Biometric Voting System

Group Members

Abdul Moiz **(140273)**

Muhammad Ahtesham Zafar **(140339)**

Muhammad Osama Shahzad **(140405)**

Bachelors of Electrical Engineering

(2014-2018)

Project Supervisor

Muhammad Ali Raza

(Lecturer)

DEPARTMENT OF ELECTRICAL ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

Biometric Voting System

Final Year Project Report

(Session 2014-2018)

DEPARTMENT OF ELECTRICAL ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

A Final Year Project Report

Presented to

Air University

in partial fulfillment of the requirements for

Degree of

Bachelors of Electrical Engineering

Biometric Voting System

SUBMITTED BY

Abdul Moiz	**(140273)**
Muhammad Ahtesham Zafar	**(140339)**
Muhammad Osama Shahzad	**(140405)**

(2014-2018)

Department of Electrical Engineering

May,2018

Approval for Submission

It is to certify that the project report titled

"BIOMETRIC VOTING SYSTEM"

has met the required standard for submission

in partial fulfillment of the requirements

for the award of degree of

Bachelor of Electrical Engineering

at Air University, Islamabad

Project Supervisor

Engr. Muhammad Ali Raza
(Lecturer)

Head of Department

Dr. Shahid Baqar

Dedication

This Research is dedicated to Allah SWT for his blessings

upon us, our beloved parents

who nurtured us and made us what we are.

Acknowledgement

This Final Year project is not just the work of three people but the efforts, help, guidance, mentorship and prayers of many other people are to be credited for this success. We are thankful to each source and person who helped making this project great and hope that we will keep getting their guidance to learn more and do greater things.

Firstly, we would like to acknowledge the faith of our mentor, **Mr. Ali Raza** for showing faith in this project and giving us a chance to take this project according to our own vision and wishes. We are thankful to respected sir for all the hours of guidance, help and the effort he has put in this project. Its only because of his vision and dedication that all this dream was constructed into reality. His countless meetings and boundless time made all the obstacles in this research just the wall of sand. We feel blessed to have him as our supervisor.

We would acknowledge the efforts and work of Dr. Waseem Khan who headed all the FYP groups and guided the groups regarding the FYP thesis writing, research methodology and individual group's confusions in the very beginning.

We would like to acknowledge the faculty and staff of Air University, specially the Department of Electrical Engineering and Computer Science for helping us and guiding us whenever we were worried and stressed, the faculty always gave us time and helped us with our findings and research methodology.

We also acknowledge all the free online training programs, their sponsors and creators. We strongly feel that such programs and video lectures help the researchers and students a lot to get the inspiration and necessary technical knowledge.

Abstract

Biometric Voting System will be a multi constitution supported, secure and fast system that will be used for thrifty elections process. We strived to create an easily customizable and upgradeable system. The lack of ability to incorporate new updates in EVM is posing a serious question over their performance [1]. The new EVM soon after their introduction becomes hackable and vulnerable due to their un ability to keep up with the flow of time.

The proposed machine will be customizable and easily upgradeable. Universal equipment will be chosen to allow their compatibility with the more upgraded on chip computer (Raspberry Pi). It will make the machine more relevant, more practical and upgradeable in the future.

The proposed machine has a strong GUI element. The Biometric Voting Machine supports colorful and interactive, specially designed interface. The graphical interface, like all the other interfaces is entirely touch input based interface. It makes understanding the working of machine to general public easier as now touch interfaces are quite common in Pakistan ranging from Biometric Verification System for money transfer to ATMs.

This machine being proposed here will have the ability to support multiple constituencies vote. Using the object oriented programming (OOP) and hierarchy concepts and making main classes and sub classes, multiple constituency voting was made possible without a glitch.

The proposed system is designed to be very practical, smart and will be able to provide a solution to bulky and heavy paper based ancient Voting System. The Biometric Voting Machines will require the one-time investment to set up and in returns will take care of the ballot boxes, papers, inks, and huge manpower.

The proposed system is not just a biometric verification or voting system but also a result declaration system. It will declare the results after the required criteria has been achieved. This will automate the voting process and hence will make the declaration of results ahead of time possible. Hence, political, establishment interference and institutional rigging claims will die in the nip and hence will greatly help to make the election process more secure and end all the controversies.

The proposed system will be based on a very powerful language, Python Programming language. It itself is the name of huge number of possibilities, if cashed properly. Creating the neural networks to verify rigging and making the machine intelligent and take necessary steps is quite possible as Python is the language the world has gone to for machine learning after dumping MATLAB.

Table of Contents

List of Figures

List of Tables

Nomenclature

BVM	Biometric Voting Machine
GUI	Graphical User Interface
IDE	Integrated Development Environment
I/O pins	Input / Output pins
ECP	Election Commission Pakistan
LCD	Liquid Crystal Display
OOP	Object Oriented Programming

Chapter 1: Introduction

"Biometric Voting System", that's not just the name of a project but a motivation. That motivation started in 5th semester when a group of 3 Air University students wished to design and create something new to support the voting process and end or at least minimize the concept of electoral rigging but making a cheap and practical system.

The problem with the present electoral system is that it doesn't have any single problem but it is filled with major flaws, the biggest of them is that it is entirely dependent on human involvement from ballot papers printing to the supply of all the voting material to polling station. The current security situation and the involvement of powerful landlords also makes this electoral system impractical.

Recently a biometric machine is being tested in Pakistan that also works like this machine under development to some extent, but the machine under development is miles ahead of the machines currently being tested in Pakistan. The Biometric Voting polling station in Pakistan currently contains two machines, one of the machine is only "Biometric Verification" machine and after biometric verification the voter continues to the second machine that is used to produce a printed ballot paper. This system is expensive (as two heavy machines are being used), complex, impractical (as it is time consuming) and still fails to automate the process of Result calculation and declaration.

The scheme is to create a voting machine that works with multiple constituencies and can securely take paper out of the equation, not just do that but also successfully and securely automate the process of declaration of result of the specific booth.

The proposed machine will use the advanced concept of Graphical User Interface (GUI). The prepared and well-cooked logic will later be coded into GUI awesome mechanism. The GUI will support touch based input and will make the process of voting more visual and user friendly, hence making the machine more suitable to use for general audience.

A separate database will be created for sake of more security and privacy. Database will also keep a check on process of rigging and dual voting. The main program will access the database after matching the fingerprint and will fetch data of that particular user.

For example, in a specific city of Faisalabad, the machine will be able to end the hustle of running for booth to booth looking where their vote is but the citizen will go to any booth and will cast his/ her vote by only carrying his

most secure identity in form of his own fingers, secondary check will be CNIC. It will also save government billions that is spent on magnetic ink, counting, complex declaration of result, rigging allegations, recounting cases and record maintenance.

1.1 Problem Statement:

"Inefficient biometric voting system and the old human dependent electoral system are the cause of rigging in institutional state which results in making the whole process of elections questionable. Moreover, massive resources are wasted while supporting the outdated electoral system with antiques such as magnetic ink, ballot paper and all the supporting material and manpower."

1.2 Circuit Diagram:

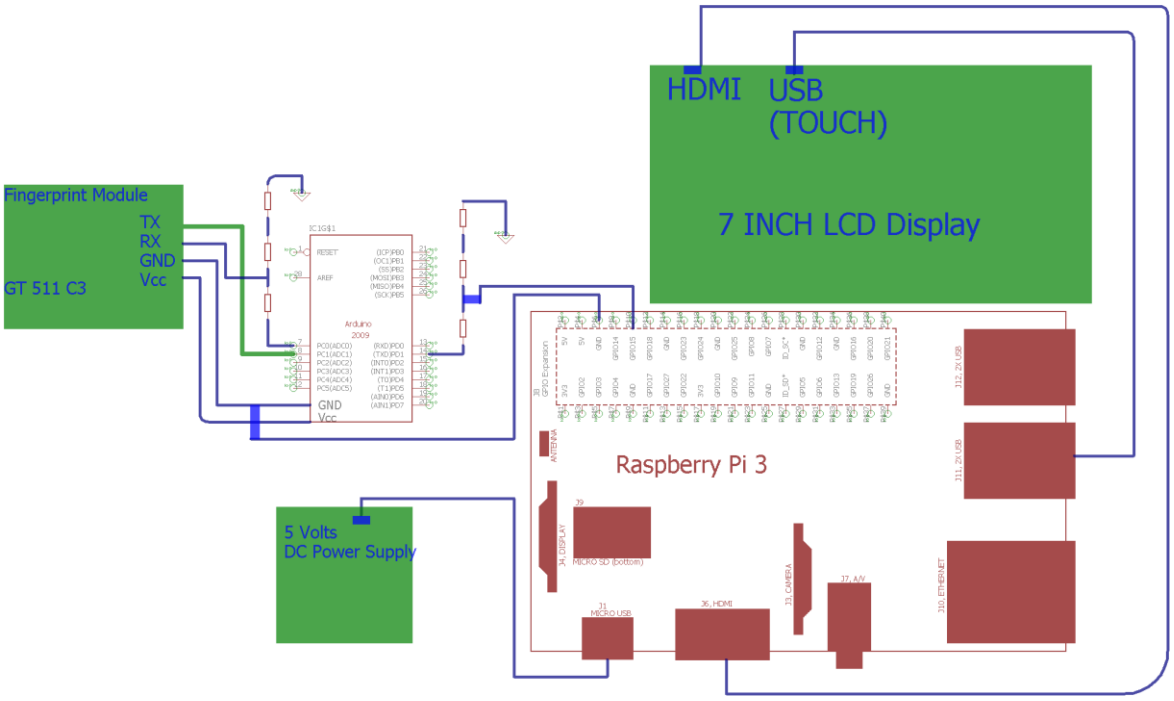


Figure 1 Circuit Diagram

1.3 Motivation:

In Pakistan, electoral rigging is quite common and it makes the whole process of voting controversial. For a democratic process, the voting and reliable electoral process is very important as people choice is the base of democracy and by making electoral system questionable, the whole democratic process becomes questionable.

Pakistan is a democratic country hence, it is very necessary for it to have a secure and sustainable voting system. Hence, our motivation is to lay first brick and be the first drop of water by creating the Biometric Voting System of the new era. We are set of individuals highly inspired by Machine Learning and deep learning algorithms and do feel that if such algorithms are to be introduced in voting machines, it will turn old machines into highly autonomous and intelligent voting systems.

Sadly, no ground work has been done previously in this regards hence, at this level we wish to lay foundation and do the groundwork by creating a customizable and working model of EVM that is able to work at the minimal level of standards and support multiple constituency voting.

1.4 Objectives:

When a project is started, it is started with certain goals and objectives in mind. In the start, it is just a scratch but turning the scratch into a fully functional machine need some sharp and well-planned objectives otherwise the misdirection will not only waste much of the precious time but also will result in misjudgments which even can result in wastage of funds and capital.

The objectives of this project are to:

- To reduce rigging
- To increase voting turnout.
- To revolutionize country’s electoral system.
- To commercialize the fully formed product.
- Creating a user friendly Graphical User Interface.
- Creating a Voting System that supports multiple constituencies.
- Create a reliable, customizable and upgradeable Voting System.
- Forming a single machine based certain level independency and automation.

1.5 Scope of Project:

This Voting machine has unmatched scope in for any type of elections, voting, polling session, and surveys. This machine can be used for private elective voting such as a club's president voting, voting of an agenda, pooling for a cause to know people opinion, taking feedback and bar's election etc.

