INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY, SHIBPUR
(Formerly Bengal Engineering and Science University, Shibpur)

"FINGERPRINT BASED VOTING MACHINE USING ARDUINO"

Project report submitted
in partial fulfillment of the requirement for the Degree of

Bachelor of Technology
In
Information Technology

Academic Session: 2017-2021

Submitted by:

Soumyajay Das (510817001)
Anindya Kundu (510817020)
Shounak Chatterjee (510817022)

Project Guide:

Mr. Shyamalendu Kandar

DEPARTMENT OF INFORMATION TECHNOLOGY

DECEMBER 2019

ACKNOWLEDGEMENT

We the undersigned solemnly declare that the project report "Fingerprint Based Voting Machine using Arduino" is based on our own work carried out during the course of our study under the supervision of Mr. Shyamalendu Kandar. We assert the statements made and conclusions drawn are an outcome of our work. We further certify that the work contained in the report is original and has been done by me under the general supervision of our supervisor. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad. We have followed the guidelines provided by the university in writing the report. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

(Soumyajay Das)
ID 510817001
Dept. of Information Technology
IIEST Shibpur

(Anindya Kundu)
ID 510817020
Dept. of Information Technology
IIEST Shibpur

(Shounak Chatterjee)
ID 510817022
Dept. of Information Technology
IIEST Shibpur

December, 2019

CERTIFICATE

It is certified that the work contained in the project report titled “Fingerprint based Voting Machine using Arduino,” by “Soumyajay Das,” “Anindya Kundu,” and “Shounak Chatterjee,” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Shyamalendu Kandar)
Assistant Professor
Dept. of Information Technology
IIEST, Shibpur
December, 2019

CONTENTS

ABSTRACT

The aim of the project is to develop a proof of concept of a prototype for a Fingerprint based Electronic Voting Machine. The brains of the device relies on Arduino development board based on the ATmega328p architecture. The system consists of two units: Acknowledgement Unit, which is the master device, and a Voting Machine. The Acknowledgement Unit is connected to a Fingerprint Module which is used for verification. The other module is a memory unit which has EEPROM chips.

The system is a big improvement over normal EVMs which are prone to human malice in the election process. This system makes voting more secure, reliable, quicker, and accurate. Also, the parts are modular, so if a part malfunctions, it can be replaced.

However, it is noteworthy that this project is not a full proof design, but only a proof of concept. For a practical design, the prototype parts should be stripped down to a miniaturized small-footprint model. The design too has scope for further improvement. by the inclusion of a microprocessor, which would make it more communication secure, reliable, and robust.

I. Overview

1. Introduction

Elections are a defining feature of a democratic government. Political electoral systems are organized by governments, while non-political elections may take place in business, non-profit organizations, and informal organizations. There are several electoral systems including paper ballots, punch cards, and optical mark sense systems, EVMs.

The fingerprint based voting system is an electronic voting machine using biometric information. It is more efficient and quicker than paper ballot voting system, and it is more secure than ordinary EVMs. In the paper based system, votes are counted manually which makes it prone to error, such as duplicates and counting miss. EVMs solve this issue to a great extent, but without a proper verification, those systems can be used to rig votes more easily as they are quicker.

The use of a fingerprint based verification step solves the problem of vote rigging as only the fingerprint of an eligible voter will activate the device. Also it can prevent a voter from voting multiple times as attendance information for voters can be stored. Thus, this project aims at developing a model of one such system which intends to make the election process transparent, simpler, reliable, and quicker.

The authorisation can be done using the fingerprint information stored in every citizen's national identity card: *AADHAR*. The fingerprint templates per polling booth can be securely stored on a chip anonymously, and this chip is then fitted on the voting machine for that booth. This ensures that only legitimate voters can cast a vote.

