Biometric Fingerprint Recognition-based system for Time and Attendance Recording

by

Ong Koon Seng

13713

Dissertation submitted in partial fulfilment of

the requirement for the

Bachelor of Engineering (Hons)

(Electrical and Electronic)

AUGUST 2014

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Biometric Fingerprint Recognition-based system for Time and Attendance Recording

by

Ong Koon Seng

13713

A project dissertation submitted to the

Electrical and Electronic Engineering Programme

Universiti Teknologi PETRONAS

in partial fulfilment of the requirement for the

BACHELOR OF ENGINEERING (Hons)

(ELECTRICAL AND ELECTRONIC)

Approved by,

____

(Dr. Mohd Zuki Yusoff)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

August 2014

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

ONG KOON SENG

ABSTRACT

This projects aims to design a fingerprint biometric system for the usage of time and attendance recording at a university level or a small organization and also create a prototype of the designed fingerprint biometric system. This system aims to solve the problem of the current method of taking attendance in university classes which by attendance sheet or paper. This method has been proven to be ineffective since it can be easily manipulated by student and the storage of data is troublesome. This project aims to solve these issues. The system is based on Arduino board which will communicate with the SM630 Fingerprint Reader Integrated module to obtain the data of the user and also record the time by using a SD card reader to store the data. Based on the all the testing done all the hardware has be integrated into one complete system. The fingerprint module testing show that the accuracy of the SM630 Fingerprint Reader Integrated is very high in comparing fingerprints for one another and the prototype is manage to record all the data of the attendance of the sample. The prototype of the system have meet with the basic requirement of the project and can be used in a practical situation.

ACKNOWLEDGEMENTS

I would like to say my thanks to my supervisor Dr. Mohd Zuki Yusoff for all his guidance and valuable input to help complete my final year project and make it possible.

I also would like to thank the FYP committee for their support and knowledge to assists the students. I would also like to thank to all the lab technicians that take their time to help to necessary input and guidance to complete the project

Lastly I would like to thank all my friends and family who are supportive enough to help complete this project and make it a success. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

PROJECT BACKGROUND

1.1 Background Study

The method of identifying an individual using fingerprint has been used since the late nineteenth century. Now most of the fingerprint identification method are automated and have been digitalize due to the increase of need for an integrated identification system within all law enforcement and security protocol. However this method of identification is currently is not used in the field of education or small organization. Therefore, the purpose of this study is to implement a practical fingerprint identification method through embedded system which can record the time and attendance of an individual in a database which can be used any organization or an institution.

1.2 Problem Statement

The conventional method of recording a student attendance is by paper which is simple but it has its own problems. This method itself has been proven to be easily manipulated by the student simple by asking their partner or friend to sign for them even though they are not there. This method is also prone to the loss of data since the attendance paper can be misplaced or missing before the attendance can be store digitally. Therefore there is a need for a better method to take attendance of students and save the record digitally.

Using fingerprint identification one can reduced the chances of the attendance data to be tampered by student or any external factors significantly.

1.3 Objectives

The objectives of the project are:

- Design an embedded system using biometric fingerprint recognition-based system for time and attendance

- Build a prototype of the embedded system implementing the biometric fingerprint system for time and attendance which can fulfil the following requirement which is able to take and attendance accordingly, save the recorded data and easy to be used by student and lecturer

1.4 Scope of Study

The project starts of by studying and revising the implementation of the fingerprint reader and the usage of microcontroller to implement the biometric fingerprint reader this include reading through published paper, articles websites and books. This is to gain ideas on how to design an embedded system of the prototype.

Then following part of the project involves in designing the embedded system of the prototype. Design of the hardware would include on the necessary functions needed for the prototype, how the microcontroller communicates with the fingerprint reader module, the hardware needed for the prototype and associated circuitry.

Once the design of the prototype is done, there is a need to build the prototype which can be done in the lab. After the prototype is done there will be need to program the hardware and test the prototype to see will it fully function according to the criteria given. Testing of the prototype will be done until the prototype meets the minimum requirement of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Fingerprint Identification

The method of identifying oneself through one's fingerprint has been used for more than a century by many different civilization[1]. In this modern era, the method of fingerprint identification are widely used in the fields of law enforcement, information security and security devices.

The basic principle of fingerprint identification lies in the ridges and valleys of the fingerprint. There are distinctive pattern of this ridges and valleys and there are also micro-pattern which are called minutiae in one's fingerprint [1, 2]. These pattern has been categorized into a few pattern like for the overall pattern of the fingerprint there are left loop, right loop, whorl, tented arch and arch. Meanwhile for minutia features are ending, bifurcation, lip-rounding, double bifurcations and bridge.

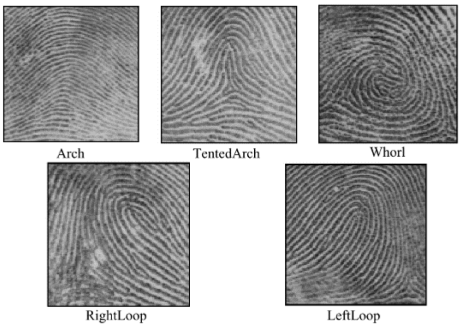


Figure 1: Categories of Overall Pattern of a Fingerprint [1]

In the process of fingerprint identification there are a few process that need to be follow in a certain order which is the order given: fingerprint image capturing, pre-processing, feature extraction and feature matching[2].

The method to identify one's fingerprint is basically lies in two general method which is minutiae matching or overall pattern matching [1, 3]. In both of these method there are some drawback like in the minutiae matching it takes time in pre-processing and recognise the features while in the overall matching it takes a lot of data compare to the other method[3]. Usually in a smaller system just uses the minutiae method for identification[2].

2.2 Biometric System

Biometric system is a system that uses a person physical features to identify a person which is exclusive to that person [4]. In this system it will capture a person biometric data and compare it within the database [4]. In a biometric system there three important factors that need to be taken account for which is the person using the system, the sensor and the features of the biometric data[5]. Usually in a biometric system there are two modes which is verification mode and identification mode [4, 6]. In verification mode, the system will compare the person feature with the person own features database. This database could be a smart card or Personal Identification Number [7]. Meanwhile in identification mode the system will compare a person biometric data with it the system database. The algorithm in the system for both mode will do the rejection or acceptance [8].

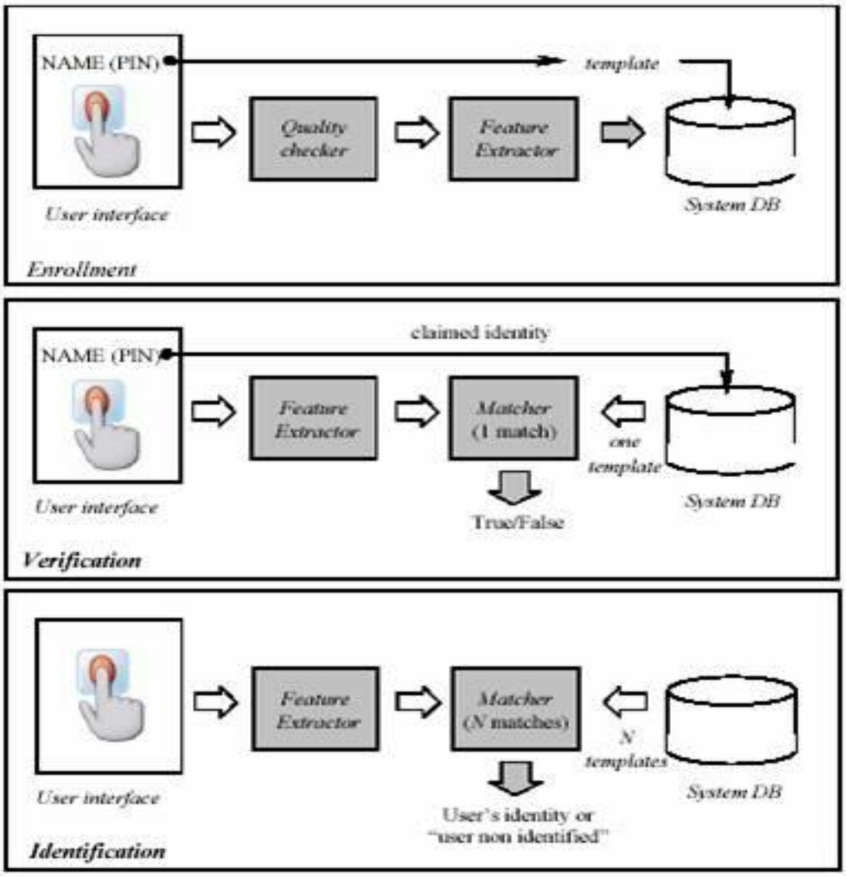


Figure 2: Block Diagram Enrollment, Verification, and Identification of a Biometric System [4]

In both identification mode and verification mode there are a few stages for a biometric system to match and enrol an individual into the system which can be seen in the Figure 3 below [9]:

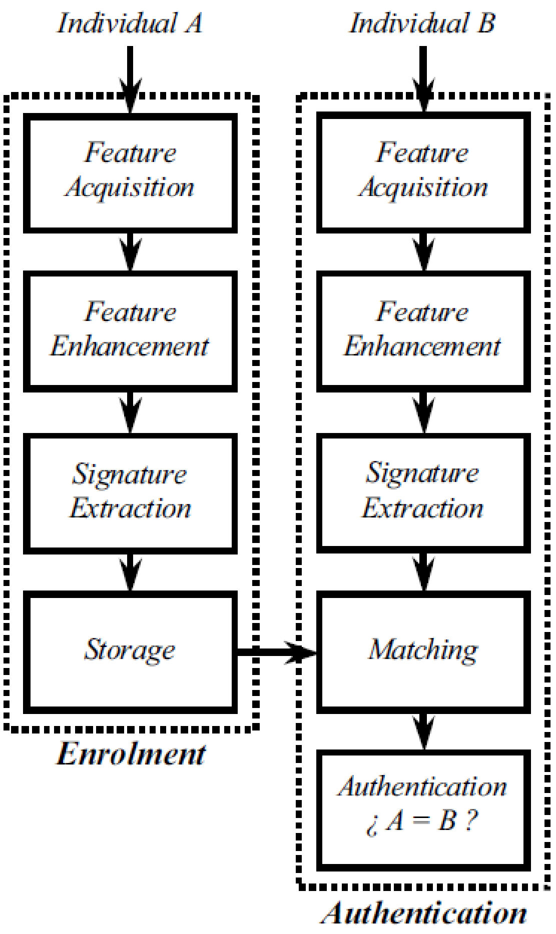


Figure 3: The stages of biometric system to Enrollment and Authentication [9].

As from the Figure 3 there are similarity in Figure 2 where the only extra detail of the diagram is the feature enhancement where the feature extracted is enhance to get a better quality biometric data. These stages are more one the firmware design of the biometric system rather than the overall view of the biometric system in Figure 2. Other than that in Figure 3 in the Enrollment stages there is a connection to the matching stage and the storage stage. This check the biometric data taken is matching to any of the previous biometric data.

As mentioned before an important factor of a biometric fingerprint is the biometric data itself. A biometric data of the system should meet this four characteristics to be usable in a biometric system which is universality, distinctiveness, permanence and collectability [10]. There are a few biometric data that fit the characteristics mentioned and one of them is fingerprints of a person. For example everyone has fingerprints which fit the universality characteristic. Meanwhile for distinctiveness a person fingerprint differ for one to another. Another than that fingerprint does not change a lot over a period of time which fits the characteristics of permanence and lastly a fingerprint is collectability because it can be measured.

In a biometric system there are four main component that make the whole system complete [4, 5]. These components are the sensor, feature extraction, feature matching and system database. As the name the suggest the sensor is the hardware used to capture the biometric data, feature extraction is captures the distinctive feature of the biometric data, feature matching compare the features with the one in the database and the system database is the one that store any capture biometric data in the system. Therefore to make a complete biometric system these four components are needed.

