business enterprise. The target demographic will be young adults in the age group 18-25 and adults in the 35-50 age group. The first group will be defined as students and the latter would be the professors/business officials. This attendance tracking could potentially be used in elementary/secondary schools and therefore children and teens would also operate this device but are advised to do so with adult supervision.

2. System Overview

This device will read a barcode, an identification number, and the RFID value. Depending on the user's specifications, the device can be set to read any of those three features. This system has four main stages: Detection, Authentication, Distribution, and Collection.

Detection is where the user is given an ID card with an RF UID tag embedded and the user would tap it against the RFID reader. In this section, the card is read properly the UID is pending authentication.

Authentication is where the UID and the ID numbers are cross referenced and it determines if the person's credentials are legitimate or purely fictional in hopes to deceive the system.

Distribution only occurs if the task is to dispense any parts/components/equipment. After the credentials have been verified, the user is permitted any equipment.

Collection is the last stage, and this is where all the data amassed is saved. This allows the administrator to keep a track of all participating members and allows them to further examine this data.



3. System Specifications & Justification

This section provides a detailed view in the basic system specifications and descriptions, as well as the justification for the product. In junction with the functional specifications outlined previously, this section will follow a similar numbering pattern when discussing justifications for each of the main four functionality requirements.

3.1 General Specifications

As described in the functional specification, Figure 1 is the block diagram for Arca:

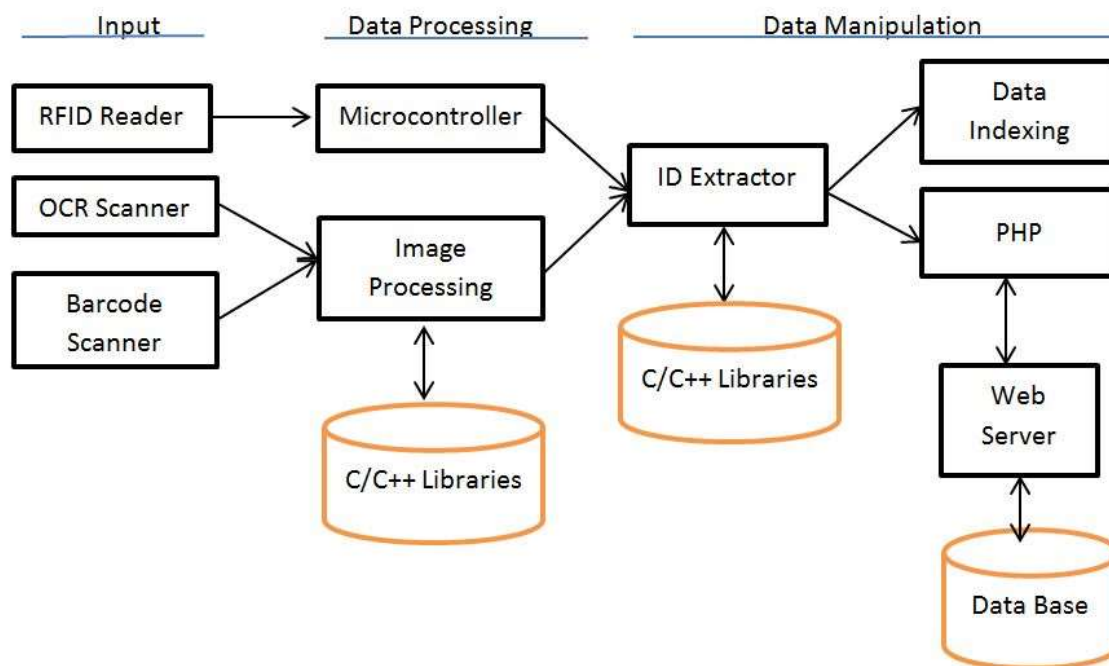


Figure 1: Block diagram for the Arca unit

The unit itself will consist of a 13.56 MHz RFID reader, an Arduino Nano microcontroller, two CMOS sensors and two lenses with F5.6 sensitivity and 550mV saturation signal [1]. We decided to use the Arduino due to its versatility and cost effectiveness with respect to the scope of this project. Once the product is in production stage, we will implement our own microcontroller, such that all electrical components can be housed on a single board. Purchasing one Arduino was found to be cost efficient rather than ordering a single PCB to replace the Arduino. In the long run, printing our own circuit board will be economical while also giving us freedom to customize it to our specification.

We had two potential candidates in selecting the optical sensor. First was a CCD sensor, which turns analog light signals into digital pixels. CCD allows for a higher resolution picture with minimal to no distortion, but at a higher cost. The second was a CMOS sensor, which uses transistors at each pixel to move the charge through wires. CMOS technology is relatively new when compared to its CCD counterpart, and it produces an image that almost on par with that of a CCD. The images from a CMOS are slightly inferior to CCD which justifies its lower cost but in recent years the gap in between the two is closing and soon CMOS might surpass CCD in terms of quality of the images. This coupled with its economical pricing, we chose the CMOS sensor to be used in our product.