After some modifications and upgrades, this machine has the scope to be used in General elections also.

1.6 Cost Analysis:

Sr. No	Name of Equipment	No. of Equipment	Price
1	Raspberry Pi 3 Kits	3	5700x3= Rs/-17100
2	LCDs (Touchscreen)	2	7000x2= Rs/-14000
3	LEDs (for display)	2	5000+1000=Rs/-6000
4	Misc. (Casing, TTL module, Supporting equipment & training)	N/A	Rs/-20,000
	GRAND TOTAL		Rs/-57,100

Table 1 Cost Analysis

Chapter 2: Literature Review:

The concept of Biometric Voting System is not alien to Pakistani society and people. Much work has been already done in this prospect. Many papers have been published, research work has been done and institutional projects are being run under humongous government institutes e.g. Election Commission Pakistan.

Voting in the United States was first introduced in 1964 when 7 counties changed this method for the presidential election. Electronic voting refers to computerized voting machines used to vote electronic ballots instead of paper ballots. Nowadays, electronic voting is also used in other countries such as: Brazil, Norway, Germany, Venezuela, India, Canada, Belgium, Romania, Australia, United Kingdom, Italy, Ireland, European Union and France.

An Electronic voting machine is an electronic machine which voters use to register their votes. All votes are record and counted electronically. This voting machines have their security technologies which are the following: Smart cards, Firewalls, Antiviral software and Cryptography such as digital signatures and certificates, encryption, etc.

Biometric Voting Machine (BVM) is not a new concept for the world. Most of the new world, Europe, America, Far East etc., have been using it [2].

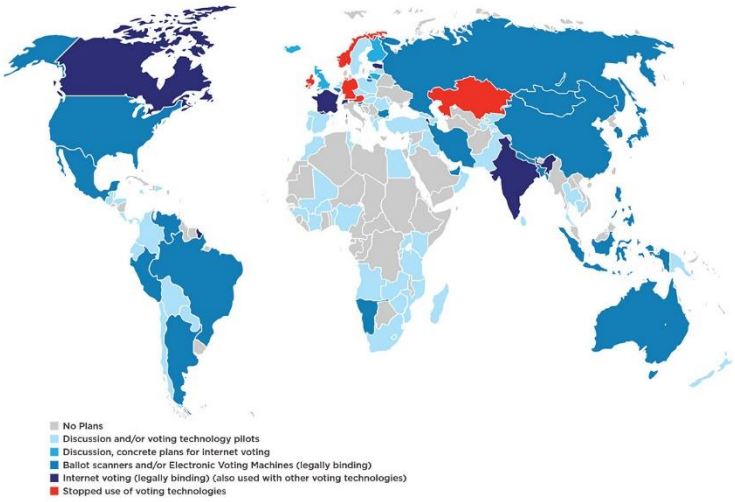


Figure 2 World map showing BVM implementations

The Biometric Voting Systems already made in Pakistan and which are being tried to be implemented are extensively studied before the start of this project. It was found out that the previous systems were not without major flaws and only provided a single solution in whole country which is itself a beg flaw. E.g. the last presidential election in United Stated provided three options to be selected from for voting. Firstly, they provide the option to print out the ballot paper and then put it in an automatic computer scanner. Basically there are 2 types of voting machines, one is printed ballots and second is direct recording electronic voting system and United States has incorporated both of them in the system.

Paper based electronic Voting System is just like the general voting system in which every voter is issued a ballot paper which is marked by electronically readable ink filled in specially designed pens. The casted vote is later counted by specially designed scanner. This way, the whole voting system is made secure and the result is automated.

2.1 Electronic Voting in USA:

There are three types of electronic voting in the United States that can be identified as: paper-based electronic voting system, electronic direct registration system, and Internet voting. The paper based E voting system has an added touch screen for the voter used. This system will print a hard copy of the ballot once the elector has finished. This ballot must be distributed to the returning officer so that it can be counted. The electronic direct record system has a touch screen with digital magnetic card buttons that will be used to make the choices. All votes are stored in a physical memory device that are sent to a special polling station for their results. Finally, Internet voting is the type of voting that is done in remote places. This type of voting service is not supervised by government officials. The most commonly used devices for this type of voting are: Personal Computer, Internet TV, also called i-voting and mobile phone.

Today, all US states use **eVoting** because it offers lower fees, improves accessibility for voters with disabilities, faster results, greater accuracy, and lower risk of mechanical errors. and human.

Figure 3 Ballot Paper Automatic Scanner used in USA Presidential Elections

In direct recording Electronic Voting Machines, firstly after authentication from many available ways, the voter chooses the candidates and later, the ballot is automatically printed in the ballot box. These voting machines limit the use of stamps and magnetic ink and automates the voting process to a certain standard still the processes of votes counting and declaration of results is completely human

dependent and is not automated so human interference and rigging is not completely phased out from the equation.

Tough the process can be automated by printing a QR code on printed ballot papers and later designing a machine to read those ballot paper in large numbers but this is only a theory and will cause unnecessary logistics complications and budget surge.

2.2 Electronic Voting in India:

Electronic voting machines ("EVM") are used in the general elections and Indian elections to carry out electronic voting of the 1999 elections, and recently in 2017 state elections held in five states across India. EVMs replaced the voting paper in local, state and public elections in India. There have been previous allegations of EVMs security. Following the judgments of the Delhi Supreme Court, the Supreme Court and the demands of various political parties, the Election Commission decided to introduce the EVM system with the voter-verified paper audit trail system.

Figure 4 BVM being used in India

EVM consists of two modules, one control unit and the second is balloting unit. The units are connected by a five-meter cable. The voting unit facilitates voting by the voter through labeled buttons while the control unit controls the polling units, stores the number of votes and displays the results on 7-segment LED screens. The controller used in the EVMs digs its driver permanently in silicon at the time of manufacture by the manufacturer. No one can change the program once the console has been manufactured.

The control unit shall be with the responsible officer or polling officer and the polling unit shall be placed inside the voting booth. The polling station provides voters with blue buttons (temporary key) classified horizontally with the candidate party code and candidate names. On the other hand, the control unit provides the officer with the "ballot" to move to the next voter, rather than issuing a ballot paper. This activates the voting unit to vote from the next voter in the class. A voter must cast his vote by simply pressing the blue button on the voting unit against the candidate and his choice code.

2.3 Electronic Voting in Pakistan:

In Pakistan there are 2 model of machines in existence. One machine is more or less the copy of Voting machines being used for elections in India. The second machine that was seen and tested in polling stations is not actually a Biometric Voting Machine but only a Biometric Verification System connected with NADRA database.

It is an impressive Biometric Verification System but generally it should not be termed as Biometric Voting Machine. The Biometric Verification System was tested and many such machines were bought by Election Commission Pakistan (ECP).

Even after the existence and use of Biometric Verification system in NA-120, the use of this sort of system has been ruled out for the upcoming 2018 elections as the use of this system hasn't been as successful lately. These machines were unable to verify 12% of that single constituency voters, which in itself is a big failure and shows that this machine is not implementable in general elections. If 12% users are unable to register in an environment where all of the workforce of ECP was focused on only single constituency.

Figure 5 Biometric Verification System recently used in NA-120 elections

Moreover, investing heavily on such non customizable machines itself is a big risk to take in this rapidly developing technological warfare as it will outdate any such machine in only 2-3 decades at top.

Figure 6 BVM design by ECP that is inspired by Indian BVM

Election Commission has also worked on a voting machine based on Indian Electronic Voting machine. Such machines were tested but the plan to develop such technologically advanced ASIC (Application Specified Integrated Circuit) based voting machine was dropped as Pakistan don't have the technological advancement to bake the wafers and design the circuits at home, and importing such machines will cause a surge in price that Pakistan can't afford right now.

Chapter 3: Modeling and Simulation

3.1 Design Procedure

The old plan was to directly connect Raspberry Pi with the Fingerprint Scanner and considerable achievements were gained in the process. Later, the design was changed a bit and Arduino Uno was introduced in the process.

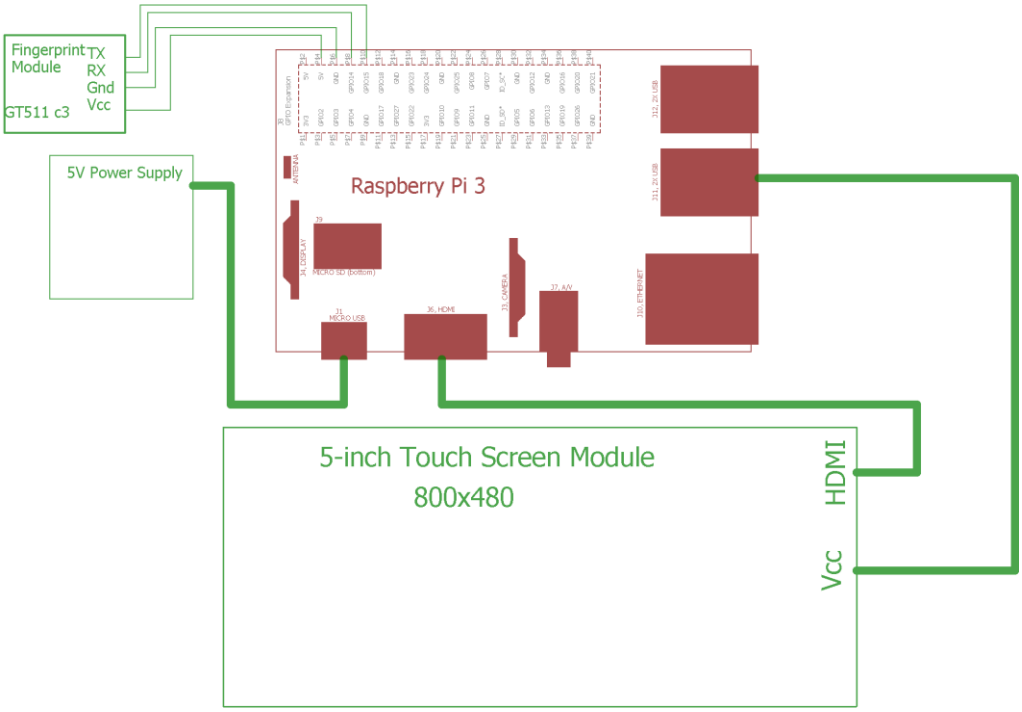


Figure 7 Old Circuit Diagram

Raspberry Pi was selected as the core because the base chip implemented language that was chosen was Python which is one of the most advanced high level language. All the new machine learning algorithm and deep learning is also being developed in Python that's why Python was preferred over other languages.

Raspberry Pi was preferred over Arduino as Raspberry provides more customization options and processing power and speed over simple on chip microcontroller systems. Raspberry Pi is the future, not just an on chip computer that it used to be. Today, its specifications make it a true computer. Work on it was done to enable its UART and connect it with the fingerprint scanner.

Though the serial communications with pi were established with Fingerprint module but later the plan was scrapped and Arduino was introduced for serial communication to neat up the things and for making the debugging easier.

The LCD was also upgraded later to 7-inch touch screen as the 5-inch touchscreen was not considered suitable because of its low resolution and small size.