In a biometric system there is a need to have an error tolerance in comparing the biometric data. This error tolerance in a biometric system is called a matching score that actually calculates the similarity between the two biometric data that have been obtained [4]. The higher the score the higher the probability of the two biometric data belong to the same person. There are times when the biometric system give false acceptance but this usually depends on the initial quality of the biometric template that have been obtained. Therefore it is important for a biometric system to have a good sensor to obtained quality biometric templates. This is error usually happen in a identification system. Meanwhile in a verification system there are two type of error that can be produced which is false match where the system mistaken two different persons to be the same person and false non-match where the two biometric measurement from the same persons is mistaken to be two different person [10].

For a biometric system usually one concern one type of biometric of an individual. There are several type of biometric which are DNA, ear, face recognition, fingerprint, gait, hand and finger geometry, iris, retinal scanning, signature verification and voice recognition [6]. Even though there a lot of different biometric that can be obtained from a person the fingerprint identification is the most common and reliable comparing to the rest which can be slightly impractical to be used as commercialize system.

2.3 Biometric Fingerprint Sensor

In a biometric system there is always a need for a sensor to extract the biometric data of a person. In a fingerprint sensor mostly uses live-scan technology. During the writing of this paper there are basically three main family of the live-scan technology which is solid-state, optical and ultrasound [11, 12]. The performance of the biometric system is affected by the quality of the biometric sample which makes the efficiency of the sensor plays a huge factor in the overall result of the system [13].

If a sensor does not work accordingly or accept a fake fingerprint template from the user it will compromise the overall recognition system[14]. This is common when one tries to circumvent the biometric system.

There are many variation fingerprint sensor in the market each with its own advantage over the others[11]. The effectiveness of each sensor varies on the situation or system to be used on. There are also a few factors that can affect the fingerprint sensor which are changes in the environment, usage of the sensor, a change in the biometric data which is temporal and changes in the sensor itself [15].

Therefore there is a standard on assessing the quality of a biometric templates that is based on three different point of views which are character, fidelity, and utility [14, 16]. From these views one can determine efficiency of a sensor in a system.

2.4 Finite State Machine Programming

There are many type of programming languages for embedded systems like C, C++, Assembly, VHDL and Verilog. The usage of these languages depend on the microcontroller or microprocessor used in the system. Although there are different but most of programming languages can be divided into two basic paradigms which is imperative and declarative [17]. Currently, imperative programming are more used compare to declarative programming since imperative programming support object-oriented features. In object-oriented programming, control flow can be done in sequential composition, branching, looping and subroutine [17]. Example of these can be seen in term of C programming where sequential is denote by semicolon, branching by **if** and **switch** statements, looping by **while, do-while** and **for** statement and subroutine by calling functions.

Therefore with this implementation of a finite state machine in the firmware of the embedded system can be used in C language programming. State machine is a description of control system which consists of a set of states, a set of transitions between states and a set of actions associated with these states or transitions [17]. Conventional implementation of state machine is by manual encoding of an abstract model such as the state transition diagram or different design pattern [18]. Other method of implementing state machine is provide design patterns for hierarchical and concurrent state machines [18]. In the Figure 4 below shows an example of an implementation of state machine. The letter b1, b2, b3, b4, b5 and b6 is a state while the arrows shows the transition each state can go to.

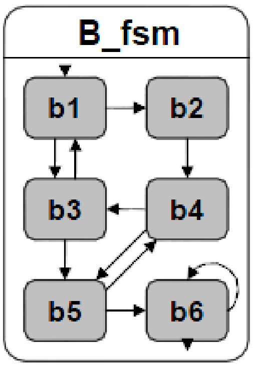


Figure 4: The Example of State Machine

CHAPTER 3

METHODOLOGY

3.1 Research Methodology

The flowchart below is the brief overview of the research methodology in this paper. The explanation of each step is located in the next sub-chapter which is the project activities.

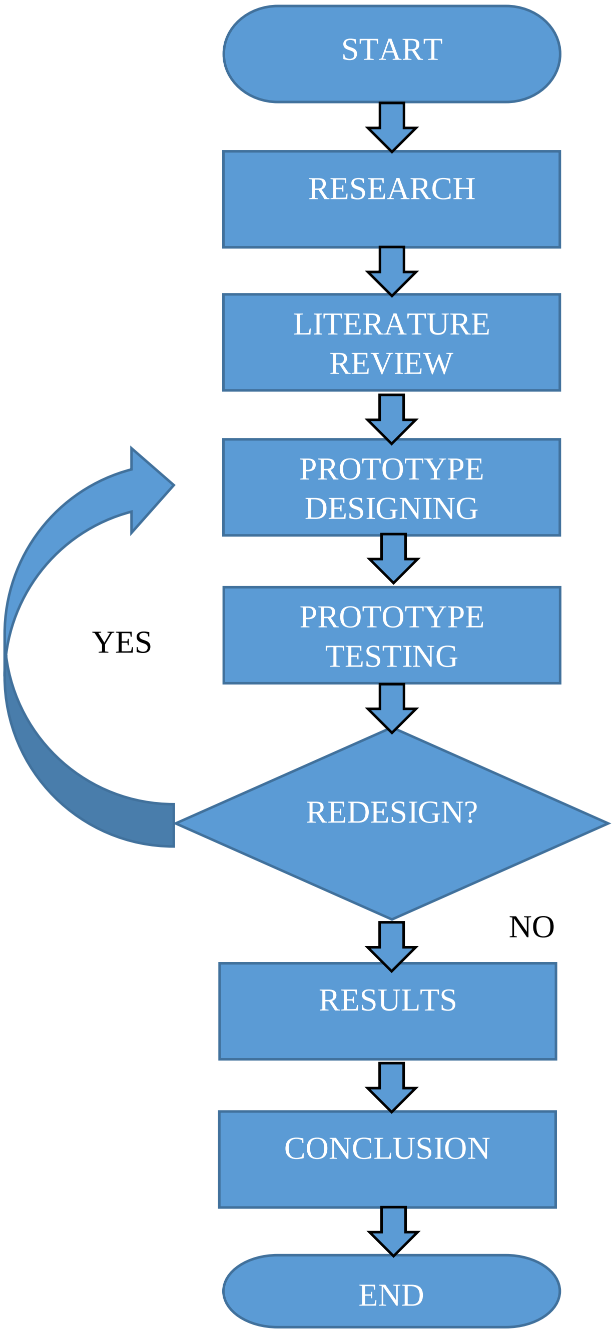
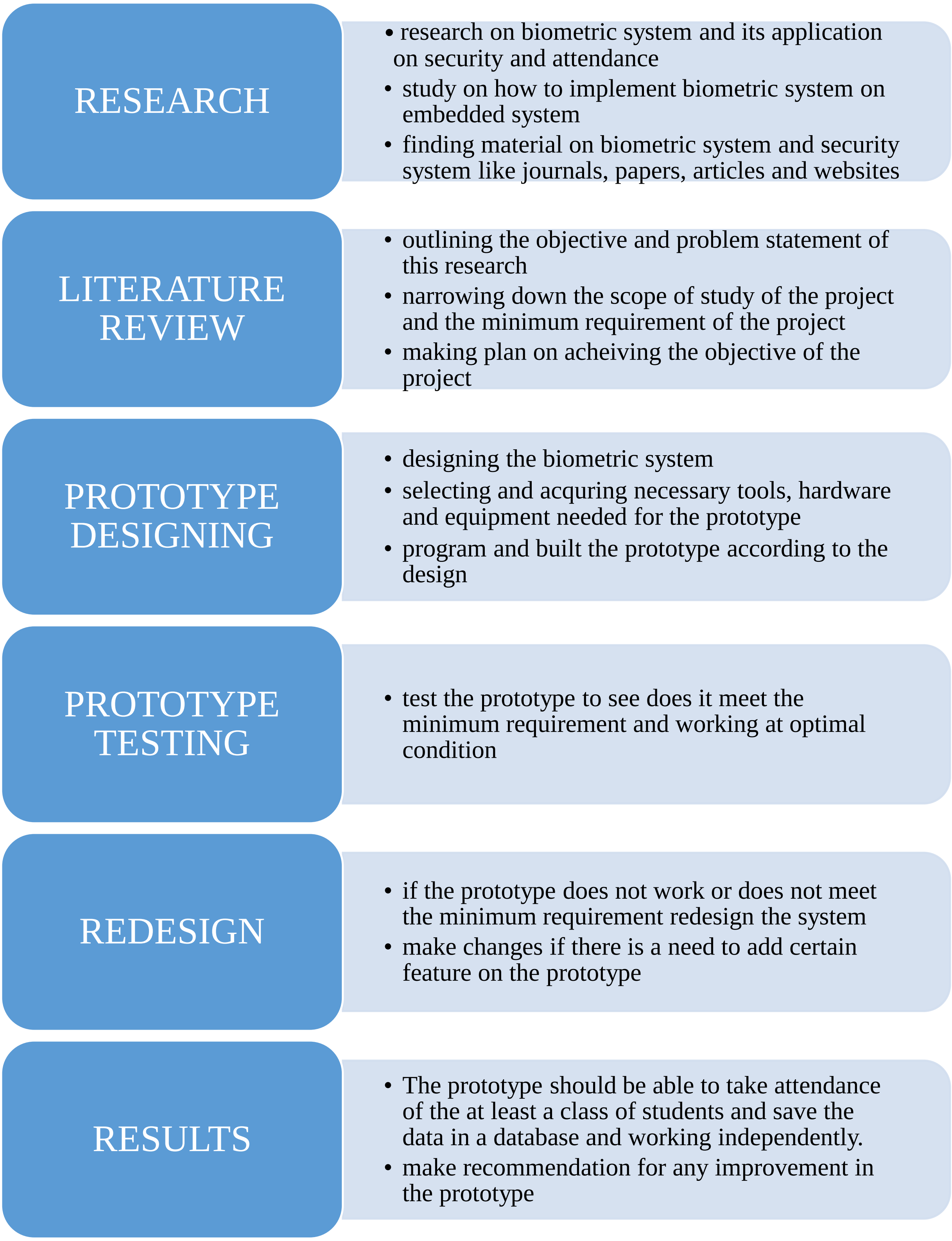


Figure 5: Research Methodology Flowchart

3.2 Project Activities

In this sub-chapter of the paper will be describing each step of the overview of the research methodology in the previous flowchart.



RESEARCH

- research on biometric system and its application on security and attendance
- study on how to implement biometric system on embedded system
- finding material on biometric system and security system like journals, papers, articles and websites

LITERATURE REVIEW

- outlining the objective and problem statement of this research
- narrowing down the scope of study of the project and the minimum requirement of the project
- making plan on acheiving the objective of the project

PROTOTYPE DESIGNING

- designing the biometric system
- selecting and acquring necessary tools, hardware and equipment needed for the prototype
- program and built the prototype according to the design

PROTOTYPE TESTING

- test the prototype to see does it meet the minimum requirement and working at optimal condition

REDESIGN

- if the prototype does not work or does not meet the minimum requirement redesign the system
- make changes if there is a need to add certain feature on the prototype

RESULTS

- The prototype should be able to take attendance of the at least a class of students and save the data in a database and working independently.
- make recommendation for any improvement in the prototype

Figure 6: Description of each step in Research Methodology

3.2.1 Prototype Design

Hardware design

After researching all the possible design of the prototype during the research and literature review stage of the project, the selection of the hardware have been made. The embedded system that will be used to create the base of the system is the Arduino embedded system. The Arduino board that will be used is the Arduino Mega 2560 as shown in the picture below.

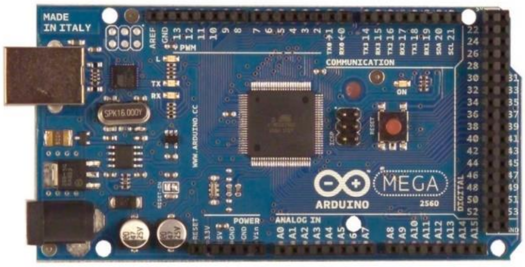


Figure 7: The Arduino Mega 2560

The main reason of the selection of this board is because of the number of I/O available on the board itself is sufficient for the usage of this system and also the firmware of the board is based on C programming style. Moreover the board can interface with many kind of third party modules. The schematics of the Arduino board is as follows which includes the overall connections of the ATmega 2560 microcontroller with the I/O ports and the necessary connections in Figure 6 and Figure 7.