From Figure 2, in order to use the CCD, all horizontal and vertical shift registers must continuously relay data as electronic signals. This becomes an issue during high speed applications and power consumption is significantly higher. From Figure 3, we see that the CMOS only needs to one readout column of circuitry, thus increasing the speed and lowering the power consumption.

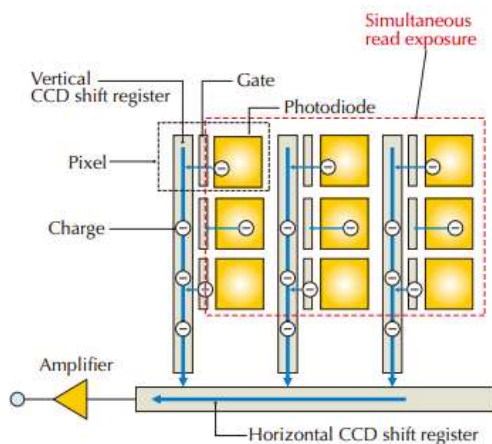


Figure 2: CCD Sensor configuration [1]

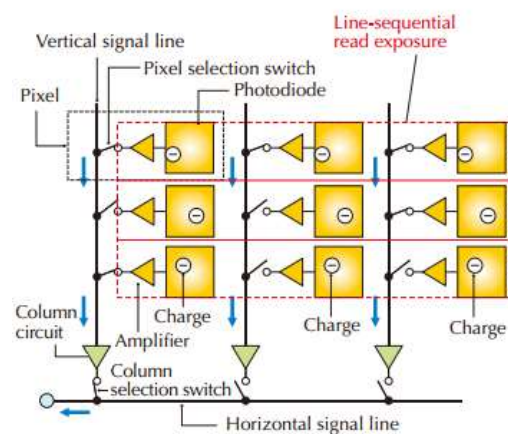


Figure 3: CMOS Sensor configuration [1]

3.2 Electrical Specifications

In order for our scanner to be functional, it will require to be connected to a computer through a USB cable. This will allow all the electronic components, the microcontroller, RFID reader, LEDs and CCD sensors within the unit to fulfill all their necessary requirements such as having a voltage tolerance boundary between 3V to 5V and a current range of 500 mA to 900mA. We chose to integrate all the units into one in order to simplify its usability by not having to worry about meeting each individual component's standard requirements since everything will run off one single input. Also by assembling everything in one piece, we were able to minimize the voltage input therefore increasing the efficiency of the unit. By connecting the device through USB to the computer and as long as the computer is connected to the internet, data storing can occur. This is a crucial part in our system since it constantly needs to access the database through an ethernet cable in order to upload it with the proper files received from the system outputs.

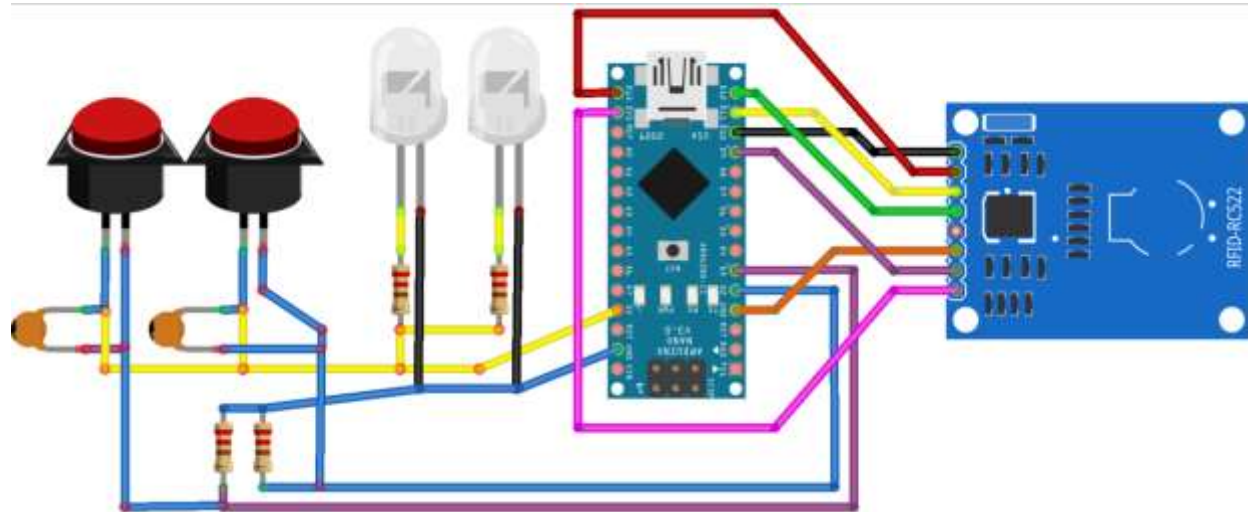


Figure 2: Wiring schematic of the circuit

3.3 Mechanical Specifications

The housing for this unit will be made of an ABS thermoplastic material via a 3D printer. The machine used was a fused filament fabrication device enabling layer-by-layer printing. The ABS filament is fed through an extruder, which is heated to 115° Celsius thus making the thermoplastic easy to mold.