2. Advantages of Fingerprint based Voting System

- **Secure**: prevents vote rigging
- **Accurate**: no miscounting of votes
- **Quick**: turnover time per voter is reduced
- **Simple**: lesser steps required in the process

3. Background

The concept was implemented using Arduino Nano, which is an open-source single-board microcontroller for building digital devices. It was programmed using the C/C++ language including some hardware libraries. The system also consists of a fingerprint sensor for verification, and EEPROM chips for offline data storage. The system consists of two units: an Acknowledgment Unit (AU), and a Voting Machine (VM). The chips and the fingerprint sensor are connected to the AU. The system loads the fingerprint templates, candidate numbers (per respective template), and list of candidate numbers with values initialized to zero, from EEPROM chips. The system verifies the fingerprint of a voter and checks whether it is valid, and if so, whether the voter has already cast a vote. The AU activates the VM after credibility check and then the voter casts a single vote. Then, the voter is marked as voted and the count for the particular candidate is incremented, and stored on the EEPROMs.

II. Components and Methods

1. Modes

The basis of the Fingerprint based Voting System (FVS) is “One Person — One Vote”. Thus, the main purpose of this system is “Prevention of Fraudulent Voting”. The system operates in 6 modes: *data loading*, *fingerprint verification*, *checking for validity of the voter*, *vote casting*, *information writing*, *ready*. These states are described below.

1.1 Data Loading

The fingerprint templates of *n* voters are previously stored anonymously from the database to an EEPROM chip. This chip is read-only. Another chip stores an entry with value ‘*0’* per voter, in the same order as the fingerprint per voter. When the Acknowledgement Unit starts, these fingerprint templates are read and written on the fingerprint sensor’s flash memory, with ID the same as the serial number. The sensor references these templates during the verification step. Also, the number of candidates is read from an EEPROM chip and conveyed to the Voting Machine to initialize the number of active buttons.

1.2 Fingerprint Verification

Before the voter can cast a vote, the voter needs to be authorized. The voter must be valid, i.e. the voter must be eligible and his records must be present in the particular polling booth where he appears. The polling staff enables the fingerprint sensor using a button on the Acknowledgement Unit. In this state, the fingerprint sensor is ready and repeatedly polling for fingerprints. The voter places his finger in the sensor; the sensor takes an image and converts it into a string. The string is then compared internally with the templates in its flash storage using a confidence algorithm to check for a match and the extent of the match. The result is then conveyed to the Arduino on the Acknowledgement Unit through serial communication. In case of an invalidity the result is conveyed by the use of hardware LEDs and an LCD present on the Acknowledgement Unit. Two of *4* status LEDs are used to indicate whether the voter is verified or not.

1.3 Checking for Validity of the Voter

Once a voter has been declared valid, a further check is made before eligibility to cast vote. If the voter has already voted, he isn't allowed another vote. As mentioned previously, an EEPROM chip stores an entry initialized to *0* for each voter. The value of the particular voter (identified by his fingerprint serial number) stored on the EEPROM is read by the Arduino and checked if it is a '*0*' or '*1*'. A '*1*' suggests that the voter has already cast his vote and he is not eligible for further voting. The result is then notified similarly as previous, using two LEDs and the LCD display on the Acknowledgement Unit.

1.4 Vote Casting

If the voter has been verified and eligible for voting, the polling staff activates the voting machine by pressing a button on the Acknowledgement Unit. A signal is sent to the Voting Machine asking it to activate itself. Once the Voting Machine is activated, an LED on the Voting Machine notifies that it has been activated. Then, the voter can cast 'one' vote by pressing a button for the respective candidate of preference. Once a button is pressed, an LED respective to the candidate is turned on and a buzzer sounds. The Arduino on the Voting Machine sends a signal to the Acknowledgement Unit with the candidate number voted for. The Voting Machine is then deactivated and initialized.

1.5 Information Writing

After the vote is casted the Arduino then reads the number of votes previously gathered for the particular candidate from the EEPROM chip, increments, and writes it back. Also, the entry for the candidate on the EEPROM is marked '*1*' to label him as have voted. The Arduino then notifies about the completion of the vote by printing on the LCD. It then follows itself to a halt state.

1.6 Ready

The devices can be '*reset*' at any point in the middle of a transaction which cancels the transaction, and initializes them. This is done by a button on the Acknowledgement Unit.

2. External Steps

The Voting System requires previous knowledge of the fingerprint information of voters, and the number of candidates. Also, the candidate numbers need to be correctly mapped to the real candidates after voting completes.