Form the schematics most of the connection used are connected to the I/O ports which can be used for other modules or devices to communicate with the microcontroller of the Arduino board. There are also specific communication ports which are shown in the schematics which are the RS232 communication ports and ISP communication for the communication with the fingerprint reader module and SD card read module respectively.

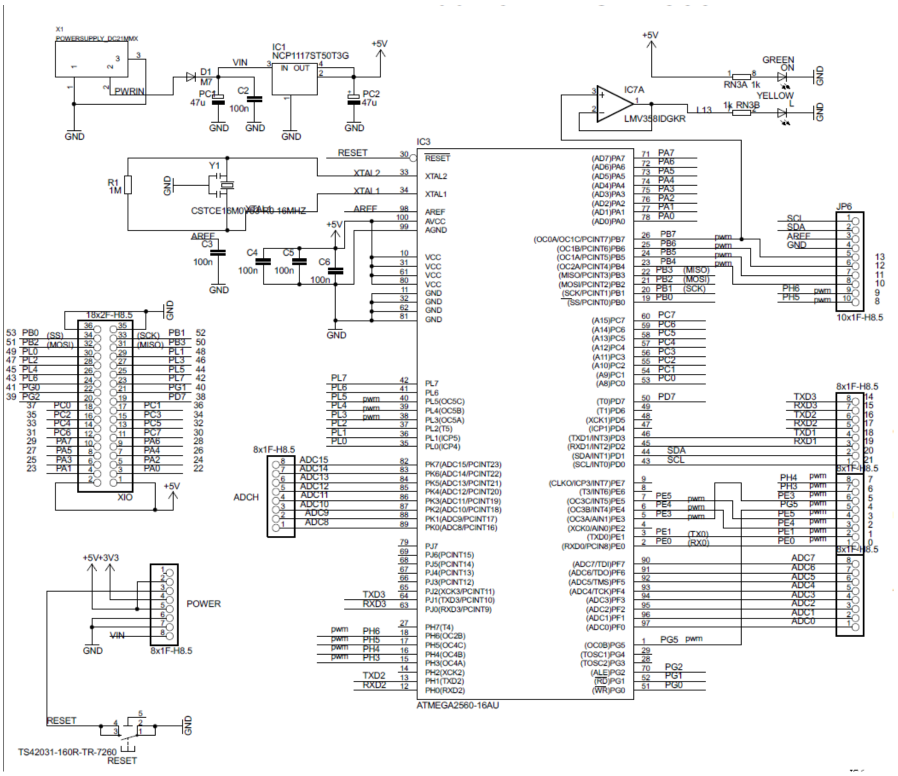


Figure 8: The schematics of the ATmega 2560 microcontroller with I/O ports and RS232 ports

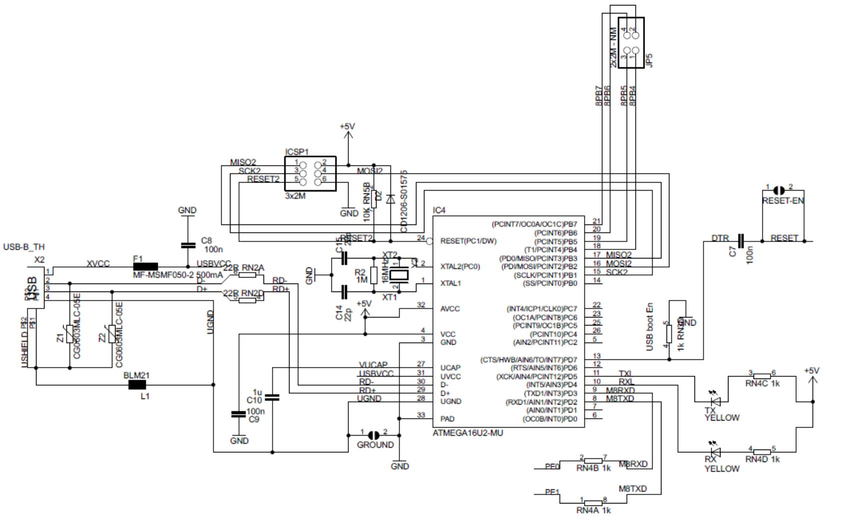


Figure 9: The schematics of the ISP communication with the ATmega 2560 microcontroller

The following hardware is the fingerprint sensor module. In this project, the Fingerprint Reader Integrated SM630 module is selected. This is because it is an optical fingerprint reader which is ideal to be used in this system due to the situation it will be used compare to a capacitive type of reader which degrade after multiple times of usage and need to be maintain constantly. Other than that the Fingerprint Reader Integrated SM630 has its own high performance DSP processor and Flash Memory to create and its own database to store up 768 fingerprint template in the module. This module also has many functions which include registering fingerprint, deletion of fingerprint, fingerprint search, fingerprint upload and download.

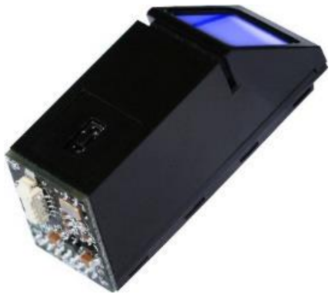


Figure 10: The SM360 Fingerprint Reader

The specification of the SM630 fingerprint Reader is as follows:

Table 1: Technical Specifications of SM630 Fingerprint Reader Integrated

Operating Voltage	4.3V – 6V
Rating Voltage	6.5V
Operating Current	< 80mA
Fingerprint Templates	768 Templates
Search Time	< 1.5s
Power-on Time	< 200ms
Tolerated Angle Offset	±45°
User Flash Memory	64KByte
Communication Baud Rate	57600bps

The next hardware is the data storage hardware which the Ethernet Shield by Cytron. The main reason this choice is because the Ethernet Shield has a SD card module and can be easily integrate with the Arduino Mega 2560 board. Therefore the data storage for the system is a SD card which can be plug out and in of the system easily. This SD card will also serve as a data storage to store the ID of student and be used to compare to the fingerprint of the student that has been registered. All the data will be stored in the SD card in .CSV file or Comma Separated Value file which can be accessed by Excel program.

The following part will be the display which will be a LCD display of 16x2, the time module which is the RTC module, the input of the system which will be the Keypad and also the power system of the system due insufficient power if using only the board output power.

The power system of the board will be using an external power source to power up the whole system. In normal usage the Arduino board should be able to perform normally but after connecting the Arduino board to multiple modules the power on the board is unable to support it causing the insufficient power to the module therefore creating problem for the module to run efficiently.

This power system is aim to use a 12V 2A power supply to power up the system which supposing to give out an output of 5V and 8A current to the system. Below is the schematics of the system.

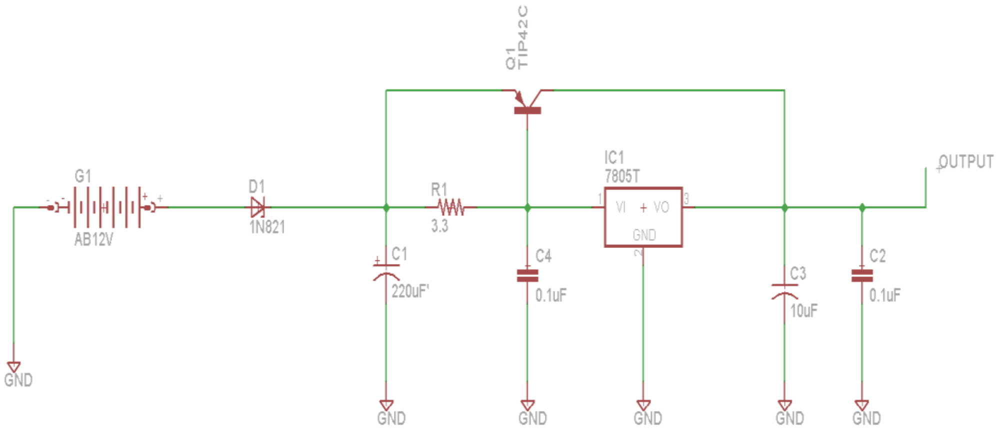


Figure 11: Power System for Arduino Board

Another hardware that is been used in the prototype is the keypad module. The keypad will be used to keying the ID of the student during registration of the student and also will be used to confirm the input keyed in the system is the right one. Other than that the keypad will also be used to keying the password of the system to delete the attendance data and also the fingerprint data of the system. The keypad module has a circuit to connect to the system which is connected to the 1KΩ resistors. The input pin and output pin of the keypad is connected to the analogue pins of the Arduino board so that during the press of the button the system can avoid noise signal. Below is the picture of the keypad module:

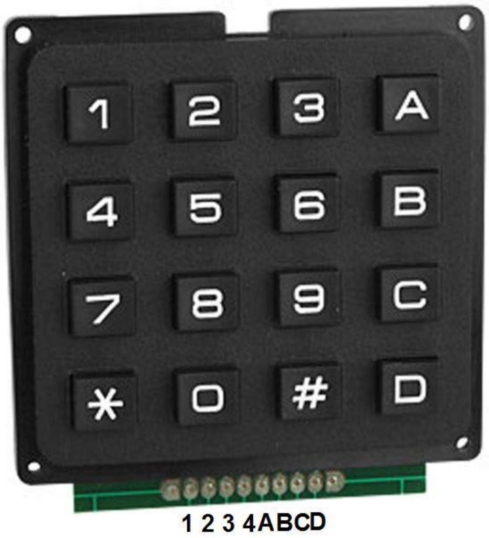


Figure 12: Keypad Module

From the hardware available the overall design should be around 9.8cm width, 17.7 cm long and with a height of 5.6cm this would be including the casing design of the prototype and also the keypad module attached to it. The dimension whole system should be around comparison size of a big tissue box which can be carried around to classes.

According to all the hardware specification hardware used in the prototype the power supply has to be around the rating of 12V 2A to properly power up the system without any module not working due to insufficient current or power like the fingerprint module.

Firmware design

The firmware design of this project is be based on the state machine style of programming. In a state machine style of programming there will be a variable to store the state of the program accordingly. At the beginning of the firmware the variable will store the state it is in currently. Once the conditions to change the state of firmware is achieved the variable will change to the next value or state. This will cause the firmware to go to the following state of the program. If the value of the state does not change the firmware will only loop in that state. This make the coding of the firmware to be more organized. The picture below show how the state machine works:

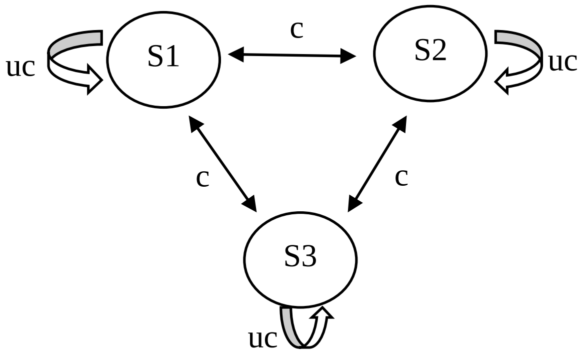


Figure 13: Representation of the State Machine

From the figure above is a simple representation of the state machine where S1, S2, and S3 is the state of the firmware. The letter 'c' is when the condition is met it will change state or a state transition. Meanwhile, 'uc' is a representation of when the condition is no met it will stay is the state.

In state machine style of programming also allows the firmware to enter different mode of the system not only repeat the whole firmware during the loop. For example, in this project there are a few mode needed which is the registration of the fingerprint, taking attendance and deleting the database. In state machine, the program can go to the part of the program like registration of the fingerprint without going through the part of the firmware for taking attendance and deleting the database.