3.1.1 Enabling UART

There are two Serial channels

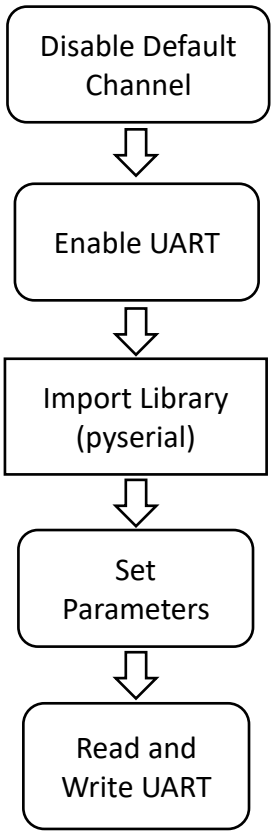
- Serial0 → ttys0
- Serial1 → ttyAMA0

Firstly, the Bluetooth is the default serial communication protocol, for our UART communications to be enabled, we have to disable the Bluetooth serial communications and enable the UART communications.

After enabling the UART, first the UART console is to be disabled for it to send and receive data properly. Afterwards, the "pyserial" library is imported and afterwards all the following parameters of serial channel are set:

- Device Selection
- Baud rate
- Data bits
- Parity
- Timeout
- Stop bit

Flowchart:



3.2 Hardware

Hardware for this project is roughly 40% of the total project. After developing the software, most of the time consumed was by the hardware integration and creating the hardware viable.

The hardware constitutes of:

- Raspberry Pi 3
- Arduino UNO
- 7-inch Capacitive LCD Touch screen
- Fingerprint Module GT 511 C3
- TTL module
- Body

3.2.1 Raspberry Pi 3:

The Raspberry Pi 3 is exactly what one expect from the latest Raspberry Pi. No, it does not have a SATA, USB C or PCIe connector. Raspberry Pi's goal was to produce a cheap computer for everyone, and adding these ports would rise the expense. Instead of satisfying energy users, Pi Foundation made every effort to please everyone. Like the Raspberry Pi 2 from the end of last year, the Raspberry Pi 3 features a new processor, the Broadcom BCM2837 64-bit ARM Cortex A53, running at 1.2 GHz.

The most important feature of the pre-launch leak surrounding the Raspberry Pi 3 is an additional wireless feature, the important news is the updated processor. Using Cortex A53, Pi 3 passed the threshold. The Raspberry Pi is not just a card used to play retro video games in simulators, and is no longer a simple decoder. The Pi 3 is a real on chip computer.

Raspberry Pie 3 is the brains of this whole system. With its 4 USB Ports, 1 GB RAM, HDMI Port, Composite Output Jack, Ethernet Out Port, 40 GPIO ports, built in Wi-Fi and Bluetooth, it is marvelous and underappreciated silicon on chip computer. Raspberry Pi is the future, it comes in different sizes and specifications and its specifications are improving each and every day. When we started the project, “Raspberry Pi 3B” was the most powerful version available but before the end of this project, it already got an upgrade in form of “Raspberry Pi B+”.

Figure 8 Raspberry Pi 3B

It was preferred on other microcontrollers because of its open ended nature (i.e. multiple solutions to future problems that may arise), it is easy to use, Python is its

core language. It can easily be integrated with multiple high tech equipment that comes in almost plug and play configuration (i.e. Camera, LED touch screen etc.). Python being one of the most powerful high level programming language fills Raspberry Pi with unlimited capabilities.

Briefly the Raspberry Pi was preferred over other components because there are many researchers extensively working on Raspberry Pi3 due to which huge amount of resources and libraries are already available & also because of its specifications and easily customizable, upgradable nature, i.e. for PIC the touch screen LCD and connection with Wi-Fi dongle is possible but it will require extra effort and coding, it won't be customizable (i.e. if an extra feature like camera, QR scan etc. are needed in future).

3.2.2 Arduino Uno

Arduino Uno is yet another powerful tool that here is being used to connect and interact with Fingerprint module effectively, process the data and send the result to the main brain of the project, Raspberry Pi 3.

Arduino UNO is a widely used, popular and marvelous chip based system also widely speculated as microcontroller. Its specialty is that it is open source that is based on ATmega328P microcontroller. It is developed by an institute known as Arduino.cc and is officially available by them, tough many unofficial versions are also available in the market.

The board is equipped with digital and analog I/O pins that can be interfaced with different expansion boards and other circuits. The board contains 14 digital pins and 6 analog pins. It is programmable with the Arduino IDE through a USB cable. It can be powered by a USB cable or an external 9-volt battery, it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo.

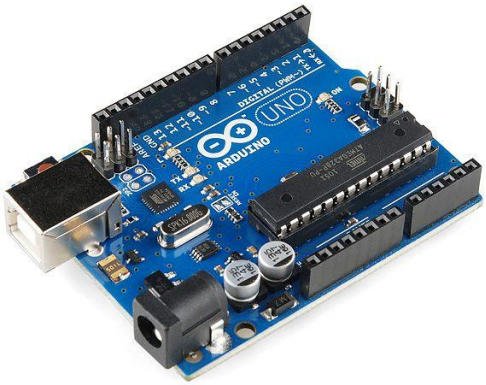


Figure 9 An Arduino UNO Model

Layout and production files for some hardware versions are available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. Uno's advice and Arduino Software's (IDE) version 1.0 were the reference versions of Arduino. They are now evolved to newer versions. The Uno card is the first in a series of Arduino USB cards. The ATmega328 on the Arduino Uno is preprogrammed with a bootloader that allows you to download new code without the use of an external hardware programmer. It communicates using the original STK500 protocol. Uno also differs from all previous cards in that it does not use the FTDI USB to Serial

driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to R2 version) programmed as a USB-serial converter.

3.2.3 GT-511-C3 Fingerprint Module

The GT-511C3 FPS (Fingerprint Scanner) is a small integrated module that consists of an optical sensor mounted on a small circuit board. The optical sensor scans a fingerprint and the microcontroller and software provide module functionality that automatically processes the scanned fingerprint.

The FPS interface is very basic consisting of only four pins - Vcc, Ground, serial transmitter and serial receiver. The module can be used in user identification projects, such as allowing access through a door by scanning a finger instead of using a key.

A GT-511 C3 is the fingerprint scanner that will be used in the project. It has its own flash storage and processor. It is a powerful machine that use UART with TX and RX pins of any microcontroller and support features such as enroll, match, delete etc. (See Appendix C for further info)

Figure 10 A GT 511C3 Fingerprint Module

3.2.4 TTL Module

A USB to TTL UART converter module is used with laptops which lacks standard serial ports. In this project, TTL module was used to connect and test the fingerprint module with laptop.

This module can be used with laptops that do not have a standard serial port. This module creates a virtual port using USB on the computer that can support various standard baud rates for serial communication. You just need to install the driver using automatic setup that installs the correct driver files for OS. After installing the driver, plug the module into any USB port of your PC. Finally, a new COM port is made available to the PC.

3.2.5 Touchscreen LCD

The LCD (Liquid Crystal Display) is the visual output device. The LCD being used in this project is a touch based LCD hence acts both as Input and Output. There are two types of touch LCDs:

- Resistive Touch LCDs
- Capacitive touch LCDs

Figure 11 A 7 inch multi touch touchscreen module

The LCD being used in this project is capacitive touch screen that supports multi touch. A 7-inch capacitive touch sensor based colored touch screen is the bread and butter of this project. The touch screen will be used to display the hard work in form of Graphical User Interface (GUI) as well as take the input in form of touch sensor.

3.3 Software

The Software constitutes 60% of the whole project, wide array of soft wares and many programming languages were learned to make this project a success.

The primary Soft wares used are:

- Microsoft Visual Studio (For C++)
- Python Programming Environment
- Tkinter (GUI library)
- MS Office
- LibreOffice
- Microsoft SQL Server

3.3.1 Microsoft Visual Studio:

Microsoft Visual studio is the coding powerhouse of Microsoft. Initially, the coding was written in C++ language and the compiler used for this purpose was Microsoft Visual Studio 2014.

Microsoft Visual Studio has three main parts:

- Code Editor
- Debugger
- Designer

3.3.2 Python

Python has a dynamic type and memory management system. It maintenances multiple programming models, including OOP, deterministic, functional, GUI & procedural, and has a comprehensive standard library. Many python interpreters are available for different operating systems. CPython, a commonly used interpreter, is an open source program.

Python is the language carrying the whole program and coding environment. The code, after being written in C++ was later converted, modified, remodified, re remodified and built in python language.

The Python's basic philosophy is summarized as:

- **Beautiful is better than ugly**
- **Explicit is better than implicit**
- **Simple is better than complex**
- **Complex is better than complicated**
- **Readability counts**

3.3.3 Tkinter (GUI-Python)

Tkinter is only a library in python but it surely has earned its place here. This powerful library is used to convert long lines of codes into Graphical User Interface and turns black lines into a colorful, pictorial and user friendly canvas.

The Tkinter module is the standard Python interface of the Tk GUI toolbox developed by "Scriptics". Tk and Tkinter are both available on most Unix platforms, as well as on Windows and Macintosh systems.

Tkinter consists of a number of modules. The Tk interface is provided by a binary extension module named **"_tkinter"**. This module contains Tk's low-level module.

The public interface is provided via a number of Python modules. The most important interface module is the Tkinter module itself. To use Tkinter, all you need to do is import the Tkinter module:

"import Tkinter"

Or:

"from Tkinter import *"

3.3.4 MS WORD

MS World was used for the documentation and in creation of the thesis while MS Excel was used in database creation and other such matters.

3.3.5 Microsoft SQL Server 2008

SQL server is a database tool devised by Microsoft. Extensive work has been done on this software for database creation and crafting purposes.

Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft. It's a highly scalable product that we run on any laptop, we have a high-powered cloud server network, and everything in between.

It is one of the most popular database management systems in the world.

SQL Server has evolved into a true enterprise information platform. SQL Server 2016 includes integrated business intelligence tools, and a range of analysis and reporting tools. This is in addition to database management tools such as database creation, backup, replication, security, and so on.

3.4 Design Details

3.4.1 Coding

Coding is something that was done throughout this project. It holds the primary position in this whole project. Initially the Pseudo code of this whole project was designed. The code has been modified till the last day of the project with certain tweaks for improving logic and performance.

The Code has been written in following languages:

- **C++**
- **Python**

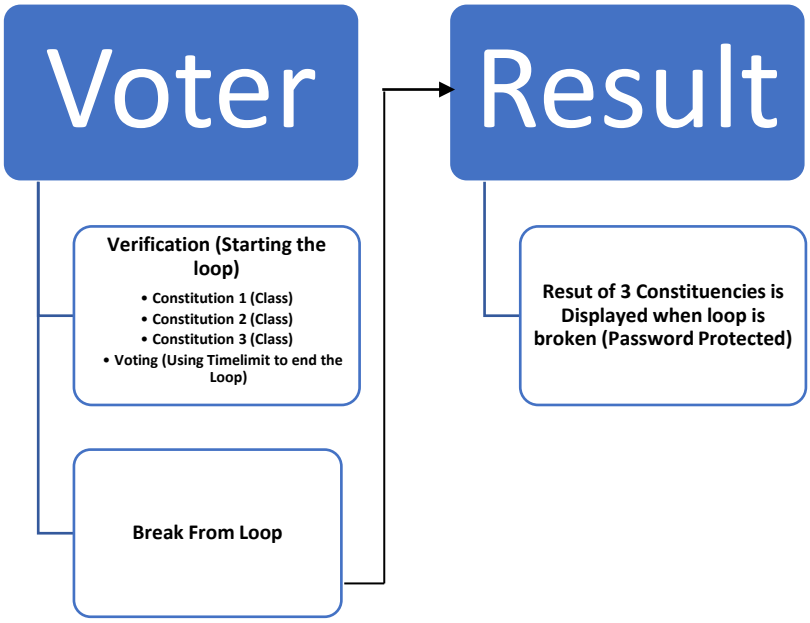


Table 2 Voting Pseudo Chart

Firstly, the logic was developed and whole program was written in C++ and tested, after which the program was converted in Python and recently the conversion has been completed.