2.1 Preceding Step

Before the device is transferred to the polling booth, a secure database of the voters and candidates per polling booth needs to be created. The database stores information about the voters including their fingerprint templates and identification. It also stores the identification of the candidates. Then, the fingerprint templates of the voters are written on an EEPROM in the same order as that in the database for correct mapping. Also, an array of the candidates is written on another EEPROM in the same order with values initialized to '*0*', as mentioned earlier. This array can be accessed later to find out which of the voters have voted. Note that, the system does not store which candidate a voter has voted for, thus, maintaining anonymity. Similarly, the candidate count and entries initialized to '*0*' are written on the EEPROM in the same order as that appearing in the database. This array is modified when votes are cast, and later retrieved to get the count for each respective candidate.

2.2 Following Step

After the voting process has terminated, the polling information can be retrieved from the EEPROM chips. Two arrays are retrieved: the candidate array which gives the number of votes per candidate, and the voter array which notifies whether a voter has voted or not. The array entries are orderly mapped to the database entries to get correct data per person.

3. Flow Order

The step order is as follows:

1. Start Acknowledgment Unit and Voting Machine
2. Load Fingerprint Templates from EEPROM #1 into Fingerprint Module

3. Read the number of candidates from EEPROM #2 and initialize Voting Machine

4. Reset states

5. Scan fingerprint

6. Based on fingerprint,

- **a.** If fingerprint match not found
  - **i.** Display “Not Verified”
  - **ii.** Go to 12
- **b.** If fingerprint match found
  - **i.** Read voter mark status from EEPROM
  - **ii.** Based on voter status,
    - **1.** If voter status is ‘*1*’
      - **a.** Display “Already Voted”
      - **b.** Go to 12
    - **2.** If voter status is ‘*0*’
      - **a.** Display “Verified”
      - **b.** Go to 7

7. Activate Voting Machine

8. Cast Vote

9. Increment vote count for respective candidate in EEPROM #2

10. Mark voter as ‘*1*’ in EEPROM #2

11. Display “Vote Completed”

12. Enter ‘*Halt*’ state

The flow order above gives the order of happenings from switching on the devices till the termination of the first vote casting. However, the process is recurring. After termination or cancellation, when the device is in ‘*Halt*’ state, it is reset before the next vote casting and the system moves to step *#4*. Also, the transaction can be cancelled, as mentioned before, anywhere in the middle. The system then too, moves to step *#4*.

4. Components

The two devices in the system use electronic components like buttons, resistors, LEDs, wires, switches, etc. in addition to “Arduino Nano”*s*. The Arduino is programmed using the C/C++ language and offers several hardware libraries for added functionalities. The **Arduino IDE** is used to program the microcontroller. The full list of components are given below.

4.1 Hardware Component List

- Arduino Nano (ATmega328p)
- Fingerprint Module (R307)
- EEPROM ICs (24LC256)
- 16x2 LCD Display (WH1602B1)
- I^2C Module (PCF8574)
- LEDs
- Push Buttons
- On/Off Switch
- Piezo Speaker/Buzzer (KPX-G1203UB)
- Wires and 9V Batteries
- Resistors (220Ω: pull down, 1kΩ: pull up)

4.2 Component Description

4.2.1 Arduino

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Its products are licensed under the GNU General Public License (GPL). Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers can be programmed using C and C++ programming languages. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project. Arduino Nano is an Arduino board design having a small footprint and based on the ATmega328p IC architecture. The design of the Arduino Nano makes it sit perfectly on a standard breadboard with pins connected to adjacent strips.