Below is the overall state machine diagram of the system that has been implemented:

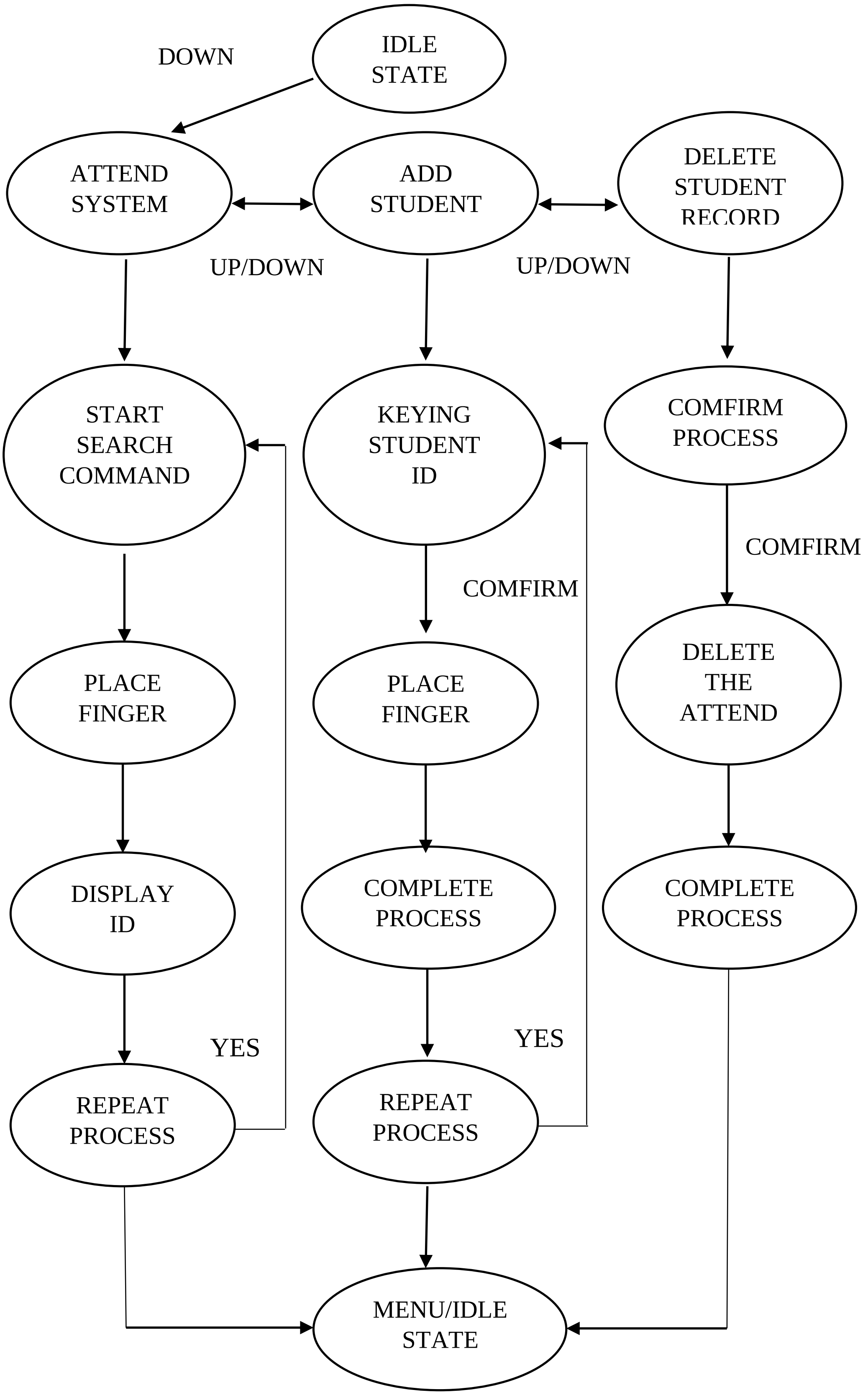


Figure 14: The State Machine Part 1

Other than that, there are other state machine in the system which could not place in the previous diagram will be placed in the diagram below:

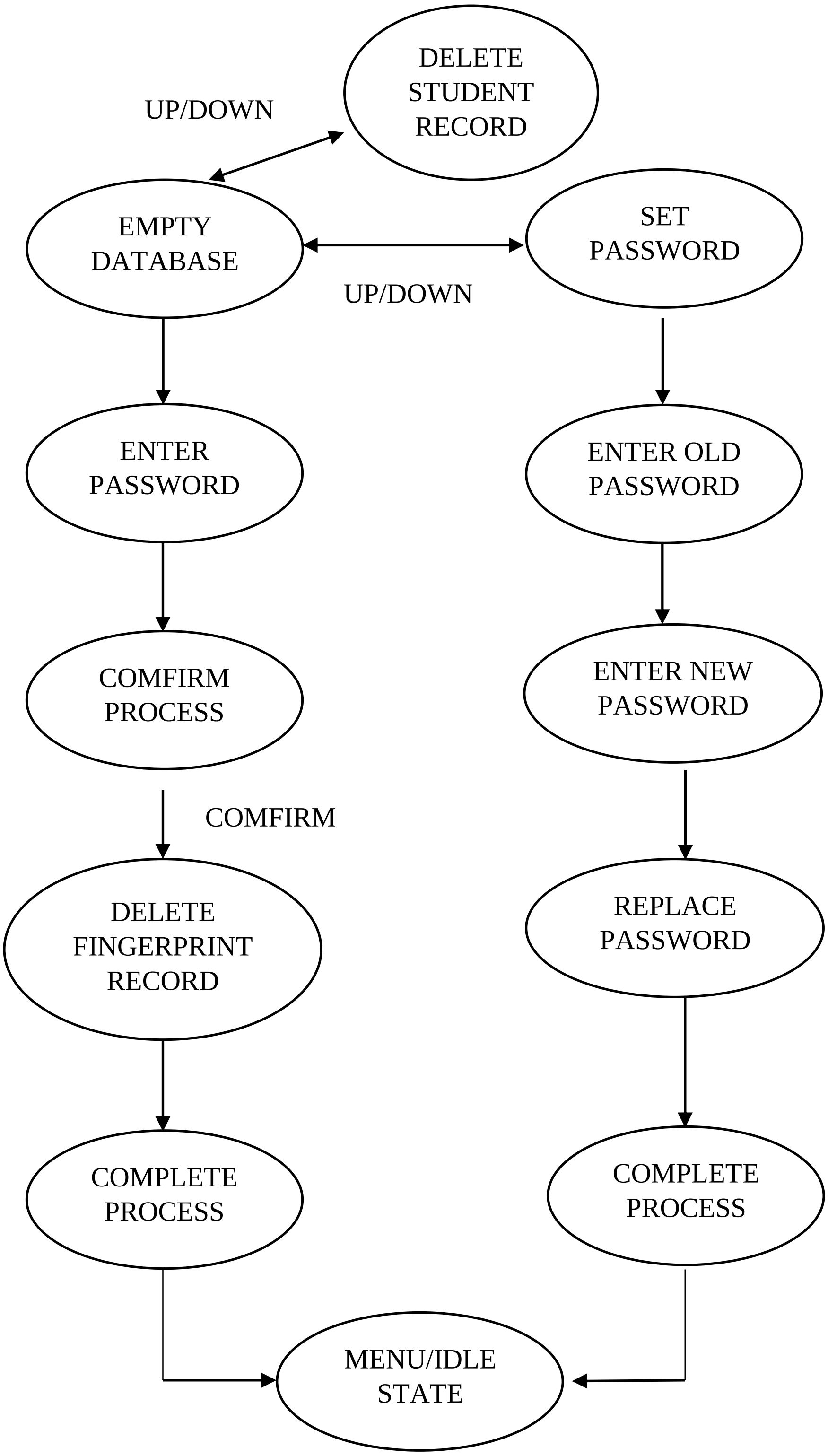


Figure 15: The State Machine Part 2

As we can see from the state machine there are actually to five mode which is the Attend System mode, Add Student mode, Delete Student Attendance mode, Empty Database mode and Set Password mode. All this mode can be accessed by using the UP/DOWN buttons of the LCD keypads and enter the mode in the system by using the SELECT button on the keypads also.

This diagram is the basic design for the whole firmware of the system structure. Most of the firmware will follow this diagram with some modification on the firmware due to complication and how the module of the system interact with the system. Overall the design of the system shown above should be working perfectly with the system. The state diagram shown is to give a general explanation of how the prototype works.

There are also state machine diagram on how the modules of the system interact with the system but that will be not be shown in this paper due to the fact that the modules interaction are usually based on protocol rather that design how the module works. The protocol of the module will be explain in the next part of the paper on the coding function of the system.

Coding Function

The SM360 Fingerprint Reader Function

In the firmware there will be a few function to run the whole system. One of the function is function to communicate between the system and the fingerprint reader module. This function will need the input of the command code and number of fingerprint to run. In this function it is have the command of registration of fingerprint, deleting of fingerprint, finding the fingerprint and also deleting the database of the fingerprint.

The communication protocol of the fingerprint module is by RS232 communication which they will send a packet of data to the fingerprint module and wait for it to respond. The packet of data send has a few parts depending on the command send this also goes the same for the data packet received from the module. In general the data packet of the module is as follows:

- Packet head – 2 bytes
- Packet flag – 1 byte
- Packet length – 1 byte
- Packet content – N bytes depending on the command
- Checksum – 1 byte

Therefore if we were to send a command of the packet head is set to 0x4D and 0x58. Meanwhile the packet flag is the type of data this data is sending or receiving. For this example the type of data will be the command type of data which will be 0x10. The packet length is the number of packet we will be sending in the packet content. For this we will take the add fingerprint command which have 3 packet length therefore the packet length will be 0x03. The next byte of data will be the packet content. As mention before this example is a add fingerprint command therefore there are 3 bytes to the packet content which the command code, the high byte of the position we are adding and the low byte of the position we are adding. If we are adding the fingerprint in position 4 of the template the high byte will 0x00 and the low byte will 0x04 this is because the high byte is the position divided by 256 and the low byte is the reminder of the division. And for the adding fingerprint command code it will be 0x40 as stated in the datasheet of the fingerprint reader module. Lastly is the checksum packet of data which is simply the addition of all the packet head, packet flag, packet length and packet content. Therefore in this example the checksum is 0xFC. Therefore the whole packet of data send is:

0x4D + 0x58 + 0x10 + 0x03 + 0x40 + 0x00 + 0x04 + 0xFC

As from the example above there is a pattern to all the command of the module. Therefore it is possible to create a function to simply the programming process of the system to send a command for the fingerprint module. This function will create the packet of data it will need to send for the type of command based on the input. Below is the base of the function not the full function.

```
void f_cmd(byte cmd, int number){
Serial.begin(57600); byte packet_content;
switch(cmd){
case SEARCH_FINGER: packet_content=5;break;
case EMPTY_DATABASE: packet_content=1;break;
default: packet_content=3;break;}
byte Hbyte = number/256; byte Lbyte = number%256; byte checksum = 0;
byte send_cmd[packet_content+5];
for(byte i=0; i<sizeof(send_cmd);i++) send_cmd[i]=0;
send_cmd[0] = 0x4D; send_cmd[1] = 0x58; send_cmd[2] = 0x10; send_cmd[3] = packet_content;
send_cmd[4] = cmd;
for(byte i=0;i<sizeof(send_cmd);i++){
checksum+=send_cmd[i];}
if(packet_content>=3){
send_cmd[5] = Hbyte; send_cmd[6] = Lbyte; checksum+=send_cmd[5]; checksum+=send_cmd[6];
 if(cmd==SEARCH_FINGER){
for(byte i=7; i>4; i--)
end_cmd[i+2]=send_cmd[i]; send_cmd[5] = 0; send_cmd[6] = 0; checksum+=send_cmd[5];
checksum+=send_cmd[6];}}
send_cmd[packet_content+5-1]=checksum;
Serial.write(send_cmd,sizeof(send_cmd)); delay(1); }
```

There is also another function for the system to receive the respond back from the module and produce a proper output according to the respond. The pattern of the data packet of the receive command is the same as the send command. In the receiving part of the communication data packet of the module will send at least two communication packet which is whether the command of the communication packet is received correct and the operation of the command is successful. From the communication packet we can determine the type of data send and determine the data we retrieve from the module. This function will check the data packet send by the fingerprint module and compare it to check if the respond is a failed or successfully operation.

The Ethernet/SD card Module

The next function that will be explained here is the SD card module in the Ethernet Shield. The SD card module is connected to the main board of the system through SPI (Serial Peripheral Interface) communication protocol which is connected the hardware connection of ICSP (In-Circuit Serial Programmer).