ABS polymer will be used primarily for its thermoplastic properties. Thermoplastics means a polymer can be heated and molded to certain specification but it can later be remolded by applying additional heat. This allows a small room for mistakes. Unlike wood, ABS can be heated reshaped, but at much lower temperatures than metals. This will also minimize waste and create a sustainable product, and environmental awareness is very important to us.

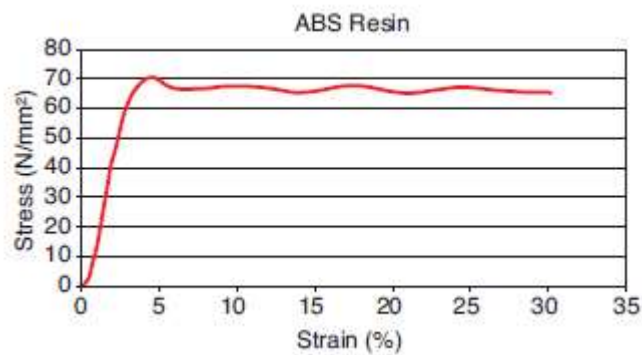


Figure 3: Stress vs Strain graph for ABS [2]

Figure 5 details the stress versus strain plot for the ABS polymer. Young's Modulus is the slope from the resting point to the yield strength and it is around 2×10^9 Newtons/meters². This large value in Young's Modulus, as seen from 0% to around 5% strain, means that the material is quite brittle and doesn't deform easily. Once past the yield strength, the material is said to be in the plastic region and it tends to elongate.

We opted to use a 3D printer because it is the easiest tool for rapid prototyping. It is a versatile tool that will print almost anything, which makes customization very simple. Compared to other types of prototyping an enclosed unit, 3D printer is relatively fast and a design can be fully printed in less than a day, usually in a few hours. Taking this and its cost effectiveness into consideration, 3D printing was the best decision for rapid prototyping.

A 3D design of the unit was created using SketchUp and later converted to a stereolithography file format as per the printer's format. The machine recognizes this format and proceeds with the printing.

3.4 Software Specifications

3.4.1 General Software

Produced software must be compatible with most recent versions of MS Windows, OSX and Linux. These 3 operating systems are backward compatible for binary and executable files. For compatibility with different versions of each OS, we consider using standard C++ and Python libraries, defining specific Macros/function for each version in header files and also make use of preprocessor directives. Identifying what is generic or specific to certain platform is achieved through experience, search and testing on different platforms.

C++ follows “Write Once Compile Anywhere” philosophy which we can recompile for different platforms using cross platform libraries and functions when needed. Also final software will be released as a complete software package, including open source software and libraries, to make installation package self-contained and installation process easier for users.

Python used for rapidly prototyping, Matlab for validation of image processing and production code mainly uses C/C++ to take advantage of fast interaction with hardware and rich library support for image processing.

To process image collected from scanner, OpenCV and Tesseract libraries are used for their good performance, compatibility with major platforms, C++ machine learning algorithms and real-time vision applications.[4]

Table 1: Tools and Effect of software

Tool	Effect
Python	Rapid Prototyping
Matlab	Validation of Image Processing
C/C++	Production code
OpenCV and Tesseract	Real Time Vision & OCR Tool
PHP, HTML	Web Elements
MySQL	Database System

Requirements for Resource management:

1. Functions and modules will be tested for any deadlock condition.
2. Hardware resources will be released after application has completed and when user closes the application to return allocated memory and serial/USB ports.

Requirement to prevent software crash:

1. Include watchdogs to avoid using resources for longer period of time and prevent stalling the software in unexpected condition
2. Exclude conditions lead to divide by zero
3. Check access memory. Avoid access to address space outside allocated memory to process/threads created by software which force OS to abort the application.
4. Errors will be captured to provide feedback to the user or resolve within the application if possible, e.g. reading a file without proper permission or create a file that does not exist.