The basic program in python has been created that takes the voter to the constitution of his choice (Later will take voter directly to his assigned constitution after fingerprint verification). After this the process of voting takes place and loop continues. Loop expires after pressing 1 (The loop will later automatically expire after voting session will expire).

When the voting process is expired, program requires a code by admin after which the results of all 3 constituencies are displayed.

Another logic is also in final stage of development in which instead of creating separate classes, a single parent class will hold the data and the subclasses will import parent's class data. This logic is just like the previous logic but is more advanced and will shorten the length of coding.

The current issue with its implementation is in declaration of result as the result of all three constituencies adds up in one variable.

3.4.2 Database

Initially work on 2 types of databases was carried out, the databases are following:

- Array Based Database
- SQL Server Based Database

3.4.2.1 Array Based Database:

```
import random
Name=['0','Muhammad Osama Shahzad', 'Ahtesham Zafar', 'Abdul Moiz', 'Engr. Ali Raza']
CNIC=['0','32304-6181218-7', '2', '3', '4']
Constituency=['0','1','1','2','3']
```

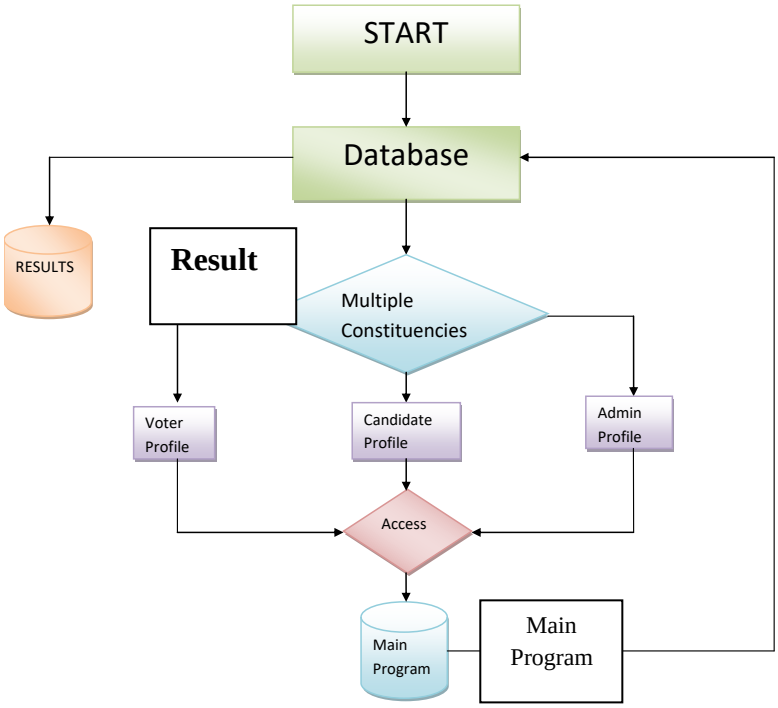
3.4.2.2 SQL Server Database:

Database is made by using SQL Server 2008. Database has important role because it contains all the information. Important Characteristics are added that are essential to recognize a voter and candidate.
Admin profile contains data of admin that can access the voting machine.
Database consists of:

1) Voter profile:
   - Contains all the personal information of voter.
   - Every voter assigned a unique voter-ID (primary key).
   - Contains profile picture of every voter.

2) Candidate profile:
   - Contains information of candidate.
   - Candidate set as foreign key for multiple accesses.
   - Party name with symbol is included.

Flow Chart:



3.4.3 GUI (Graphical User Interface)

Tkinter is a python library that is imported to work with Graphical User Interface. Then main function is defined, in which we define class and initialize it. In class, we define objects, title our window, specify its size, and define and pack all buttons and label we are using.

Basically in tkinter the program is created where the shapes are created and defined and its logics are also interconnected with created GUI. (i.e the input is taken not from key but by clicking on a circular button on screen which is created using this tkinter library)

Chapter 4: Results & Discussion

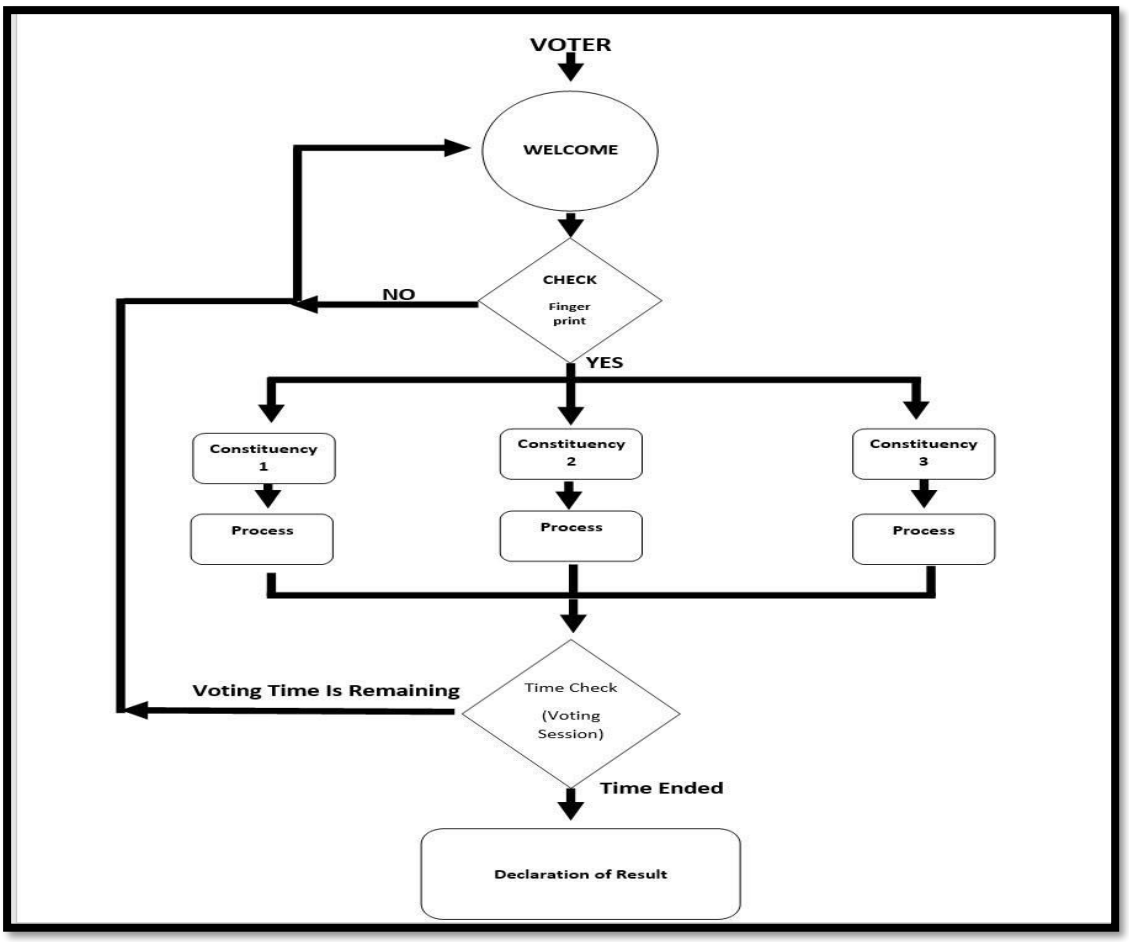
4.1 Python Base Program: (See Appendix for code example)

Python base code is the code that is running the whole system.

As explained in the table below, the code is following some multiple hierarchy OOP concepts of class and subclass.

The voter comes and has to place his fingerprint after he is shown "Welcome Screen". After the welcome screen, the voter has to place the fingerprint. The fingerprint module compares the fingerprint and sends acknowledgement to Arduino which sends it to Raspberry Pi and hence this way it is accessed by main code.

Table 3 Voting Flowchart



There after getting the acknowledgement, code accessed the database and before going to main database, it passes through "Dual Voting filter" that

checks if the person has already voted, if voted then the program stops and if not then the data of the person, including Name, CNIC, Constituency etc. are shown and the person automatically access his own constituency candidates and thus cast a vote.

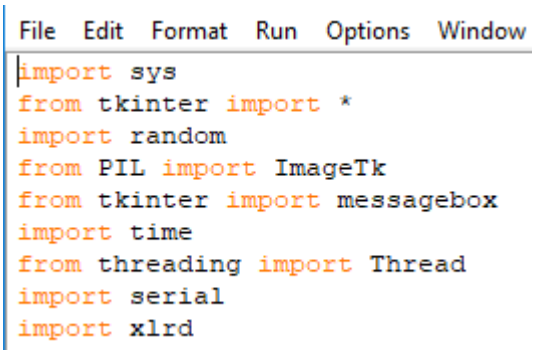
Similarly, multiple constituency Voting is carried out and when the time ends then after putting the pass key, the Presiding officer can end the voting session and declare result of particular constituency.

4.2 Graphical User Interface

Import Libraries & Defining main function:

On the top import all libraries we are using in program. Also import tool we are using for user interface in Python. In our case, we are using tkinter 3.0 as shown below on left hand side figure.

Our main program run from main function. Define a main function, create a variable using TK and pass it to class as argument as shown in right hand side figure.



Defining a class and designing a canvas:

Initialize a class and define a main function. In main function, create variable which accept a value from main. Within a function, create a CANVAS.

Canvas is a rectangular area intended for drawing pictures. In our case, Canvas use to show image in background stored in directory. Create a variable and store the value of canvas in it. Assign the location of directory in which picture you want to show as bg is store.

Specify the parameters and geometry of Canvas in x, y ,z axis.

```
class    Last_Window():

    def __init__(self,master):
        C = Canvas(master)
        C.pack(expand=True, fill=BOTH)
        imagel = ImageTk.PhotoImage(file="picll.gif")
        C.img=imagel
        C.create_image(0,0, anchor=NW, image=imagel)
```

After designing CANVAS, set the parameters of Window with Canvas in it. Name that window and set its geometry with root as master which Tk.

```
self.master=master
self.master.geometry('1280x720+0+0')
self.master.title('LAST WINDOW')
```

Creating a widgets and packing on CANVAS:

Widgets like labels ,buttons and image buttons are been use to open a function or to a message.

For any widgets to be use, first of all create its self widgets and pass arguments in it.Specify its geometry, colour and also pass CANVAS variable.

To use **button as a widget**, also write function command as argument which suit that button.Also specify the name of button.