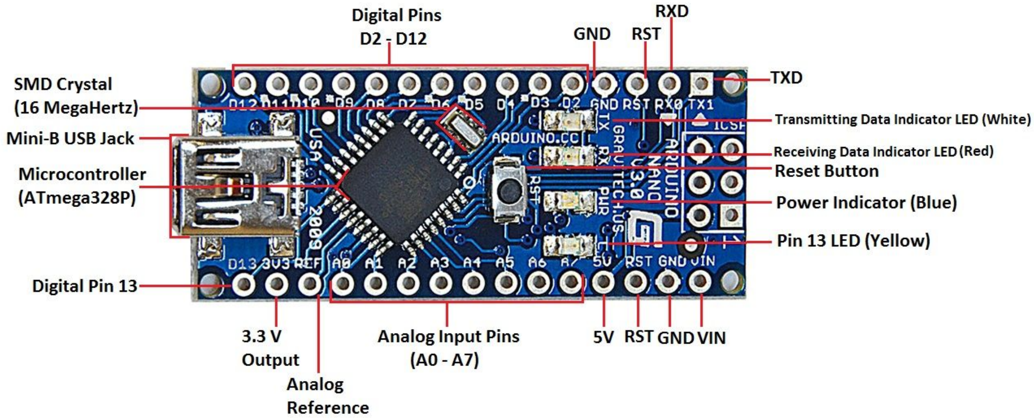


Fig. 4.1: Arduino Nano

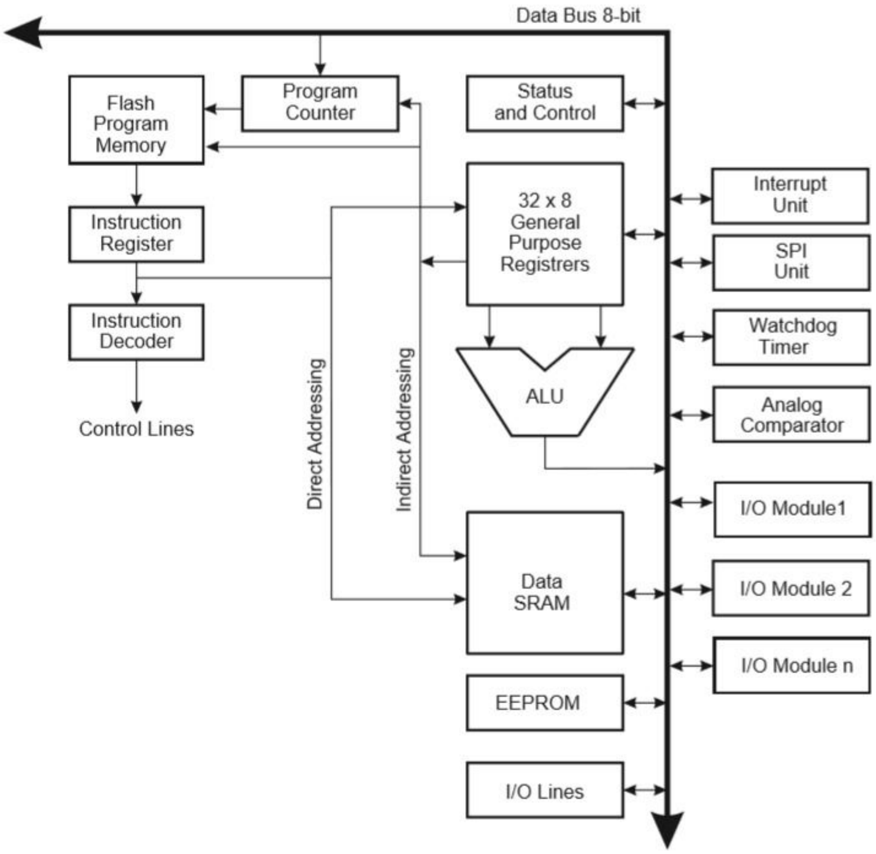
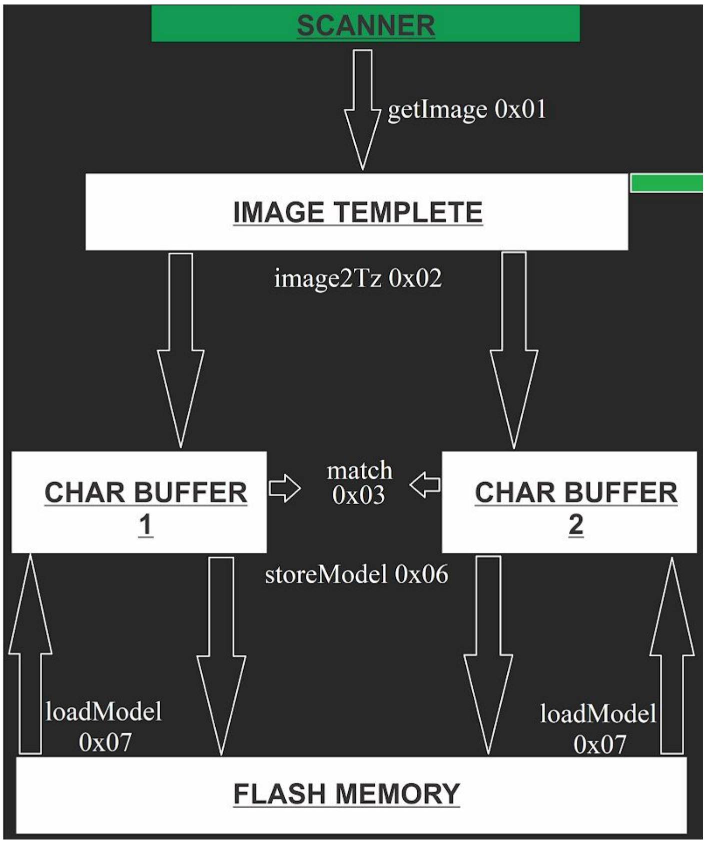