Web interface Requirement:

1. Developed using mainly HTML 5.0 and PHP.
2. ID/barcode read by scanner or from RFID tag, will be stored in Excel, text file or on a server.
3. Prototype phase of project uses SFU servers and database system to store and retrieve information about students.
4. All student's data bounded to their student ID.
5. Following a RFID or scanner read, administrator have following options:
 - a. Add a new student to a database,
 - b. Save in an excel sheet,
 - c. Update the existing student account
 - d. Grant permission to acquire a toolkit
 - e. Take attendance.

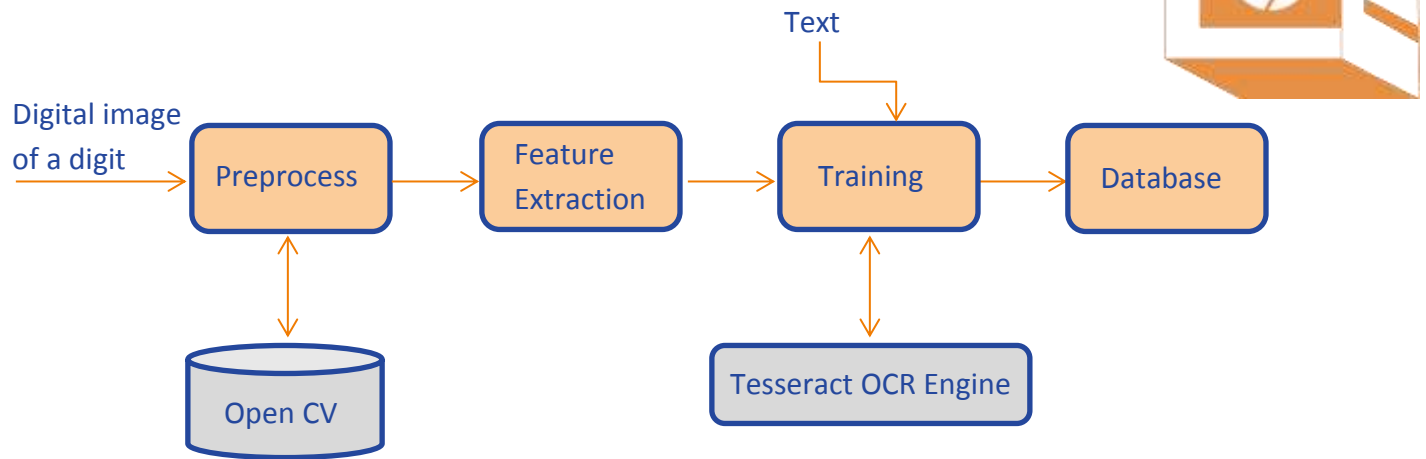


Figure 4: OCR system Training phase

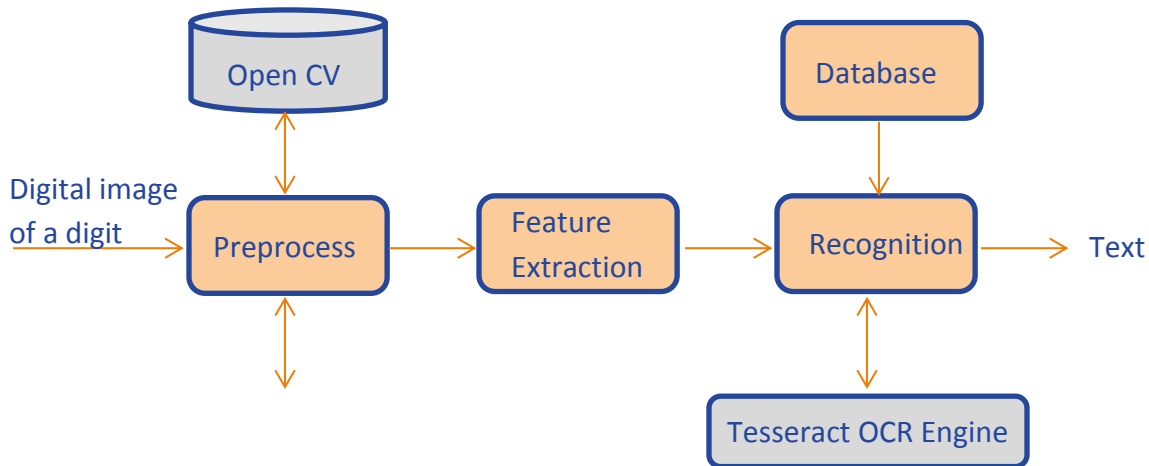


Figure 5: OCR system Tracking phase

3.4.2 Optical Character Recognition

The objective of an Optical Character Recognition (OCR) engine is to detect the region of number characters (the SFU ID) and recognize those number characters. Depending on the ambient light, the quality of image will be different depending on lighting of the environment in which the scan is taking place. As a result, the OCR engine may malfunction if the quality of image is poor. To resolve the issue with ambient light, we decided to develop our system in a closed loop system where the lighting and quality of the image is under our control.