To use **label as widgets**, write its geometry ,colour size and message you want to show on screen and pack it in x,y z axis.

```
self.master=master
self.master.geometry('1280x720+0+0')
self.master.title('LAST WINDOW')
localtime=time.asctime(time.localtime(time.time()))
self.labell=Label(C,font=('arial',24,'bold'),text="Thank You For your particiation..",fg='blue',anchor='w').pack(side="top", padx=4, pady=4)
self.button2=Button(C,font=('arial',18,'bold'),text="Retry",fg='green',command=self.ret).place(relx=0.9, rely=0.4,anchor=SE)
self.button2=Button(C,font=('arial',18,'bold'),text="Results",fg='blue',command=self.res).place(relx=0.6, rely=0.3,anchor=SE)

master.mainloop()
```

To show **image button as widgets**, create self widget and specify its argument.In argument, first specify Canvas variable. After that assign the location of image you want to show as button . Specify the area of button as image button.

```
self.button2=Button(cwgt,font=('arial',14,'bold'),text="pti",fg='blue',command=self.two)
photo2=ImageTk.PhotoImage(file="pti.jpg")
self.button2.config(image=photo2,width="216",height="135")
self.button2.place(relx=0.5, rely=0.5,anchor=CENTER)
cwgt.create_window(130,210, window=self.button2, anchor=CENTER)
```

Also specify the geometry of image button on main canvas.Specify its width and hight.With geometry and name , when button is clicked , there would be some function assosiated with that button which would be perform . So we must also write command function as argument in self widget.

Creating function and connection with button widget:

As button widget is not like label widget. There is some function associated with every button . On clicking that button, funtion associated would be perform .In self widgets, command function as argument must be written. So by clicking button , program would go to that function and

```
                #----------------------
def goto(self):
    root2=Toplevel(self.master)
    self.master.withdraw()
    myGUI=Confirmation(root2)

def myquit(self):
    self.master.destroy()
```

Complete that command. In above figure, when button assosiated with goto will be press, present window will be close and and next class with name Confirmation will be perform.
If myquit function to be perform, program will quit from operation…

CANVAS and Widgets in Welcome window:

In welcome window, canvas use to set flag image as background. 3 labels been use on top of window with blue color. Also 2 image buttons are use with **EXIT** and **ENTER** name. So exit and enter functions will be interface with both button. Name of window is welcome window written on top of screen.

Welcome Screen:

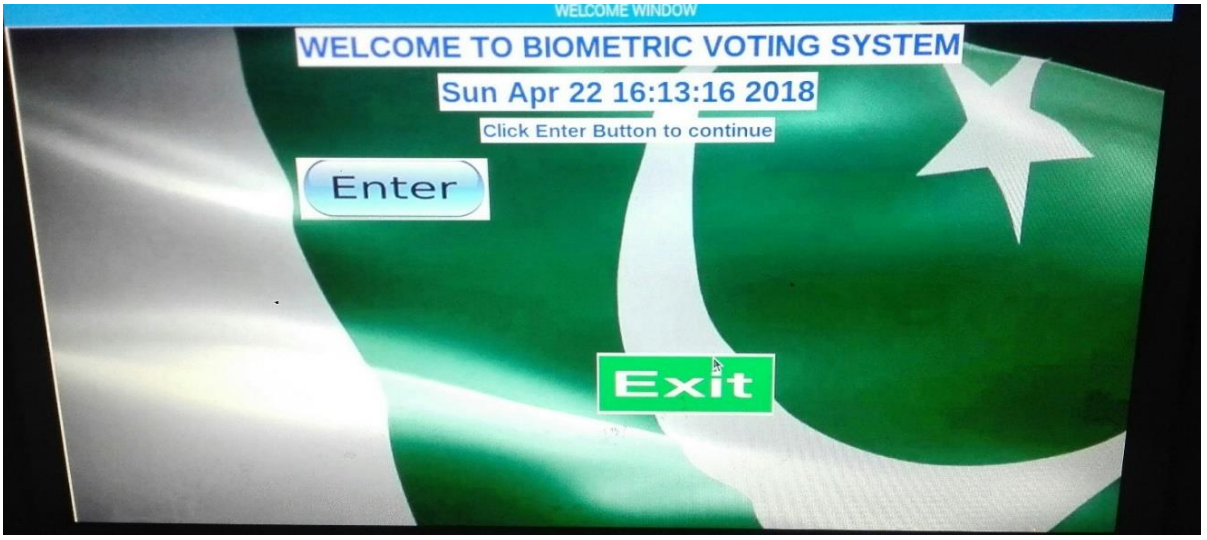


Figure 12 Welcome Screen

Starting the Voting Process:

When enter is pressed then the voting process starts and the machine shows the message on canvas to place the fingerprint on scanner for verification.

Figure 13 Canvas 2

Basic Voter's Data:

If the fingerprint is verified then the program access database and passes through the dual vote filter. Only if the vote of that particular id hasn't been casted before, then the information of the voter is shown on the LCD.

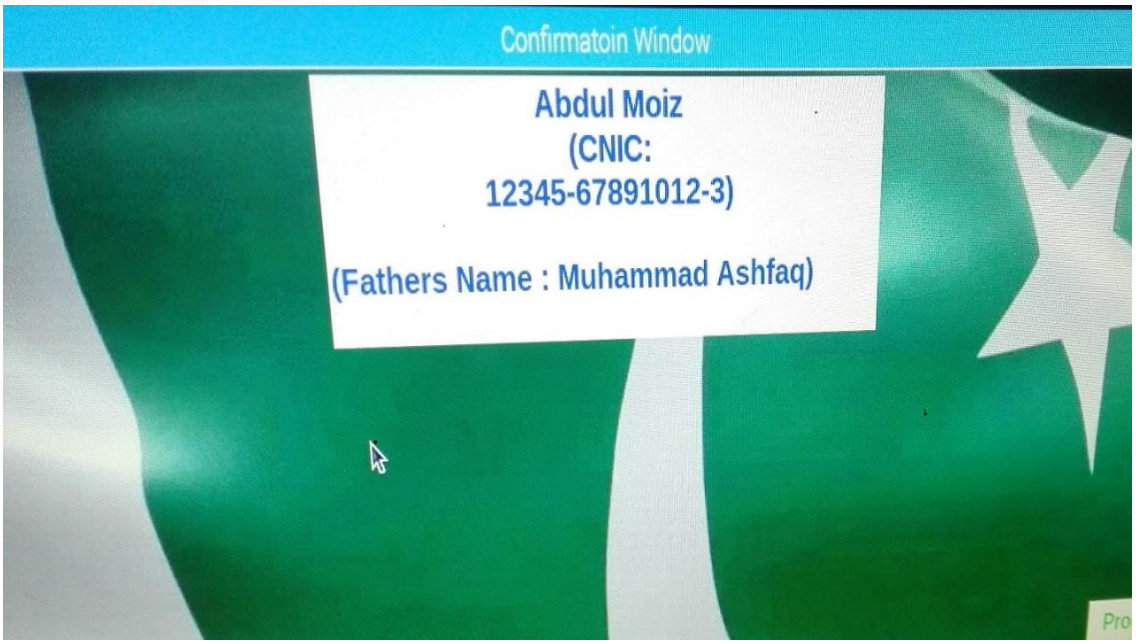


Figure 14 Canvas Displaying Voters Data

Voting Screen:

After the data is shown, the voter accepts the data and gets the access to voting screen from where the party of one's choice can be chosen for to be voted.

Figure 15 Vote Casting Canvas

Vote Casted:

After the selection of party, the vost is casted and Thankyou screen is shown.

Figure 16 Thankyou Screen

Dual Vote Filter:

If the fingerprint is recogonized but the vote on that fingerprint has already been casted then the following window is shown.

Figure 17 Dual Vote Filter Working

Unidentified ID:

If the fingerprint placed is not registered in the constituency then the follwinng window is shown:

Figure 18 Unidentified Fingerprint

Showing Result:

After the process of voting has been completed than the upon clicking the result button, the result window is shown which declares the result after automatic counting.

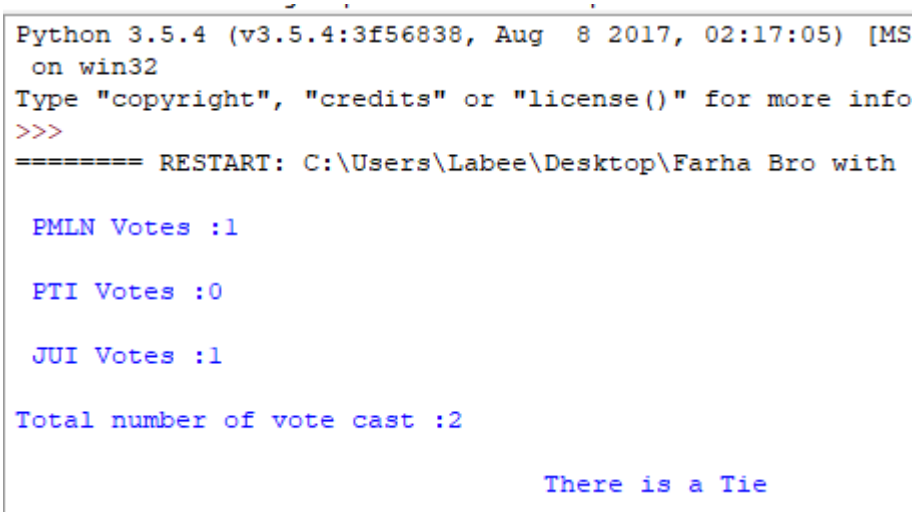


Figure 19 Result Declaration Screen

4.3 Arduino Code working

Code is divided into two sections.

Enrolling the Fingerprint and comparing the fingerprint.

511c3 is a fingerprint scanner connected serially with the controller (Arduino)

First we have to import libraries of serial communication and the fingerprint scanner 511c3

Arduino have 1 default UART pins numbered 0 for receiver and 1 for transmitter.

We need two UART pins because one UART pin for communication with Scanner and other UART pin for communication with Raspberry pi. To make Arduino I/O pins act as UART pins we need to import library for serial communication. The library used to make I/O pins act as UART pins is **SoftwareSerial.h**. We can make any two pins as UART pins.

The syntax used to make UART pins are as follows:

SoftwareSerial myserial (**RX** Pin, **TX** Pin)

Rx = Receiver Pin

Tx = Transmitter Pin

The other library we are importing is fingerprint scanner library **FPS_GT511C3.h**.

This library contains all the serial commands of the fingerprint scanner. Fingerprint scanner works with a command packet used to send by the controller and in response it sends the respond packet.

4.3.1 Enroll:

To enroll the fingerprint, first of all in void setup of Arduino we have to set the baud rate to communicate with scanner. The default baud rate is 9600. Then open command packet is sent to scanner to make it ready and to check its default state. To make the finger visible, turn ON the LED available in the Scanner. Now the scanner is ready to store fingerprint. Call the Enroll function

Then send enroll command packet to start enroll. 511c3 fingerprint scanner takes three times input to store the fingerprint. So, we have to send three times enroll command to store one fingerprint. After enrolling it will show "Enrolling successful".

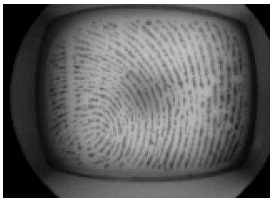


Figure 20 A captured fingerprint in image form

4.3.2 ID Recognition:

Fingerprint scanner stores the finger image in form of a template. When we put the finger on the scanner it generates a template and compares the template with the stored template. If the template matches the stored template it then generates the id i.e. in the form of integer.

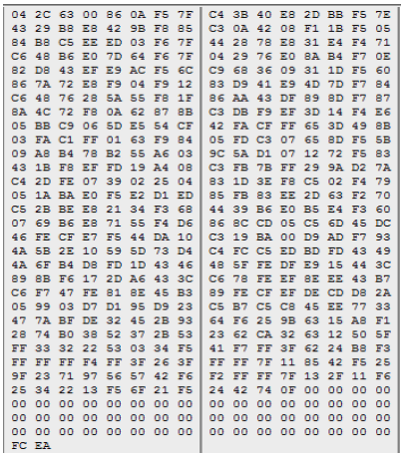


Figure 21 Fingerprint Templates

4.3.3 Interfacing

Fingerprint scanner 511c3 is a serial communication device. It has 4 four pins.