The communication SPI protocol there are four logic signals which are SCLK (Serial Clock), MOSI (Master Output, Slave Input), MISO (Master Input, Slave Output) and SS (Slave Select). This communication protocol communicate in either master/slave mode where the master will initiates the data frame of the communication. The master device can communicate with multiple slave devices by choosing which slave device to communicate through the SS signal pin. The SPI communication protocol is mainly handle by using the “SDFat.h” library which is available in open source.

By using the library mentioned before there are a few function that can be created to be used in the prototype to save the data. The function are Sd_check(), Write_storage(), Read_storage and Empty_data(). The first function which the Sd_check() function is a function to initialise the setting of the SPI communication and also check the present of the SD card on the module. If there is no SD card on the module the function will display an error on the LCD display. The coding of the function is as follows:

```
boolean Sd_check(){
if (!sd.begin(chipSelect, SPI_HALF_SPEED)) sd.initErrorHalt();
  boolean SdCard_status = card.init(!SPI_HALF_SPEED, chipSelect);
  if (!SdCard_status) {
    msg("SD Error");}}
```

From the function above, the first line of the function it is to initialise the SD card module. From the line there is a chipSelect variable which is the ICSP pin for SS logic signal for this prototype the value of the chipSelect is the pin 53 in Arduino Mega 2560. The second line of the state is to get the SD card status in the module. This will get a value which can be used to compare.

The next function is the Write_storage() function which will write the data of the student we want to save in a .CSV file. As mentioned before the file can be open by using Excel program. The basic structure of the write function in the firmware is as shown below:

```
void Write_storage( int ID, int Storage){
  ofstream sdout(fileName1, ios::out | ios::app);
    sdout << ID << pstr(",")<< Storage << pstr(",")<< pstr("\n")<< flush;
  if (!sdout){
    msg("SD Write Err");
    sdout.close();
    menu();}
  sdout.close();}
```

From the function the first line is to initialise the function to write in .CSV file. The variable fileName1 is the name of the file that will be write into for example like “attend.csv”. the ios::out statement is to create a file with that name if the file does not exist. Meanwhile, ios::app will add the data to the file rather than rewrite the whole file if the file existed already and the data is available. The next statement is writing the data into the file by using sdout instruction. The pstr(“,”) instruction will separate the data we put into the file by column. In the function above the ID will be in the first column and the Storage data in the second column. The pstr(“\n”) like in the C programming which is end line statement. The following line of instruction is the checking statement whether the writing process is done correctly or is there and error.

After creating the file and saving the data and record in the SD card, the system also have to read the recorded data to get certain value like which ID match with the fingerprint template in the fingerprint reader since the ID of the student are keyed in the system and the number that identify which fingerprint it belong is based on the position where the fingerprint template is saved. Therefore, the function Read_storage() is created to read the data and obtain the value to be used by the system. Below is the overall structure of how the function should look like.

```
int Read_storage(){
 int id =0;
 char c1,c2;
 ifstream sdin(fileName1);
 if (!sdin.is_open()){
  msg("SD open err");}
 while (sdin >> id >>c1>> storage>>c2){
  if(c1 != ',' || c2 != ',' ){
   msg("Error Storage");
   menu();}}
 if (!sdin.eof()){
  msg("Error Read");
  menu();}}
```

In the function above there is a declaration of variables to store the variable that will be read in the file. Then the instruction of ifstream sdin(fileName1) is to set the communication to read the SD card. Then the next part of the function is to check if the SD card file can be read, if not send an error message. Then there will a while loop to read the data row by row. The condition of while have to be the same as the one we write in the store data file. For example in the function above is based on the structure we wrote in the Write_storage() which is sdin>>id>>c1>>storage>>c2. The variable c1 and c2 is to check wherever there a comma in between id and storage. If the condition is not fulfil the loop will break and produce an error.

For the next function of the SD card is the clear data function which is Empty_data() function. This is the same as the write function but only writing zero value on the data and overwrite the file by setting the instruction at the ofstream sdout(fileName1,ios::out). That should be the only different in the emptying data. That is all the function of the SD card to write all the data in a .CSV file.

The RTC (Real Time Clock) Module

The module uses 2 pin to communicate with the system which is the SDA (data line) and the SCL (clock line) pin. To communicate with the RTC module which is a I2C/TWI device the library of Wire.h is used which has the setting of the communication for the RTC module. The library uses 7 bit addresses throughout to determine if it's being written or read from and get the value.

The next setting that is needed to get the data form the RTC module is to include the library of is the DS1307RTC.h. in this library we are able to set the value that we obtain from the RTC module and convert it to the proper data and time. We get the variable of the time in tmElements_t form this variable. The value of day, month, year, hour, minute and second by using the instruction tm.Day or tm.Month. We read the RTC module whether is working or not by using the instruction of RTC.read(tm) this is needed for the previous instruction tm.Day and tm.Month.

Other than there is another instruction to check the RTC module whether it is working or the signal of the module is cut off by using the instruction RTC.chipPresent(). This will tell the system where the connection of the RTC module is there or not. The RTC module needs to be set before we can used it by using a small program to synchronise the module with the computer system clock. This program can be found on the internet which is named SetTime.

The RTC module is read in the firmware is in the function. When the function is called the data of the time and date is taken from the RTC module the instances where the RTC function is called is during the menu state of the program and also during the attendance reading of the student to get the accurate time of the student that have their attendance taken during that time. To keep the data of time during that period of time the data of the time is saved in a few variable then stored in the SD card.

The following is the function of the RTC module to get the time:

```
void time()
{
  if(RTC.read(tm)){
    makeTime(tm);
    Day = tm.Day;
    Month = tm.Month;
    Year = tmYearToY2k(tm.Year);
    Hour = tm.Hour;
    Min = tm.Minute;
    lcd.setCursor(0,1);
    lcd.print(tm.Day);
    lcd.print('/');
    lcd.print(tm.Month);
    lcd.print('/');
    lcd.print(tmYearToY2k(tm.Year));
    lcd.setCursor(10,1);
    lcd.print(tm.Hour);
    lcd.print(':');
    lcd.print(tm.Minute);}
  else
  {
    if(RTC.chipPresent()){
      msg("RTC  stopped");}
    else{
      msg("RTC error");}
  }
}
```

As we can see in the function the firmware check whether the RTC module is present or not before starting it function to read the time. If there is an error during the reading of the RTC module the function will display the error accordingly if the RTC module is not connected properly it will display “RTC error” but if the display is “RTC stopped” that means the is a problem communicating with the RTC module.

The Keypad Module

The keypad check which button that is been pressed by checking the row and activating the column one by one. When one column is activated we check the each row if the signal is connected to the row that mean the button on that is pressed if not it goes to the next column to check it. If none of the row and column is connected it will repeat the cycle. This can be seen in the picture below:

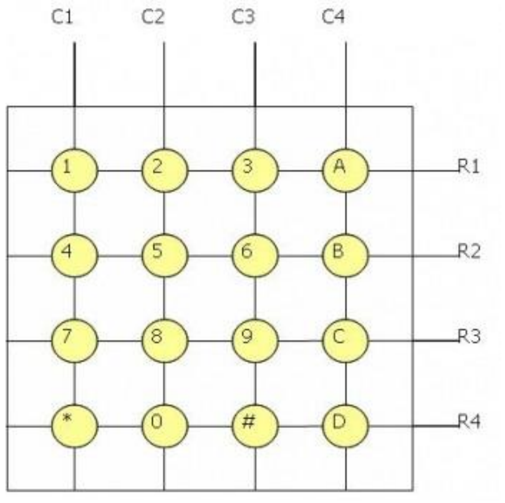


Figure 16: Keypad Structure on Button Pressed

In the function we activate column C1 therefore if the button 1, 4, 7 or * is pressed therefore either R1, R2, R3 or R4 will connect and produced a signal this process will be the same for the other columns. This will repeat until a button is pressed on the system reset. The signal that we expect to get is a high signal but due to noise of the wire or the hardware it make it hard to detect. Therefore the signal is check by using the analogue pin of the system to detect the value of the signal to reduce the detection of noise. The testing of the keypad will be explained on the next part of the paper

All this function is to help the main part of the firmware to be less space taking and also reduced the amount of line in the firmware for the same process. The coding for any function which is not shown in this part of the paper will be placed in the Appendices.

Casing Design

The lastly is the design of the casing for the prototype to make it easy to store and carry around. The design of the casing by using Perspex and using PCB stand to make as support. The Perspex design is based on a top and bottom platform which a rectangular shape plate of 177mm × 98mm x 3mm. The measurement is for both top and bottom plate. The drawing of the plate is done by Autocad program and the Perspex cutting is done in the lab by the technician based on the drawing. Below is the drawing ot the top and bottom plate:

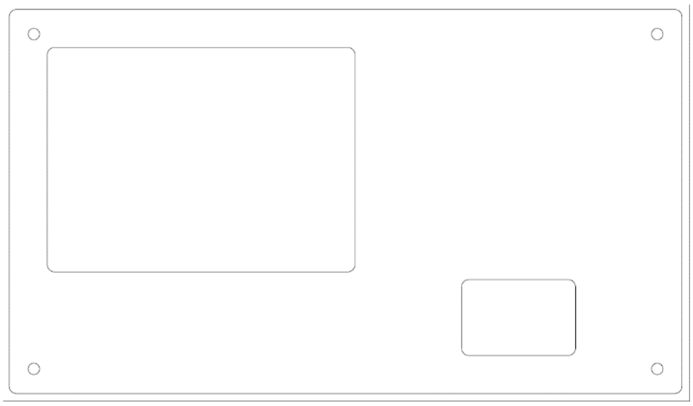


Figure 17: The Top Plate Drawing

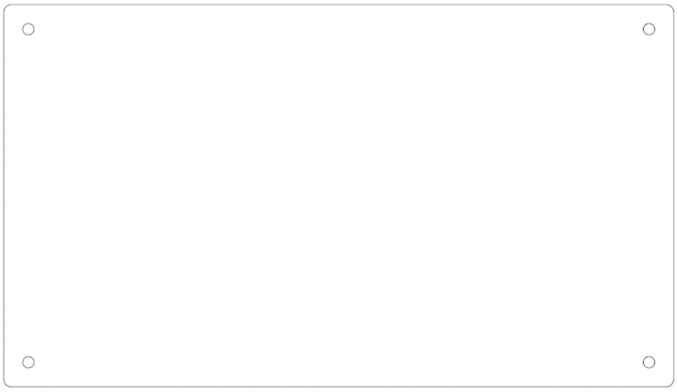


Figure 18: The Bottom Plate Drawing

As we can see from the drawing the top plate has two holes in the middle to display the LCD and the keypad module. The bigger hole is to display the LCD display and the smaller hole is for the connection of the keypad module to connect to connection in the casing. Meanwhile the bottom plate of the casing is empty and only has the hole for the screw of the PCB stand.

By using 4 × of 30mm and 4 × 20 mm of the PCB stand we can create the complete casing. The final result of the casing is as follows:

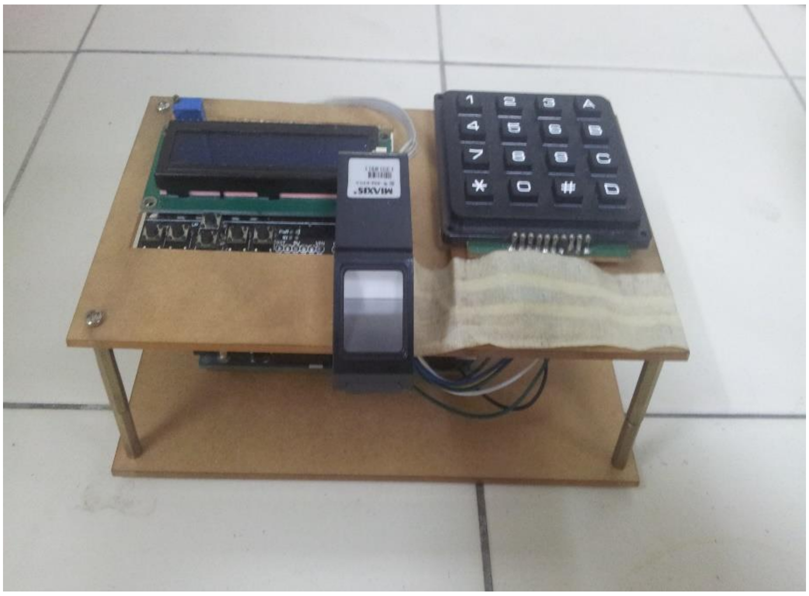


Figure 19: The Final Product of the Casing

From the final product of the casing the side of the casing is open for easy connection of the power supply and also to allow the easy removal of the SD card from the system. As from the picture, the keypad is placed outside and the connection is done through the hole of the casing.