Identifying the characters consists of multiple stages. Once the image is captured using a camera, the software will analyze the region where contains the characters. To do this, we find the coordinates for the corners of the image, Figure 8 is a sample image where the picture

contains both ID and barcode value. The ID number consists of 9 digits and barcode values consist of 14 digits. The goal of this software is to read both these number and give an output as a string of 23 digits. Once the coordinates of the image is detected, now it's time to break down the image into two regions. Since the ID and barcode are being read independently and for simplicity, we separate their regions. Figure 9 demonstrates the separation of barcode and ID.



Figure 6: Sample image



Figure 7: Getting the edges in the image

Once we have the regions separated, we operate the morphology. By performing morphology, we will be able to analyze and process the geometrical structures, which is based on set and lattice theory. In other words, the edge of the image will dilate and erode. Erosion and dilation work by analyzing the input image and examining the translation of the structure of the elements. In order to calculate the dilation of the image, we look at its binary inputs. For each pixel in the image, we would have a 0 or 1. This will help us separate the texts that we want to detect from its background.

If the shape of the image differs from the input image, the coordinate of its origin will most likely be different. To overcome this challenge, we decided to have a fixed shape of the image every time. In other words, we created a fixed box, with fixed lighting, in order to replicate the same type of image for every scenario. Although the surface of cards might be different and cause inconsistency, but the coordinates of the image would still remain the same. If E is Euclidean space (integer grid) and let A be a binary image in E , in order to get the erosion of A , using B as our structural element, we can use the following expression:

$$A \ominus B = \bigcap_{b \in B} A_{-b}$$

The dilation of A by B is given by:

$$A \oplus B = \bigcup_{b \in B} A_b$$

Next, the software will get the contours of the morphology picture, as was shown in Figure 9. This will help us to fully capture the region of the card number and barcode number by contours.

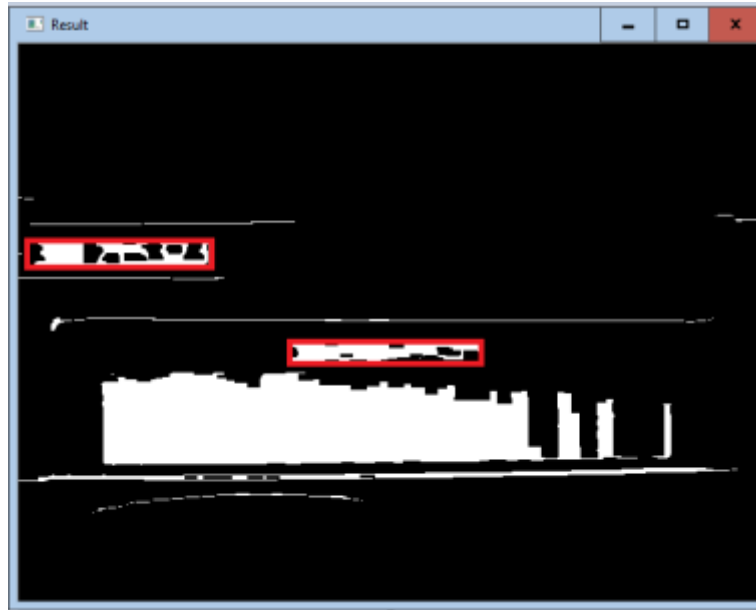


Figure 8: Regions of ID and barcode, captures by contours

At this stage we start training each characters, since the font and size is going to be consistent in all the cards, if we train all the digits, we will be able to use comparison for character recognition. In our case, we have two sets of numbers; a range of 1-9 for ID and a different set of 1-9 for barcode. Once we have these pictures trained in our system, we can cross reference the binary image and compare the 0s and 1s.

Once all digits are trained, each character (in this case digits) is taken individually and we process them individually, we have to keep repeating this process over and over again until we get the most accurate result. Figure 11 shows an example of where the value is detected.

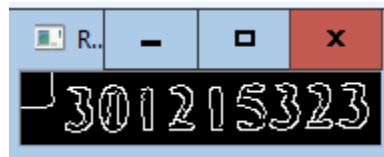


Figure 9: Getting the digital character in the region

3.4 RFID Specifications

MFRC522 RFID reader/writer is used for reading RFID tags. Internal transmitter of MFRC522 is able to communicate read tag without any additional circuitry and demodulation, decoding and error detection during read and writes are handled internally by the module itself. Table 2 shows required values for this project.