1st pin is transmitter and connected to the receiver of the Arduino at pin 8

2nd pin is receiver and connected to the transmitter of the Arduino at pin 7

3rd pin is ground and connected to ground of Arduino

4th pin is VCC 5v connected to VCC of Arduino

Command packet consist of – bytes. This command packet is sent by the controller through the transmitter. Transmitter is connected to the receiver of the scanner. Receiver receives it and performs the action. The transmitter of the scanner transmits the response packet to receiver of the Arduino.

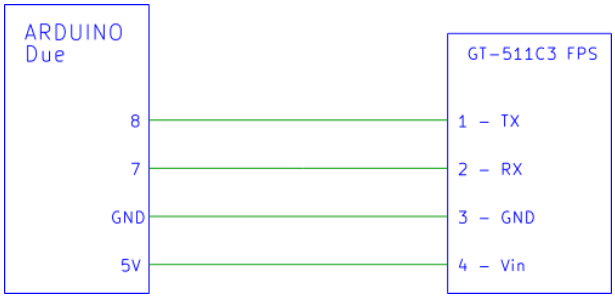


Figure 22 Arduino Connected with Fingerprint module

4.3.3.1 Problems faced:

I/O pins of the Arduino gives 5 volts and 0 volts when used as serial.

Fingerprint scanner 511c3 operates on 3.3 volts. So we need to reduce the volts by using voltage divider.

4.3.4 Arduino connection with Raspberry pi

Arduino connected to the fingerprint scanner compare and produce ID number. This ID number is then serially sent to the raspberry pi.

Raspberry pi also operates on 3.3 volts so we need to reduce the voltage by using voltage divider.

UART transmitter pin1 of Arduino is connected to receiver pin 10 of raspberry pi.

Ground of Arduino, scanner and raspberry pi is common.

4.4 LCD Interface:

The DSI display is designed to work with Raspberry Pi models that have mounting holes in a HAT footprint.

SI display auto detection is disabled by default on these boards. To enable detection, add the following line to:

/boot/config.txt:

ignore_lcd=0

Power the setup via the PWR IN micro-USB connector on the display board. Do not power the setup via the Pi's micro-USB port: the input polyfuse's maximum current rating will be exceeded as the display consumes approximately 400mA.

NB: With the display connected to the GPIO I2C pins, the GPU will assume control of the respective I2C bus. The host operating system should not access this I2C bus, as simultaneous use of the bus by both the GPU and Linux will result in sporadic crashes.

4.4.1 Screen Orientation

LCD displays have an optimum viewing angle, and depending on how the screen is mounted it may be necessary to change the orientation of the display to give the best results. By default, the Raspberry Pi display and Raspberry Pi are set up to work best when viewed from slightly above, for example on a desktop. If viewing from below, you can physically rotate the display, and then tell the system software to compensate by running the screen upside down.

To flip the display, add, anywhere in the file **\boot\config.txt**, the following line:

lcd_rotate=2

This will vertically flip the LCD and the touch screen, compensating for the physical orientation of the display.

4.5 Database:

Libre Excel is being used as the database finally after extensive testing of different databases.

LibreOffice is the default office of Raspberry Pi 3B hence it's database is easily connectable inside the Raspberry Pi. Its 2 sheets are being used. First sheet keeps all the data of voters of the constituencies while the second sheet is for dual voting check.

When one person votes, then the info is registered in second sheet so that the person may not vote twice.

Two libraries are being used to access, read/write on the worksheets of LibreExcel.

- Xlrd
- openpyxl

4.6 Block Diagram:

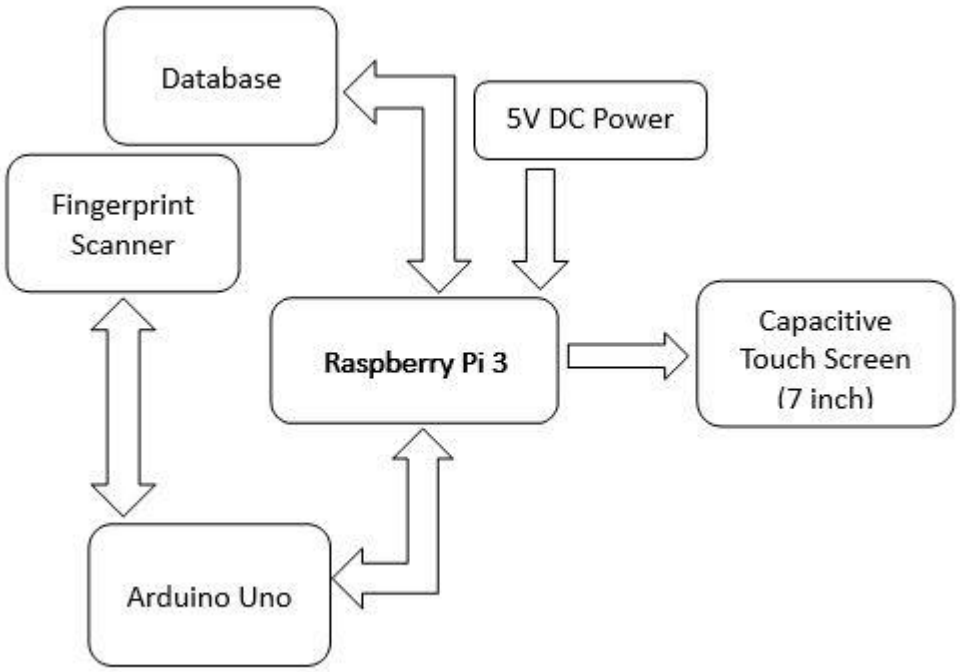


Figure 23 Block Diagram

Chapter 5: Conclusion and Future Recommendations

5.1 Conclusion

Summary of work done till now:

- Extensive work has been done on array based database, SQL Server database and libre excel database has been done and database has been created on libre excel.
- Fingerprint Scanner successfully integrated with Arduino Uno & Fingerprints being successfully stored and compared.
- Arduino successfully sending acknowledgements to Raspberry Pi 3.
- Raspberry Pi 3 successfully accessing data of the relevant voter from the database.
- Extensive and mature programming logic for the whole process of multiple constituencies in python has been created and tested.
- Touch screen LCD successfully integrated.
- The Programming logic has been converted into GUI environment.

The project is progressing rapidly in all the 3 dimensions; a strong foundation has been laid that will take this project to completion.

5.2 Future Development & Recommendations:

- Embed the whole coding into a proper app.
- Decentralize the program hence securing the voting process from hacking.
- Store the database online and connect multiple EVM with it to vote simultaneously.
- Make the system secure and self-aware by adding the neural networks and introducing machine learning scheme to counter rigging.
- Store the fingerprint template inside the Raspberry Pi and use the machine learning algorithm to compare the fingerprint templates.

References

[1] B. S. o. Justice, "Voting System Security and Reliability Risks," Newyork University School of Law.

[2] P. Wolf, "Introducing Biometric Technology in Elections," International Idea, Stockholm, 2017.

[3] M. S. W. B. Shahzad Memon, "Review ofFinger Print Sensing Technologies," in *IEEE*, 2008.

[4] G. a. J. Alaguvel.R, "Biometrics using Electronic Voting System with Embedded Security," in *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2013.

[5] O. A.-J. T. S. H. a. M. S. E. Mohammed Khasawneh, "A Biometric-Secure e-Voting System for Election," in *5th International Symposium on Mechatronics and its Applications (ISMA08)*, Aman, 2008.

CHECK U

ORIGINALITY REPORT

13%	13%	1%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDE

PRIMARY SOURCES

1	en.wikipedia.org	Internet Source
2	www.bravenewballot.org	Internet Source
3	startingelectronics.org	Internet Source
4	www.sunrom.com	Internet Source
5	Submitted to Al Quds University	Student Paper

Appendices

Appendix A

Python (Main Code)

```
def main():
 time=int(1)
 while(time==1):
  x=random.randrange(0,6)
  print(x)
  if (x==0 or x==1 or x==2 or x==3):
     print ('Good Morning'+Name[x])
     print ('Your CNIC Is: '+CNIC[x])
     print ('Your Constituency Is: '+Constituency[x])
     print('\nWelcome to Voting\n')
     if(Constituency[x]!='0'):
        cons=int(x)
        print (cons)
        if cons==1:
           print('Welcome to Political Voting Booth:')
           ##This is the Dual Vote key
           Constituency[x]='0';
           PElection.setData()
        else:
         if cons==2:
            print('Welcome to Favourite Actor Award Booth:')
```

```
            ##This is the Dual Vote key
            Constituency[x]='0';
            FavActor.setData()
        else:
            if cons==3:
                print('Welcome to Favourite Lab Engineer Award Booth:')
                ##This is the Dual Vote key
                Constituency[x]='0';
                FavLabEng.setData()
            else:
                print('\nNULL VOTE\n')
    else:
        ##Dual Vote Check
        if (Constituency[x]=='0'):
            print('Your vote has already been casted\n')
    time=int(input('Enter 1 to Continue Voting and 0 for Result\n'))
    if(time==0):
     result()
else:
     print ('Your Constituency isn\'t Registered here ')
```

FOR ENROLL (Arduino)

```
#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"
FPS_GT511C3 fps(8, 7); // (Arduino SS_RX = pin 4, Arduino SS_TX = pin 5)
void setup()
{
	Serial.begin(9600); //set up Arduino's hardware serial UART
	delay(100);
	fps.Open();	//send serial command to initialize fps
	fps.SetLED(true);	//turn on LED so fps can see fingerprint
	Enroll();	//begin enrolling fingerprint
}
void Enroll()
{
	// Enroll test
	// find open enroll id
	int enrollid = 0;
	bool usedid = true;
	while (usedid == true)
	{
		usedid = fps.CheckEnrolled(enrollid);
		if (usedid==true) enrollid++;
	}
	fps.EnrollStart(enrollid);
	// enroll
```

```
Serial.print("Press finger to Enroll #");
Serial.println(enrollid);
while(fps.IsPressFinger() == false) delay(100);
bool bret = fps.CaptureFinger(true);
int iret = 0;
if (bret != false)
{
    Serial.println("Remove finger");
    fps.Enroll1();
    while(fps.IsPressFinger() == true) delay(100);
    Serial.println("Press same finger again");
    while(fps.IsPressFinger() == false) delay(100);
    bret = fps.CaptureFinger(true);
    if (bret != false)
    {
        Serial.println("Remove finger");
        fps.Enroll2();
        while(fps.IsPressFinger() == true) delay(100);
        Serial.println("Press same finger yet again");
        while(fps.IsPressFinger() == false) delay(100);
        bret = fps.CaptureFinger(true);
        if (bret != false)
        {
            Serial.println("Remove finger");
            iret = fps.Enroll3();
            if (iret == 0)
            {
```

```
                                        Serial.println("Enrolling Successful");
                                }
                                else
                                {
                                        Serial.print("Enrolling Failed with error
code:");
                                        Serial.println(iret);
                                }
                        }
                        else Serial.println("Failed to capture third finger");
                }
                else Serial.println("Failed to capture second finger");
        }
        else Serial.println("Failed to capture first finger");
}
void loop()
{
  Enroll();
        delay(10000);
}
```