Fig. 4.2: Arduino ATmega328p architecture

4.2.2 R307 Fingerprint Module

R307 is a fingerprint identification module which takes Synochip DSP as the main processor and optical sensor. The module performs series of functions like fingerprint enrollment, image processing, fingerprint matching, searching and template storage. Fingerprint processing includes two parts: fingerprint enrollment and fingerprint matching (the matching can be 1:1 or 1:N). In addition, the R307 consists of a high-speed DSP, high-performance fingerprint alignment algorithm, high-capacity flash chips, to provide a stable performance. The R307 fingerprint module has two interfaces: TTL UART and USB2.0. USB2.0 interface can be connected to the computer; R232 interface is a TTL level.The default baud rate is 57600, and can be changed in reference to a communication protocol.



(a) (b)

Fig 4.3: (a) R307 Fingerprint module, (b) R307 architecture

4.2.3 EEPROM IC 24LC256

The Microchip Technology Inc. 24LC256 is a 32K x 8 (256 Kbit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.8V to 5.5V). It has been developed for advanced, low-power applications such as personal communications or data acquisition. This device also has a page write capability of up to 64 bytes of data. This device is capable of both random and sequential reads up to the 256K boundary. Functional address lines allow up to eight devices on the same bus, for up to 2 Mbit address space. This device is available in the standard 8-pin plastic DIP, SOIC, TSSOP, MSOP, DFN and 14-lead TSSOP packages. It communicates over I²C Serial Bus.