Table 2: RFID Module Specifications [3]

MFRC522 Property	Value
Frequency	13.56 MHz
Max Transfer Speed	848 kD
Card Type	ISO/IEC 14443 A/MIFARE
Communication Protocol	I2C

Arduino Nano 3.0 board containing an Atmel 328 microcontroller is chosen for this project because it is cheap, easy to use and have a big community of users. It is also compatible with many modules available in the market at a good price and performance. It connects to a PC using a micro USB cable which complies with our user friendly requirements for this project as well. I2C bus is used for communication between MFRC522 and Arduino board due to its simplicity. Pin map for both devices are shown in Figure 12 and 13. [6]

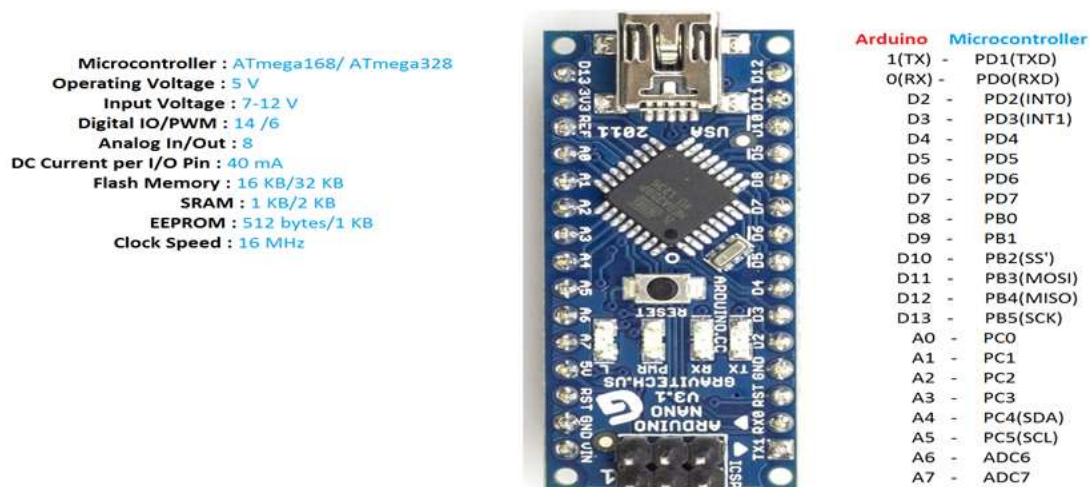


Figure 10: Arduino Nano Microcontroller

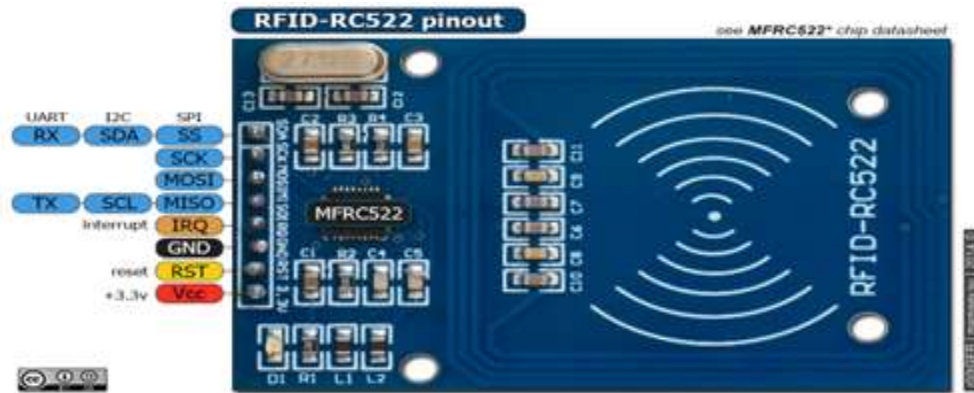


Figure 11: 13.56 MHz Radio Frequency Reader

Arduino IDE is used to program the Atmel microcontroller and MFRC522 and C library are used in this stage to receive ID from MFRC522 and pass it to the host computer using a serial port. On the host computer, another module (written in C++), finds the serial port connected to Arduino, continuously checks for new data arrival and checks if it is a numeric ID of specific length (ASCII character encoding).

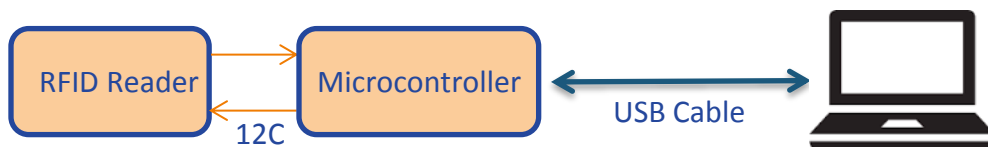


Figure 12: RFID, Microcontroller, PC Relationship

Finally ID is sent to foreground text editor chosen by the user or clipboard. Program running on the host computer, uses system functions provided by operating system to check for serial port events and communicate with other applications such as Excel. Both Arduino board and MFRC522 have small size and low price and help the team to meet size and price constraints of prototype and final product. Arduino board is powered through USB cable with an input of maximum 5V and output voltage from Arduino has been set to 3.5V (using Arduino pinMode function) which meets MFRC522 input voltage, both comply with our functional requirement.