FOR COMPARE (Arduino)

```
#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"
FPS_GT511C3 fps(8, 7); // (Rx,Tx)
void setup()
{
	Serial.begin(9600); //set up Arduino's hardware serial UART
	delay(100);
	fps.Open();	//send serial command to initialize fps
	fps.SetLED(true);   //turn on LED so fps can see fingerprint
}
void loop()
{
	// Identify fingerprint test
	if (fps.IsPressFinger())
	{
		fps.CaptureFinger(false);
		int id = fps.Identify1_N();
		if (id <200)
		{//if the fingerprint matches, provide the matching template ID
			//Serial.print("Verified ID:");
			Serial.print(id);
		}
		else
		{//if unable to recognize
			Serial.print("E");
```

```
                }
        }
        else
        {
                //Serial.println("Please press finger");
        }
        delay(3000);
}
```

Python Serial Communication

```
import time
import serial
ser=serial.Serial('/dev/ttyAMA0', baudrate=9600,
parity=serial.PARITY_NONE,stopbits=serial.STOPBITS_ONE,
bytesize=serial.EIGHTBITS)
time.sleep(1)
while True:
   if ser.inWaiting() >0:
      data=ser.read()
      za=data.decode('utf-8')
      #print (data, end="")
      print(za, end=" ")
```

Python GUI (Main Code only)

```
import sys
from tkinter import *
import random
from PIL import ImageTk
from tkinter import messagebox
import time
from threading import Thread
import serial
cast_vote = 0
cast_vote1 = 0
cast_vote2 = 0
pmln_votes = 0
pti_votes= 0
jui_votes = 0
vote=0
class Welcome():
    def __init__(self,master):
        C = Canvas(master)
        C.pack(expand=True, fill=BOTH)
        image1 = ImageTk.PhotoImage(file="pic11.gif")
        C.img=image1
        C.create_image(0,0, anchor=NW, image=image1)
        self.master=master
        self.master.geometry('800x480+0+0')
        self.master.title('WELCOME WINDOW')
```

```
        localtime=time.asctime(time.localtime(time.time()))

        self.label1=Label(C,font=('arial',24,'bold'),text="WELCOME TO
BIOMETRIC VOTING SYSTEM",fg='blue',anchor='w').pack(side="top",
padx=4, pady=4)

self.label1=Label(C,font=('arial',24,'bold'),text=localtime,fg='blue',anchor='w')
.pack(side="top", padx=5, pady=4)

        self.label1=Label(C,font=('arial',16,'bold'),text="Click Enter Button to
continue",fg='blue',anchor='w' ).pack(side="top", padx=6, pady=4)

self.button2=Button(C,font=('arial',14,'bold'),text="Enter",fg='blue',command
= self.goto)

        photo1=ImageTk.PhotoImage(file="enter.jpg")

        self.button2.config(image=photo1,width="195",height="64")

        self.button2.place(relx=0.5, rely=0.5,anchor=CENTER)

        C.create_window(390,180, window=self.button2, anchor=CENTER)

        self.button3=Button(C,font=('arial',12,'bold'),text="Quit",fg='blue',
command= self.myquit)

        photo=ImageTk.PhotoImage(file="exit.jpg")

        self.button3.config(image=photo,width="190",height="70")

        self.button3.place(relx=0.5, rely=0.5,anchor=SE)

        C.create_window(800,450, window=self.button3, anchor=SE)

        master.mainloop()

    def goto(self):

        root2=Toplevel(self.master)

        self.master.withdraw()

        myGUI=Confirmation(root2)

    def myquit(self):

        self.master.destroy()
```

Appendix B

Timeline

Table 4 Timeline A

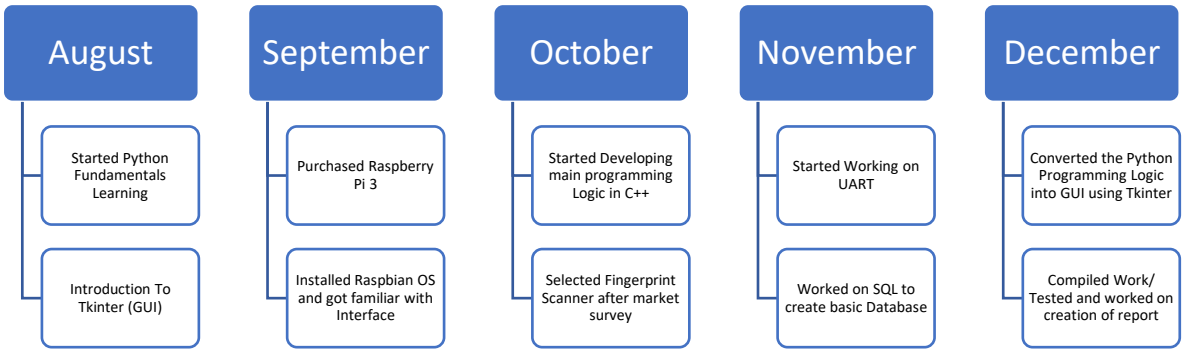
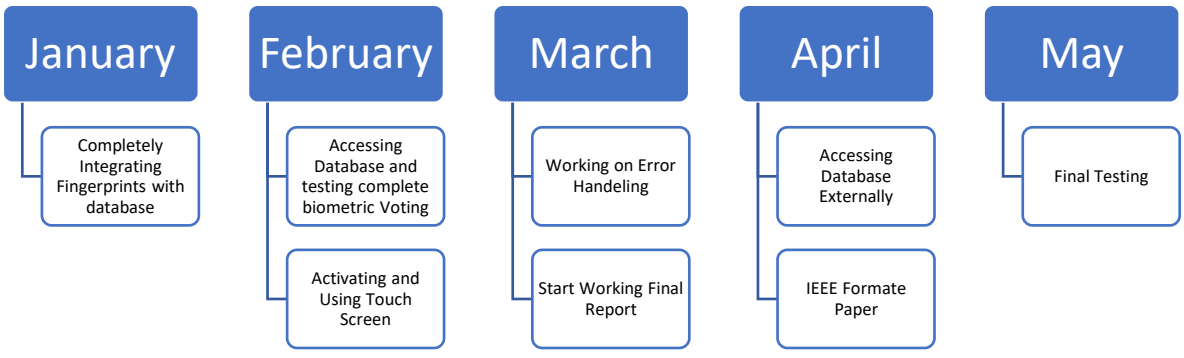


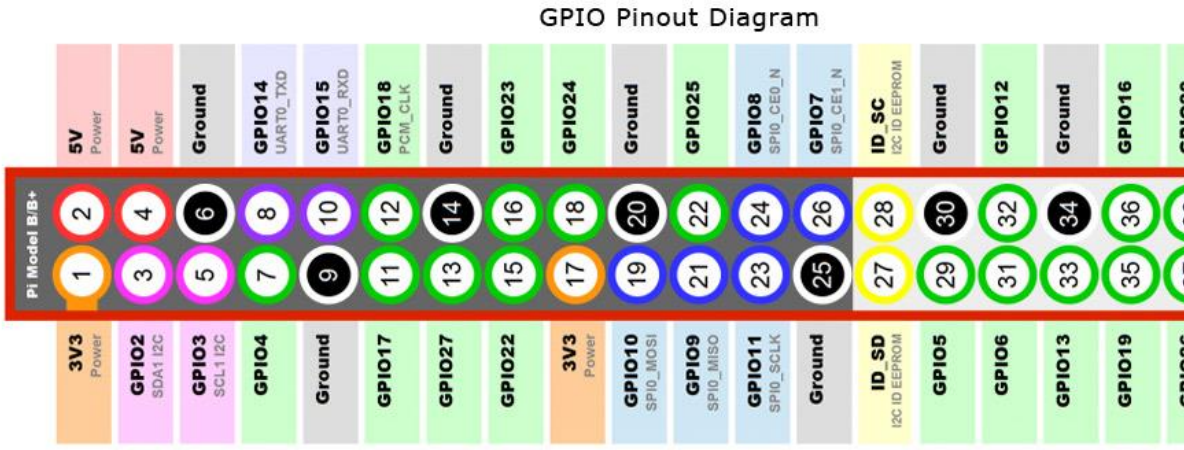
Table 5 Timeline B



Appendix C

Raspberry Pi 3 Datasheet

GPIO Pins:



Raspberry Pi

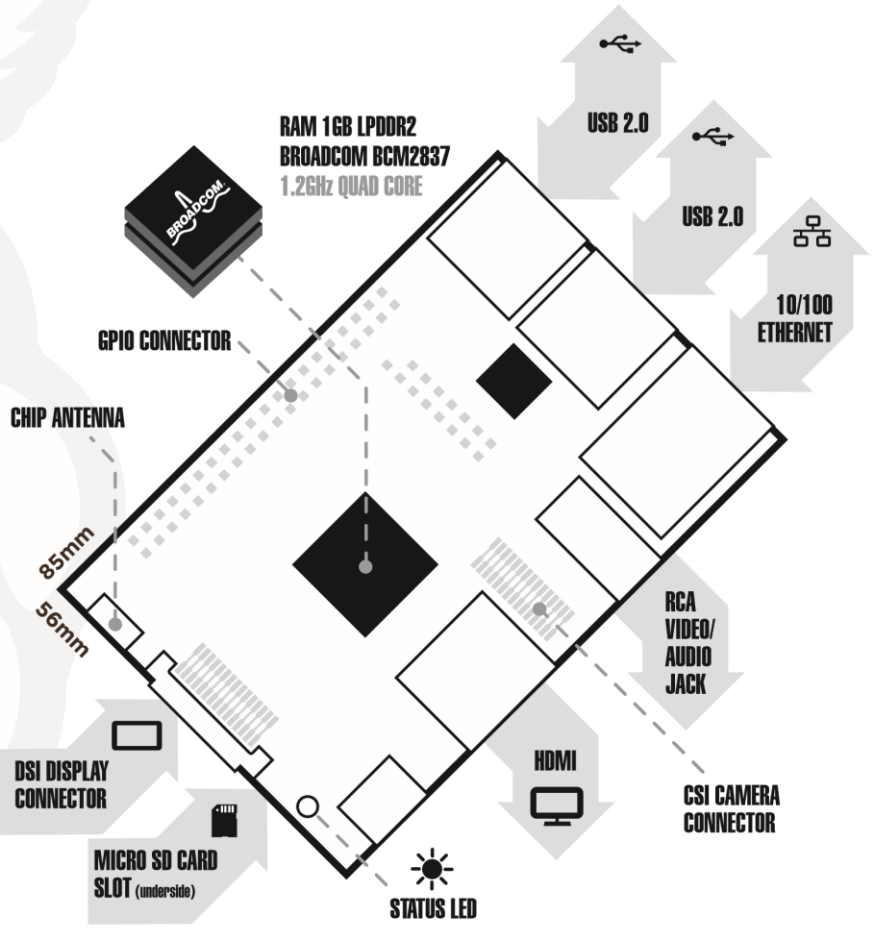
Raspberry Pi 3 Model B

Product Name	**Raspberry Pi 3**

Product Description

Raspberry Pi 3 model B brings you a more powerful processer, 10x faster than the first generation

RS Part Number	**896-8660**



RS

Raspberry Pi

Raspberry Pi. Additionally, it adds wireless LAN & bluetoth.