From this the system is about the size of 177mm × 98mm × 56mm which is about the size of a small tissue box. This size should be easy for the user to carry around and be used in a classroom. The casing is also to allow the easy modification and programming if needed.

3.2.2 Prototype Testing

Hardware

The following will the testing of each part of the hardware individually to test whether the hardware obtained is working accordingly. The first hardware to test is the LCD display. This is because it is the basic function of the system and it will be used to display result of the other hardware testing later on. Here are the procedure of the LCD testing.

Testing of LCD Display

1. Stack the LCD Keypad Shield on the Arduino board.
2. Run program to display “Attendance System” on LCD.
3. Observed result.

The next part of the testing is the RTC module which will act the internal clock of the system. In this project, we only use the four pin in the RTC module only which are the VCC, GND, SDA and SLA pin. This four pin are important to activate, write and read the data in the RTC module to get the time. In this test, the LCD Display is used to display the time that has been obtained from the RTC module.

Testing of RTC Module

1. Connect the RTC module pin of VCC, GND, SDA and SLA to the Arduino board
2. Check whether the LED on the RTC module light up
3. Stack the LCD Keypad Shield to the Arduino Board
4. Program and run the program to set the time in the RTC module
5. Program and run another program to display the time of the LCD display
6. Observe the result.

The following hardware to test is the fingerprint reader module. To communicate with the fingerprint reader module the RS232 communication protocol is used. The module will wait for the host to send the right packet of data then the module will respond according the command of the packet data by sending back the confirmation packet data. Therefore to test the fingerprint module the board have to send the right packet data and wait for the respond of the module. For example, if we send a packet of data of 0x4D + 0x58 + 0x10 + 0x01 +0x46 + 0xFC, the respond should be 0x4D + 0x58 + 0x30 + 0x01 + 0x01 + 0xD7. The command packet we have send is to empty the database of the fingerprint reader module. In the Serial Monitor the value they show is in decimal therefore there is a need to change the value of the respond received form the Serial Monitor to hex value. The communication of the module uses a baud rate of 57600 bps. In the testing of the fingerprint module we will be using a sub program in the Arduino IDE to check the respond of the fingerprint module.

Testing of Fingerprint Reader Integrated SM630

1. Connected the fingerprint reader module to the Arduino board VCC, GND, RX, and TX pins.
2. Program to send "Empty Database" command and run the program in the board.
3. Open the Serial Monitor.
4. Set the baud rate to 57600 bps of the Serial Monitor.
5. Reset the board to send the command packet again.
6. Check the respond of the packet in the Serial Monitor.

The following hardware to test is the SD card reader module or for this project is the Ethernet Shield. In this test, the usage of the example given in the Arduino IDE is used to test the functionality of the SD card reader in the Ethernet Shield. The example used will be the "Datalogger" to test the SD card reader. In this program will check the SD card reader and also write a .txt file to the SD card. The testing procedure is as follows:

Test for SD card reader

1. Stack the Ethernet Shield on the Arduino Board.
2. Place a SD card into the SD card slot.
3. Program and run the program on the Arduino Board.
4. Open the Serial Monitor to check the output of the program
5. Power down the board and remove the SD card
6. Check the SD card for the file.

The last hardware to test is the keypad module for this prototype. In this test, we check the functionality of the keypad by pressing the button and check the whether the button pressed is correct by displaying the result in the serial monitor of the Arduino. The program testing the keypad is loaded to the Arduino main board and the keypad module is connected to the main board. The testing procedure is as follows:

Test for Keypad Module

1. Connect the keypad module to the main board.
2. Load the testing program into the board.
3. Run the program on the Arduino Board.
4. Open the Serial Monitor to check the output of the program.
5. Press the button 1 then check the display of Serial Monitor.
6. The step 5 is repeated with the other buttons.
7. After all the button is pressed we check the Serial Monitor.

This are all the main part of the hardware that is need to be tested to ensure the system is running properly before the firmware design of the biometric fingerprint system is tested or implemented to the system.

Performance testing

In this section of the report is the performance testing of the hardware used in the project mostly on the fingerprint reader module performance.

Fingerprint Module Integrated SM630 Respond Time Performance

1. The system is setup with the firmware loaded in the system
2. The LCD display keypad is pressed to move to the “Empty Database” section
3. The select button is pressed to start the process and the time for the fingerprint module to respond is taken
4. Once the display show “Data Empty” time taken is stopped.
5. The time taken is recorded.
6. This process is repeated for at least 10 times

Fingerprint Module Integrated SM630 Finding Fingerprint Accuracy Test

1. The system is setup and the firmware of the system is loaded.
2. The “Empty Database” function is used to empty the data in the fingerprint module
3. All ten fingerprint are registered into the system using the “Add fingerprint” function with all different ids.
4. The system is reset.
5. The “Attendance System” function is used to check for the first fingerprint register. The first fingerprint is placed when “Processing” is displayed.
6. The result produced by the function is observed and recorded.

The next performance test is to check the data record in the SD card based on the attendance we have taken. This file will record all the attendance in different time when a student record his attendance. In this test we have already recorded different fingerprint with different ID’s. The attendance of the student is recorded at different time. The process of the test is as follows:

The Performance Testing for Student Attendance Recording

1. The system is setup and the firmware of the system is loaded.
2. The “Empty Database” function is used to empty the data in the fingerprint module
3. All ten fingerprint are registered into the system using the “Add fingerprint” function with all different ids.
4. The system is then is reset.
5. The “Attendance System” function is used to recorded the attendance of the student at a specific time
6. Then the system is closed.
7. The system is then started. The “Attendance System” function is then used again on a different time.
8. The result is then checked in the .CSV file

The next test is a live test on a situation where a user used the system to register himself or herself and also try the system at the same time. In this test we try about 40 people to test the reliability of the system. Like the pervious test all the data of the attendance and registered student are recorded. This live test is done by different samples and also during different time to test how the system perform in a real situation.

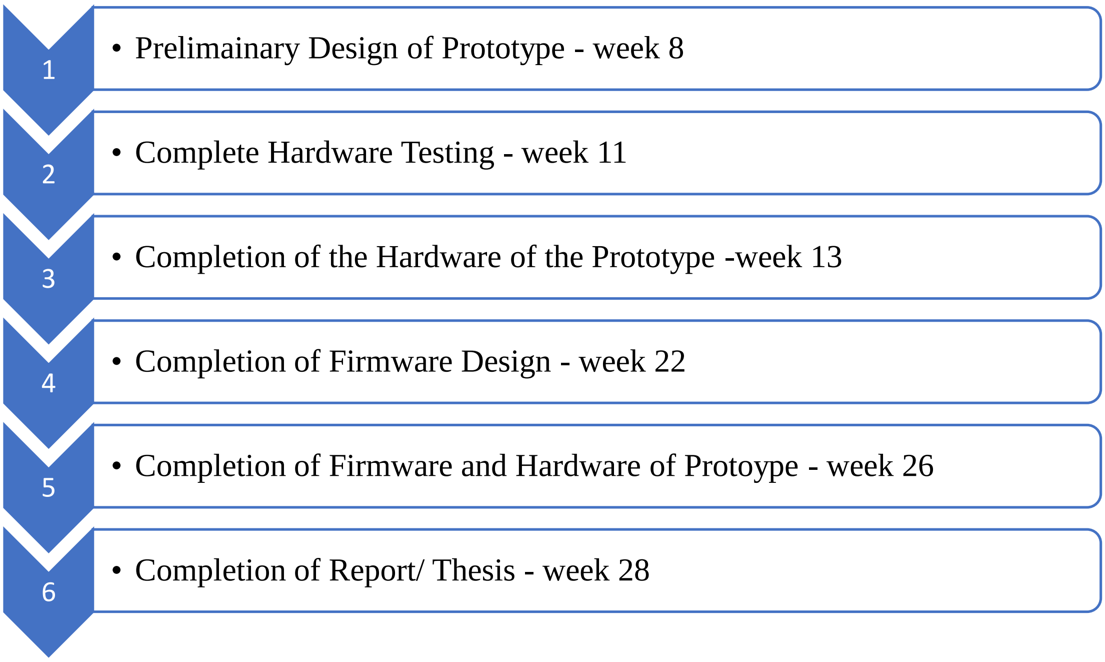
The Live Testing of the System

1. The system is setup and the firmware of the system is loaded.
2. The system is on until sample comes.
3. When sample arrive enter “Add student” mode to register sample to the system.
4. The sample enter the “Attendance System” mode to record the attendance of the sample.
5. The process is repeated for the other samples.
6. The result is checked in the .CSV file

This is all the performance testing of the system all the result of the testing can be seen in the result and discussion part of this paper.

3.3 Key Milestones

The following figure will explain the key milestones of this project based on the submission dates given by the FYP coordinators:



- Prelimainary Design of Prototype - week 8
- Complete Hardware Testing - week 11
- Completion of the Hardware of the Prototype -week 13
- Completion of Firmware Design - week 22
- Completion of Firmware and Hardware of Protoype - week 26
- Completion of Report/ Thesis - week 28

Figure 20: Key Milestones for FYP

3.4 Gantt Chart

Table 2: Gantt Chart for FYP

NO	ACTIVITIES	WEEK																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	TITLE SELECTION	■	■																										
2	PRELIMINARY RESEARCH AND LITERATURE REVIEW		■	■	■	■	■																						
3	PRELIMINARY DESIGN OF PROTOTYPE						■	■	■																				
4	HARDWARE TESTING								■	■	■	■																	
5	BUILDING PROTOTYPE									■	■	■	■	■															
6	FIRMWARE DESIGN										■	■	■	■	■	■	■	■	■	■	■	■	■						
7	FIRMWARE TESTING AND TROUBLESHOOTING															■	■	■	■	■	■	■	■	■	■	■			
8	COMPLETION OF FIRMWARE AND PROTOTYPE																							■	■	■	■		
9	REPORT/THESIS																		■	■	■	■	■	■	■	■	■	■	■

CHAPTER 4

RESULT AND DISCUSSION

4.1 Hardware Testing Result

In the early stage of the project there are some hardware testing that have been done. From the test we managed to test the functionality of all the hardware and all of the hardware are in working condition.