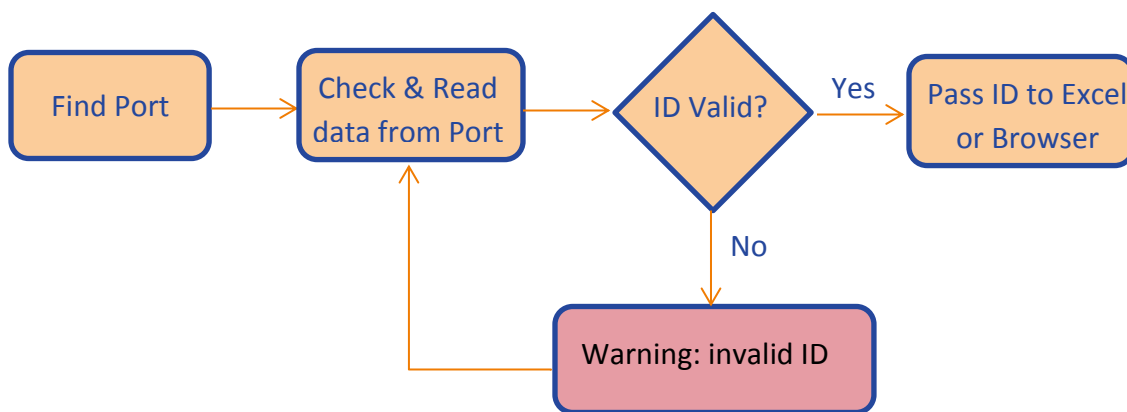


Figure 13: Validity block diagram

4. Testing

We have split up testing in four main sections which are unit tests, integration testing, system testing, and user acceptance testing. Table 3 outlines the time spent on each test and Figure 15 shows an overview of the testing timeline in the form of a Gantt chart.

Table 3: Testing Timeline

Task Name	Start Date	End Date	Duration
Hardware Test	10/18/15	10/20/15	3d
RFID/Scanner HW Test	10/18/15	10/19/15	2d
Mechanical/Electrical Test	10/19/15	10/20/15	2d
SW Input Test	10/20/15	10/27/15	6d
Input Capture Unit Test	10/20/15	10/20/15	1d
Input OCR Integration Test	10/23/15	10/23/15	1d
Input Serial Host Module	10/26/15	10/27/15	2d
Data Manipulation Test	10/30/15	11/03/15	3d
OCR Validation Test	10/30/15	11/03/15	2d
RFID to Text Field Test	11/02/15	11/03/15	1d
Web Interface Test	11/05/	11/16/15	8d
Local Server Test	11/05/15	11/06/15	2d
Remote Server Test	11/09/15	11/10/15	2d
Interface/Stress Test	11/16/15	11/16/15	1d
System Testing	12/02/15	12/07/15	4d
User Acceptance Test	12/02/15	12/07/15	4d