Raspberry Pi 3 Model B

Specifications

	Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture DMA
Memory	1GB 2
Operating System	Boots from Micro SD card, running a version of the Linux Windows 10
Dimensions	8 x 56 x
Power	Micro USB socket

Connectors:

Ethernet	1/100 BaseT Ethernet
Video Output	HDMI (rev 1.3 & Composite RCA (PAL
Audio Output	Audio Output 3.5mm jack, USB 4 x USB 2.0
GPIO Connector	40pin 2.54 mm (100 mil) expansion header: Providing 27 GPIO pins as well as +3.3 V, +5 V and
Camera Connector	15pin MIPI Camera Serial-
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector lanes and a clock
Memory Card Slot	Push/pull Micro

Key Benefits

- Low
- 1 x faster
- Consistent board
- Added

Key Applications

- Low cost
- Media
- Industrial/Home
- Print
- Web
- Wireless access
- IoT
- Roboti
- Server/cloud
- Security
- Gami
- Environmental sensing/monitoring (e.g. weather

Processor	Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0.

GT511C3 Datasheet

1. Concept

This device is one chip module with;

- fingerprint algorithm
- optical sensor

The major functions are the followings.

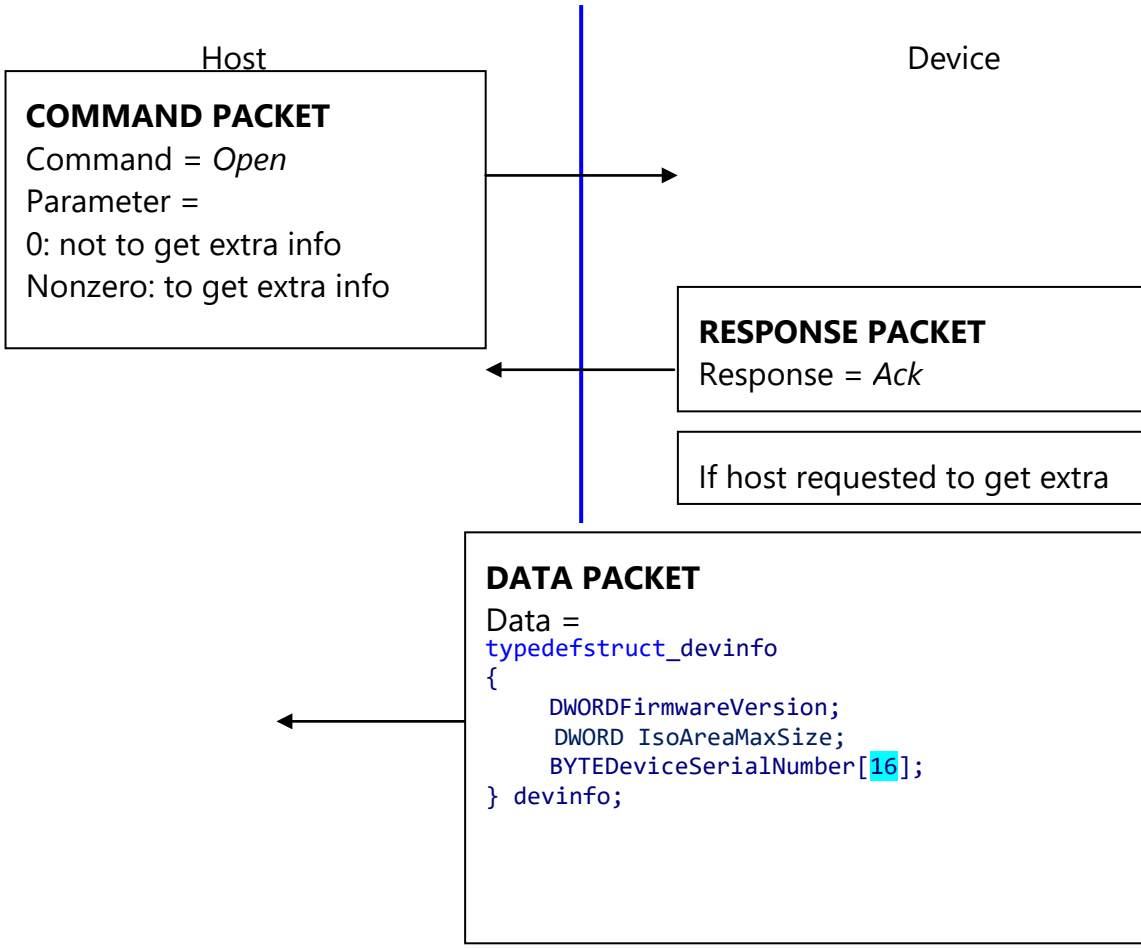
- High-accuracy and high-speed fingerprint identification technology
- Ultra-thin optical sensor
- 1:1 verification, 1:N identification
- downloading fingerprint image from the device
- Reading & writing fingerprint template(s) from/to the device
- Simple UART & USB communication protocol

Technical Specification

Item		Value
CPU		ARM Cortex M3 Core (Holtek HT32F2755)
Sensor		optical Sensor
Effective area of the Sensor		14 x 12.5(mm)
Image Size		202 x 258 Pixels
Resolution		450 dpi
The maximum number of fingerprints		200 fingerprints
Matching Mode		1:1, 1:N
The size of template		496 Bytes (template) + 2 Bytes (checksum)
Communication interface		UART, default baud rate = 9600bps after power on USB Ver1.1, Full speed
False Acceptance Rate (FAR)		< 0.001%
False Rejection Rate(FRR)		< 0.1%
Enrollment time		< 3 sec (3 fingerprints)
Identification time		< 1.0 sec (200 fingerprints)
Operating voltage		DC 3.3~6V
Operating current		< 130mA
Operating environment	Temperature	-20℃ ~ +60℃
	Humidity	20% ~ 80%

Protocol: Command Details

5.1. Initialization(*Open*)



Open command is used to initialize the device; especially it gets device's static info.

Protocol: Commands Summary

In a command packet *Command* can be one of below.

Number (HEX)	Alias	Description
01	*Open*	Initialization
02	*Close*	Termination
03	*UsbInternalCheck*	Check if the connected USB device is valid
04	*ChangeBaudrate*	Change UART baud rate
05	*SetIAPMode*	Enter IAP Mode In this mode, FW Upgrade is available
12	*CmosLed*	Control CMOS LED
20	*GetEnrollCount*	Get enrolled fingerprint count
21	*CheckEnrolled*	Check whether the specified ID is already enrolled
22	*EnrollStart*	Start an enrollment
23	*Enroll1*	Make 1st template for an enrollment
24	*Enroll2*	Make 2nd template for an enrollment
25	*Enroll3*	Make 3rd template for an enrollment, merge three templates into one template, save merged template to the database
26	*IsPressFinger*	Check if a finger is placed on the sensor
40	*DeleteID*	Delete the fingerprint with the specified ID
41	*DeleteAll*	Delete all fingerprints from the database
50	*Verify*	1:1 Verification of the capture fingerprint image with the specified ID
51	*Identify*	1:N Identification of the capture fingerprint image with the database

52	*VerifyTemplate*	1:1 Verification of a fingerprint template with the specified ID
53	*IdentifyTemplate*	1:N Identification of a fingerprint template with the database
60	*CaptureFinger*	Capture a fingerprint image(256x256) from the sensor
61	*MakeTemplate*	Make template for transmission
62	*GetImage*	Download the captured fingerprint image(256x256)
63	*GetRawImage*	Capture & Download raw fingerprint image(320x240)
70	*GetTemplate*	Download the template of the specified ID
71	*SetTemplate*	Upload the template of the specified ID
72	*GetDatabaseStart*	Start database download, **obsolete**
73	*GetDatabaseEnd*	End database download, **obsolete**
80	*UpgradeFirmware*	Not supported
81	*UpgradeISOCDImage*	Not supported
30	*Ack*	Acknowledge.
31	*Nack*	Non-acknowledge.

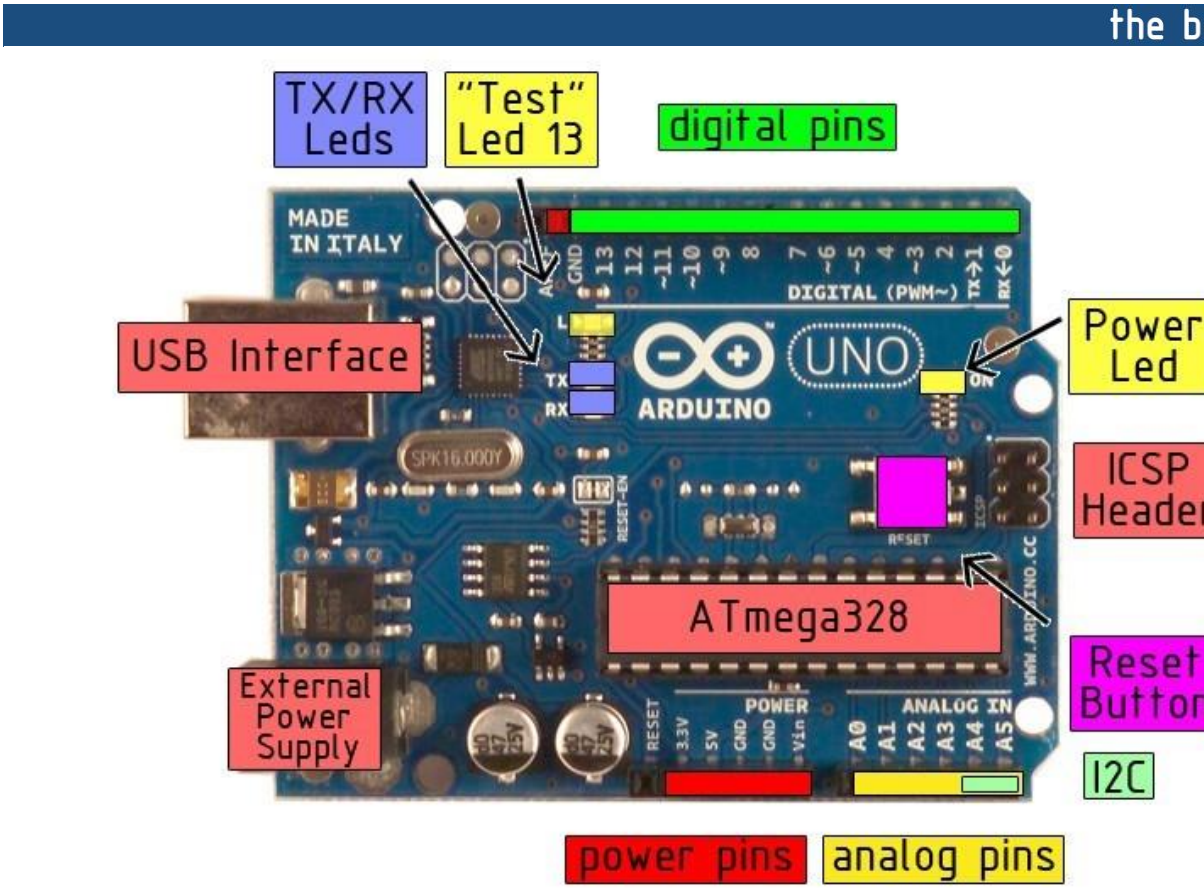
Arduino UNO Datasheet

Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Sum

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz



The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

Power

- **GND.** Ground pins.

Me

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Ou

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and Atmega328 ports.

Communica

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears

as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-toserial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

Program

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

Automatic (Software) R

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Prote

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteri

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.