4.1.1 LCD Display Result

The LCD Display is working properly when tested individually and also when the whole system is combined. This can be shown in the picture below:

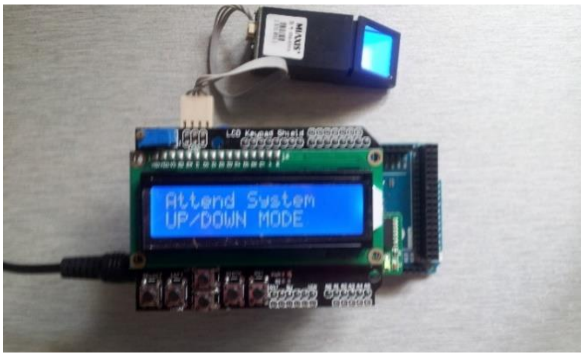


Figure 21: The Result of LCD Display Testing

4.1.2 RTC Module Result

The RTC Module managed to display the time of the clock according to the time of the computer used. The result is shown below:

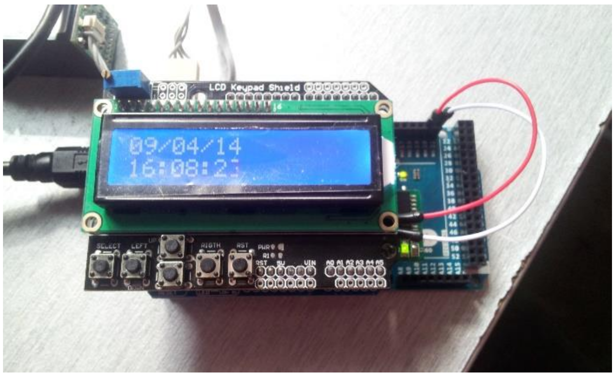
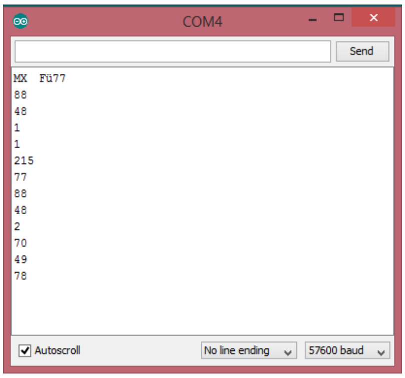


Figure 22: The Result of RTC Module Testing

4.1.3 Fingerprint Reader Module Result

The testing of the fingerprint module show that the fingerprint module is working is properly without any error.



```
MX  Fü77
88
48
1
1
215
77
88
48
2
70
49
78
```

☑ Autoscroll

No line ending

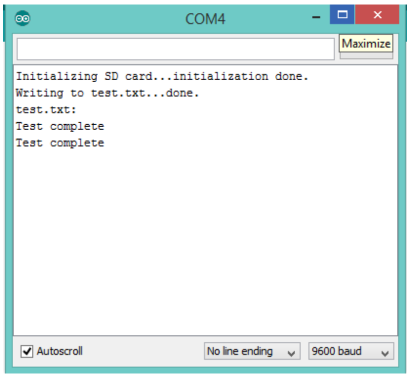
57600 baud

Figure 23: The Result of the Respond of the Fingerprint Reader Module

As mentioned before the code we send is in hex file while the output on the Serial Monitor is in decimal. Therefore there is a need to convert this output shown into hex. In this result the data we send is the Empty Database code which is 0x4D + 0x58 + 0x10 + 0x01 + 0x46 + 0xFC. Therefore the we received should be as follows: 0x4D + 0x58 + 0x30 + 0x01 + 0x01 + 0xD7 and 0x4D + 0x58 + 0x30 + 0x02 + 0x46 + 0x31 + 0x4E. this can be seen in the picture above.

4.1.4 SD Card Reader Module Result

The SD card reader managed to create a .txt file in the SD card and be used in the computer. The result of the test is as follows:



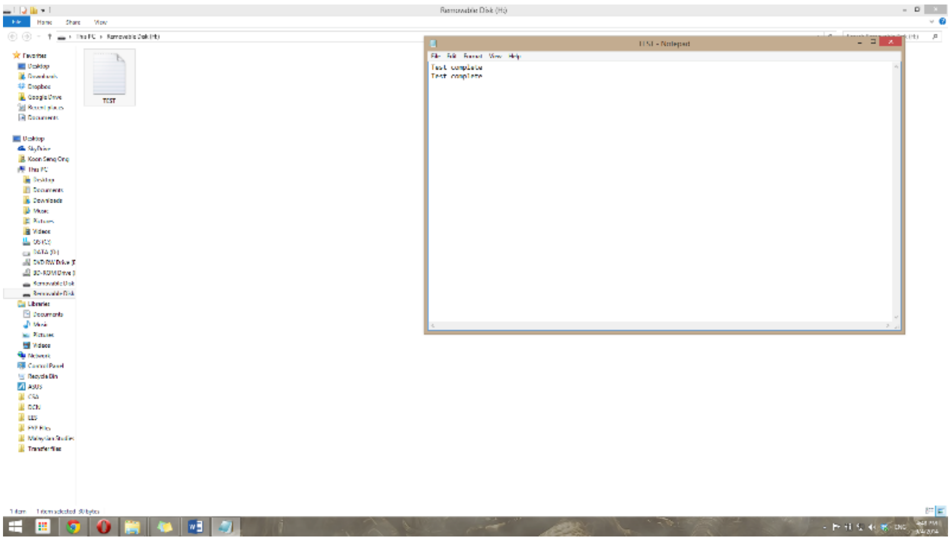
```
Initializing SD card...initialization done.
Writing to test.txt...done.
test.txt:
Test complete
Test complete
```

☑ Autoscroll

No line ending

9600 baud

Figure 24: The Serial Monitor result in writing a txt file in the SD card

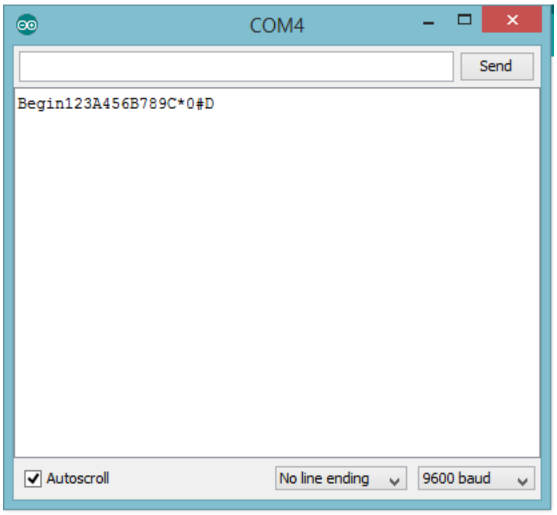


```
Test complete
Test complete
```

Figure 25: Screenshot on the .txt file created in the SD card and the content of the .txt file

4.1.5 Keypad Module Result

In the test the keypad is working perfectly and displaying all the correct numbers and char according to the button that is pressed. The result of the test is as follows:



```
Begin123A456B789C*0#D
```

☑ Autoscroll

No line ending

9600 baud

Figure 26: The Result of the Keypad Module Test

From the figure above we can see that all the buttons in the Serial Monitor window is shown which shows that the keypad is working. The table in the next page show the tabulation of the result.

Table 3: The Tabulation of Keypad Module

Test	The button pressed	Display result
1	1	1
2	2	2
3	3	3
4	A	A
5	4	4
6	5	5
7	6	6
8	B	B
9	7	7
10	8	8
11	9	9
12	C	C
13	*	*
14	0	0
15	#	#
16	D	D

4.2 Performance Testing Result

4.2.1 Fingerprint Module Integrated SM630 Respond Time Performance

Below is the table of the result of the respond time of the fingerprint module. According to the result the average respond time of the module is 1.17s.

Table 4: Test Result for Respond Time Performance of Fingerprint Module

Test	Time Taken (s)
1	1.3
2	1.0
3	1.1
4	1.2
5	1.2
6	1.1
7	1.3
8	1.2
9	1.2
10	1.1

4.2.2 Fingerprint Module Integrated SM630 Finding Fingerprint Accuracy Test

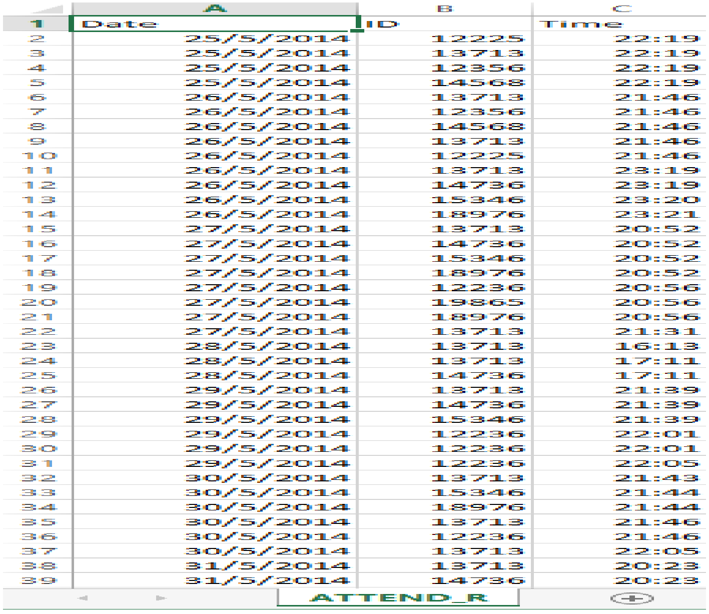
Below is the test result of the accuracy of the fingerprint module. For the result obtained we can conclude that the fingerprint has an accuracy of 100% in identifying the fingerprint.

Table 5: The Result of the Test of Accuracy of the SM630 Fingerprint Reader Module

No.	Fingerprint	Student ID register	Student ID Found
1	1	13713	13713
2	2	12244	12244
3	3	19988	19988
4	4	17788	17788
5	5	12365	12365
6	6	14785	14785
7	7	16633	16633
8	8	19999	19999
9	9	16767	16767
10	10	12222	12222

4.2.3 The Performance Testing for Student Attendance Recording

The result of the test show that the prototype manage to record the student attendance data which includes the ID of the student and the time of the student using the system for attendance recording. The result of the test is shown below:

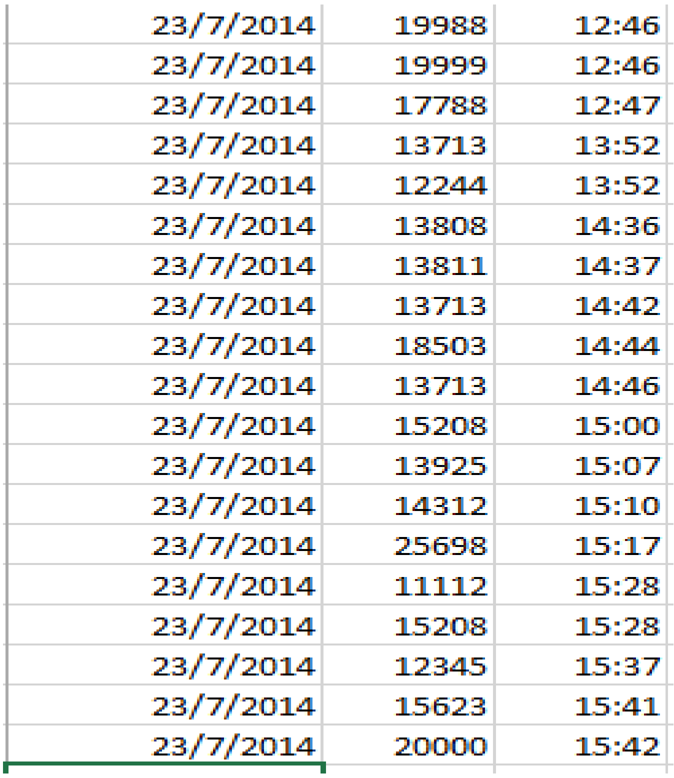


	A	B	C
1	Date	ID	Time
2	25/5/2014	12225	22:19
3	25/5/2014	13713	22:19
4	25/5/2014	12356	22:19
5	25/5/2014	14568	22:19
6	26/5/2014	13713	21:46
7	26/5/2014	12356	21:46
8	26/5/2014	14568	21:46
9	26/5/2014	13713	21:46
10	26/5/2014	12225	21:46
11	26/5/2014	13713	23:19
12	26/5/2014	14736	23:19
13	26/5/2014	15346	23:20
14	26/5/2014	18976	23:21
15	27/5/2014	13713	20:52
16	27/5/2014	14736	20:52
17	27/5/2014	15346	20:52
18	27/5/2014	18976	20:52
19	27/5/2014	12236	20:56
20	27/5/2014	19865	20:56
21	27/5/2014	18976	20:56
22	27/5/2014	13713	21:31
23	28/5/2014	13713	16:13
24	28/5/2014	13713	17:11
25	28/5/2014	14736	17:11
26	29/5/2014	13713	21:39
27	29/5/2014	14736	21:39
28	29/5/2014	15346	21:39
29	29/5/2014	12236	22:01
30	29/5/2014	12236	22:01
31	29/5/2014	12236	22:05
32	30/5/2014	13713	21:43
33	30/5/2014	15346	21:44
34	30/5/2014	18976	21:44
35	30/5/2014	13713	21:46
36	30/5/2014	12236	21:46
37	30/5/2014	13713	22:05
38	31/5/2014	13713	20:23
39	31/5/2014	14736	20:23

Figure 27: The Result of Attendance Recording

4.2.4 The Live Testing of System

The result of this test show that the prototype can be used in a practical situation to register and take attendance of the student. The 40 sample used in this test show to have manage to record their fingerprint in the system and be used to take attendance and record it in the SD card. The result of the recorded attendance is shown below:



23/7/2014	19988	12:46
23/7/2014	19999	12:46
23/7/2014	17788	12:47
23/7/2014	13713	13:52
23/7/2014	12244	13:52
23/7/2014	13808	14:36
23/7/2014	13811	14:37
23/7/2014	13713	14:42
23/7/2014	18503	14:44
23/7/2014	13713	14:46
23/7/2014	15208	15:00
23/7/2014	13925	15:07
23/7/2014	14312	15:10
23/7/2014	25698	15:17
23/7/2014	11112	15:28
23/7/2014	15208	15:28
23/7/2014	12345	15:37
23/7/2014	15623	15:41
23/7/2014	20000	15:42

Figure 28: The Result for Live Testing

4.3 Discussion

Form the result we obtain from the hardware testing we can conclude that most of the hardware that is being used in the prototype are working accordingly when tested individually. This mean when the prototype is combine and some of the module are not working as they should be the issue is not the module itself but on the connection or the wiring of the module to the main board. Other than that, it could mean that there is not enough power or current to make the module work or the programming in the system need to be adjusted to allow the module to work properly. This testing of hardware individually allows troubleshooting the system easily and allow to pinpoint the problem when the prototype does not work as it is supposed to.