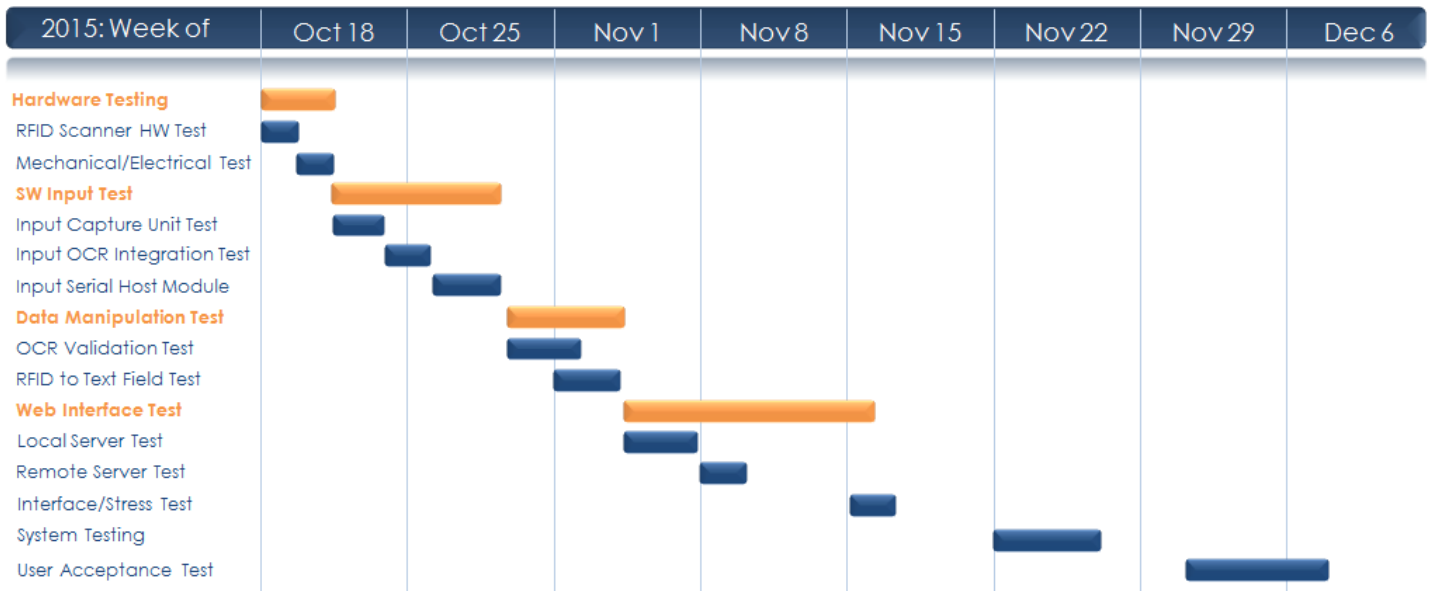


Figure 14: Gantt Chart showing testing milestones

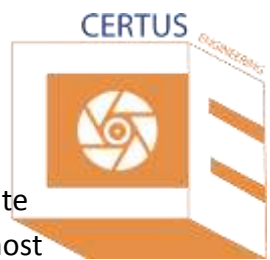
4.1 Unit Tests:

Software/hardware test cases are planned and performed to ensure correctness of each module in the system. Test cases will mostly check functionality of each module in boundary conditions and validation of our functional requirements.

These tests are performed earlier in the production phase or before adding any new and small software/hardware components to the project. Resistance of the box containing RFID/Scanner to drops, pressure and heat is tested. Each function or class created by our team is tested with variety of inputs to test critical situations such as stack overflow.

4.2 Integration Testing:

We will combine multiple unit tests to generate integration tests and observe how related components work together. As we add more components in later versions, these will be tested for proper integration. Integration tests are done after individual unit tests.



These tests are performed frequently in development stage to fix bugs or validate communication between components. For example, communication between host computer and microcontroller when scanning or tapping a RFID tag is tested to insure arrival of data in a short time and correct format. Electrostatic charge collected on the box is also measured when RFID and scanner both are used for a long period of time.

4.3 System Testing:

In this test phase, we focus on testing functionality of system as a whole. We will perform stress tests in this phase to ensure system can function properly under expected heavy use also to validate our functional and non-functional requirements. These tests are performed later in production stage when individual units are completed and final product or prototype is available for testing. Data propagation through system is observed and total latency, efficiency and correctness of the whole system is analyzed in this stage.

4.4 User Acceptance Testing:

End user tests will be performed to validate some of non-functional features of the system such as learnability, power efficiency and speed. Also customer satisfaction will be evaluated during this process. Dates of the tests will be between Nov 20th 2015, and Nov 30th 2015.

These tests are mostly done towards the end of production of final product or prototype however user feedback is required after completion of any front end unit. Ease of use, user satisfaction and learnability of final product will be analyzed using these tests and user feedback.

5. Conclusion

These design specifications will serve as a basis for Certus Engineering's execution of the Arca product. Arca is currently in development phase by Certus engineers who are committed to provide a reliable, functional and safe solution to human recognition and security problems. This document was meant to simplify the larger more daunting tasks into smaller subsections ultimately making it easier to analyze and execute them.

6. References

- [1] 2015. [Online]. Available https://pro.sony.com/bbsccms/assets/files/cat/camsec/solutions/E_CMOS_Sensor_WP_110427.pdf, 2015. [Accessed: 09- Nov- 2015].
- [2] 2015. [Online]. Available: <http://www.shimadzu.com/an/industry/petrochemicalchemical/n9j25k00000dmyx5.html>, 2015. [Accessed: 09- Nov- 2015].
- [3] 2015. [Online]. Available: <http://www.ti.com/lit/ds/slos758e/slos758e.pdf>, 2015. [Accessed: 09- Nov- 2015].
- [5] C. Loughnane, 2015. [Online]. Available: <http://www.pdnotebook.com/2011/07/whento-use-nonlinear-finite-element-analysis/>, 2015. [Accessed: 09- Nov- 2015].
- [6] E. Aziz and C. Chassapis, 'Probabilistic Simulation Approach to Evaluate the Tooth-Root Strength of Spur Gears with FEM-Based Verification', Engineering, vol. 03, no. 12, pp. 1137-1148, 2011. [Accessed: 05- Nov- 2015]