In the performance testing of the fingerprint reader module we managed to test the reliability and the response time of the module to allow us to take account how the module need to react during the programming of the fingerprint reader module. From the result of the test, the fingerprint module should be able to work perfectly to attendance of the student without producing a false result or an error during the recording the student attendance in the system.

The second performance test is the basic requirement of the prototype which to take the attendance of the students and also record the time when the attendance was taken. The result show that the prototype manage to do the basic requirement of the project, which mean the prototype of the Biometric Finger-Print Based System for Time and Attendance Recording is a success and it is working properly.

The last testing which is the live test of the system shows that the system is working properly in a real live situation. The system manage to detect the fingerprint even though the fingerprint sample is taken from the side or from any position as long as the reader can read the whole fingerprint. There is one time where the reader can read the sample due to a smudge on the fingerprint reader but this is solved easily by cleaning the reader and also the sample fingerprint.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The result of the tests shows that all the hardware of the prototype and the performance of the fingerprint reader module is working perfectly in the system and optimally. This mean that the system if combined together should be able to work given the proper firmware design and the circuit design of the whole system.

The performance testing of the record attendance shows that the prototype managed to take the attendance of the students and the time when the attendance is taken. This show that the prototype has meet it minimum requirement of the project which is:

I. Able to take time and attendance.
II. Save the recorded data digitally.
III. Easy to be used by student and lecturer.

In the last test of the project, the prototype is used in a real live situation by using 40 sample to test the prototype. The result of the test shows that the prototype is working properly in the situation given even though there is a instance where the prototype produce an error due to the sample fingerprint is smudge and unable to read the fingerprint properly.

Overall the prototype is working properly and can be implemented in a real live situation of a lecture class.

5.2 Recommendation

There are still some minor problem that can be found in the prototype which is the power consumption of the prototype is too high to be used by a battery power supply due to the hardware used. The prototype have to use at least a power supply of 12V and 2A to be fully operational without any problem. This can be solve by using different modules in the prototype which takes less power for example the SD card module.

Other than that, the prototype used a different ways of firmware design to allow the prototype to function much more efficiently and take less time for attendance. For example, the attendance of the student is taken when the fingerprint is detected rather than entering a mode to detect the student attendance.

Lastly the design of the casing can be improved to be much more compact and easier to be carry by the user. The current casing need the user to carry the prototype like a box rather than something portable.

That is all the recommendation of the improvement of the prototype at this current time.

References

[1] G. Aguilar, G. Sanchez, K. Toscano, M. Salinas, M. Nakano, and H. Perez, "Fingerprint Recognition," in *Internet Monitoring and Protection, 2007. ICIMP 2007. Second International Conference on*, 2007, pp. 32-32.

[2] S. Zhou and X. Lu, "Fingerprint Identification and its Applications in Information Security Fields," pp. 97-99, 2010.

[3] M. S. Alam and M. Akhteruzzaman, "Real time fingerprint identification," in *National Aerospace and Electronics Conference, 2000. NAECON 2000. Proceedings of the IEEE 2000*, 2000, pp. 434-440.

[4] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 14, pp. 4-20, 2004.

[5] R. Chunxiao, Y. Yilong, M. Jun, and Y. Goping, "Feature selection for sensor interoperability: A case study in fingerprint segmentation," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009, pp. 5057-5062.

[6] D. Kumar and R. Yeonseung, "A Brief Introduction of Biometrics and Fingerprint Payment Technology," in *Future Generation Communication and Networking Symposia, 2008. FGCNS '08. Second International Conference on*, 2008, pp. 185-192.

[7] T. Hninn Thiri, M. M. Sein, and A. Saw Nay La, "A Reliable Technique for Personal Identification or Verification," in *Micro-NanoMechatronics and Human Science, 2007. MHS '07. International Symposium on*, 2007, pp. 265-269.

[8] P. J. Phillips, A. Martin, C. L. Wilson, and M. Przybocki, "An introduction evaluating biometric systems," *Computer,* vol. 33, pp. 56-63, 2000.

[9] M. Fons, F. Fons, N. Canyellas, Canto, x, E., *et al.*, "Hardware-software co-design of an automatic fingerprint acquisition system," in *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on*, 2005, pp. 1123-1128 vol. 3.

[10] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: security and privacy concerns," *Security & Privacy, IEEE,* vol. 1, pp. 33-42, 2003.

[11] B. Ashwini, J. S. Digambarrao, and S. P. Patil, "Performance analysis of finger print sensors," in *Mechanical and Electronics Engineering (ICMEE), 2010 2nd International Conference on*, 2010, pp. V1-169-V1-174.

[12] L. Lugini, E. Marasco, B. Cukic, and I. Gashi, "Interoperability in fingerprint recognition: A large-scale empirical study," in *Dependable Systems and Networks Workshop (DSN-W), 2013 43rd Annual IEEE/IFIP Conference on*, 2013, pp. 1-6.

[13] P. Grother and E. Tabassi, "Performance of Biometric Quality Measures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 29, pp. 531-543, 2007.

[14] S. J. Elliott, S. K. Modi, L. Maccarone, M. R. Young, J. Changlong, and H. Kim, "Image Quality and Minutiae Count Comparison for Genuine and Artificial Fingerprints," in *Security Technology, 2007 41st Annual IEEE International Carnahan Conference on*, 2007, pp. 30-36.

[15] F. Alonso-Fernandez, R. N. J. Veldhuis, A. M. Bazen, J. Fierrez-Aguilar, and J. Ortega-Garcia, "Sensor Interoperability and Fusion in Fingerprint Verification: A Case Study using Minutiae-and Ridge-Based Matchers," in

Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on, 2006, pp. 1-6.

[16] F. Alonso-Fernandez, J. Fierrez, J. Ortega-Garcia, J. Gonzalez-Rodriguez, H. Fronthaler, K. Kollreider, *et al.*, "A Comparative Study of Fingerprint Image-Quality Estimation Methods," *Information Forensics and Security, IEEE Transactions on,* vol. 2, pp. 734-743, 2007.

[17] K. Gruttner and W. Nebel, "Modelling Program-State Machines in SystemC&#x2122;," in *Specification, Verification and Design Languages, 2008. FDL 2008. Forum on*, 2008, pp. 7-12.

[18] C. Angelov, K. Xu, G. Yu, and K. Sierszecki, "Reconfigurable State Machine Components for Embedded Applications," in *Software Engineering and Advanced Applications, 2008. SEAA '08. 34th Euromicro Conference*, 2008, pp. 51-58.

APPENDICES

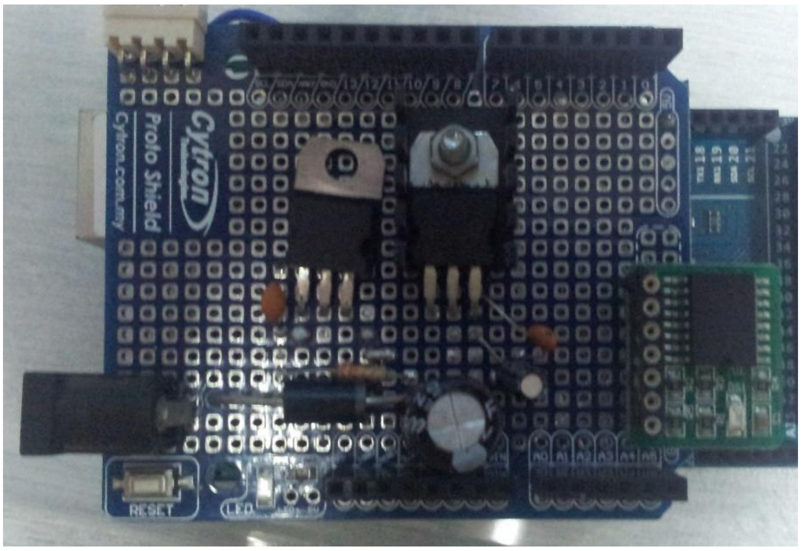


Figure 29: The Power Circuit of the System

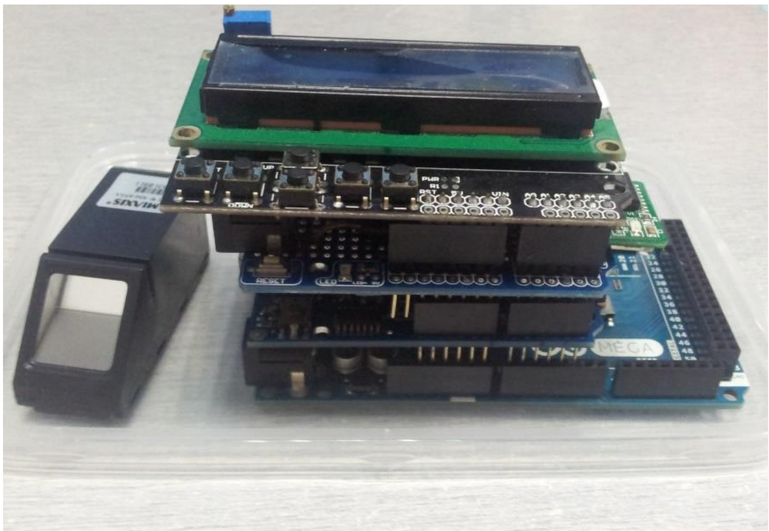


Figure 30: The Complete Integration of the System

Coding Functions

The Fingerprint Module Respond Coding

```
boolean f_respond(byte cmd){
  boolean end_flag =false;
  boolean success = false;
  byte addfp_count =0;
  while(!end_flag){
    int packet_length = 9;
    byte resp_temp[packet_length];
    for(byte j=0; j<packet_length; j++){
```

```
    while (Serial.available()==0) {
      msg("Processing"); }
    resp_temp[j]=Serial.read();
    if(j==3) packet_length = resp_temp[3]+5; }
  byte response[packet_length];
  for(byte j=0;j<packet_length;j++) {
    response[j]=resp_temp[j]; }
  switch(resp_temp[5]) {
    case OP_SUCCESS:
    if (cmd == SEARCH_FINGER) {
      msg("Finger Search"); }
    else if (cmd == ADD_FINGER) {
     msg("Finger Added");
     addfp_count++; }
    else if(cmd == EMPTY_DATABASE){
      msg("Data empty");
      end_flag = true;
      success = true; }
    break;
    case PROCESS_FAILED:
    msg("Process fail");
    end_flag = true;
    break;
    case FP_FOUND:
    msg("Finger Found");
    student_id_S = response[6]*256+response[7];
    end_flag = true;
    success = true;
    break;
    case FP_UNFOUND:
    msg("Finger Miss");
    end_flag = true;
    break; }
  if (cmd==byte(ADD_FINGER)&&addfp_count>1) {
    success = true;
    end_flag = true; } }
Serial.end();
return success; }
```