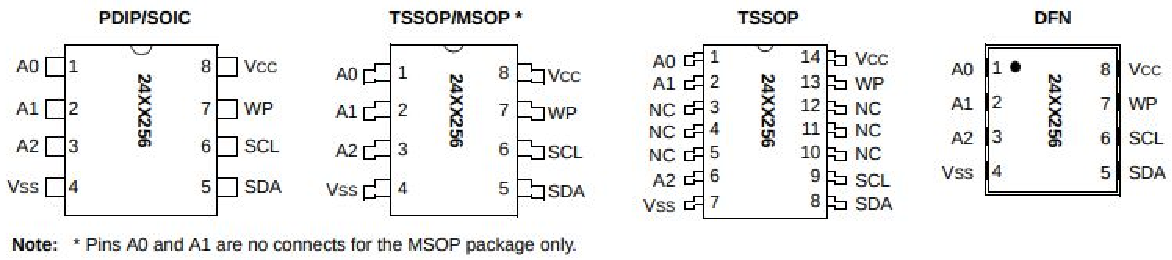


Fig 4.3: 24LC512 package types

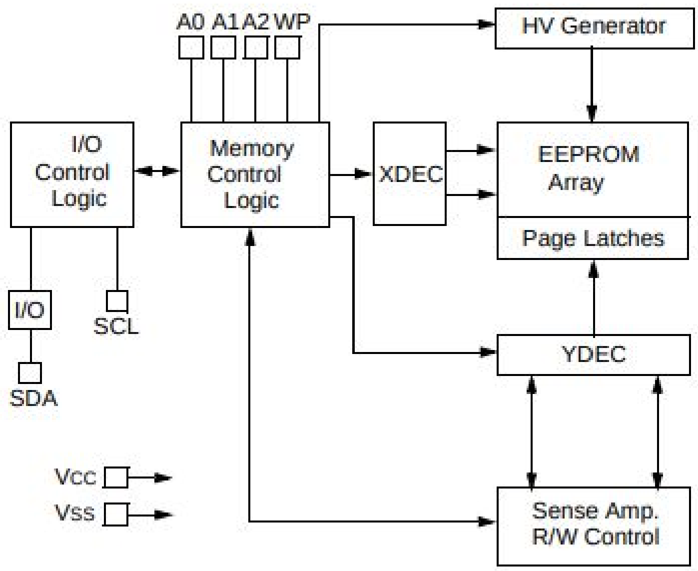


Fig 4.4: Block diagram

4.2.4 WH1602B1 16x2 LCD Display

The WH1602B1 LCD display is a 16-column, 2-row LCD device. Each character segment is composed of 5 columns and 8 rows of dots. It consists of a controller/com driver and a segment driver. The LCD should be instructed about the position of the characters and it achieves it using an MPU. It includes a set of registers each corresponding to a particular segment. The display module consists of 16 input pins having 8 data pins in addition to power, ground, LED power, LED ground, contrast pin, register select, read/write, and an enable.

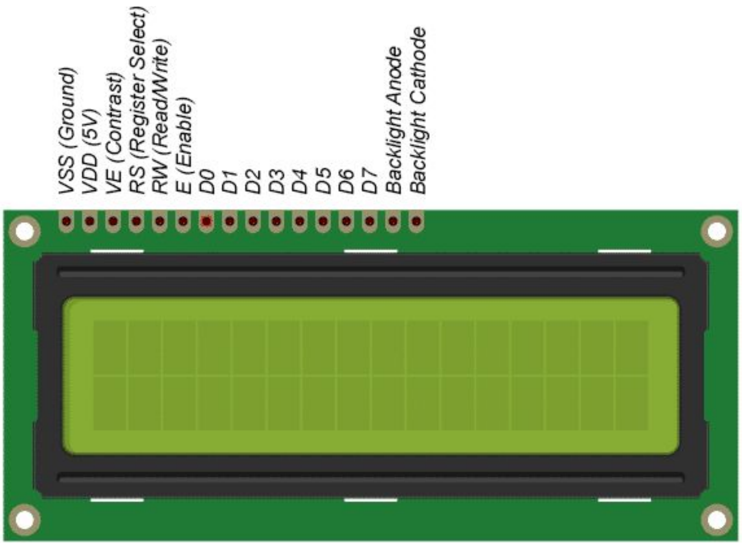


Fig 4.5: 16x2 LCD module

4.2.5 I²C Module

I²C (Inter-Integrated Circuit), is a synchronous, multi-master, multi-slave, packet switched, single-ended, serial computer bus. It is widely used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra-board communication. This module has an inbuilt PCF8574 IC that converts I²C serial data to parallel data for the LCD display. The PCF8574 device provides general-purpose remote I/O expansion for most microcontroller families by way of the I²C interface [serial clock (SCL), serial data (SDA)]. The device features an 8-bit quasi-bidirectional I/O port (P0–P7), including latched outputs with high current drive capability for directly driving LEDs. Each quasi-bidirectional I/O can be used as an input or output without the use of a data-direction control signal. At power on, the I/Os are high. In this mode, only a current source to VCC is active. The modules for LCDs come with a default I²C address of wither 0x27 or 0x3F. The module has a potentiometer for contrast adjustment.

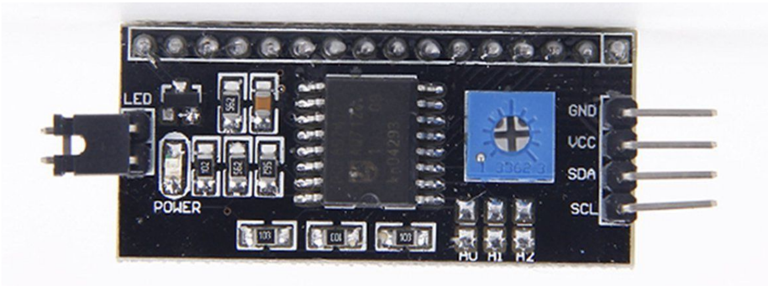


Fig 4.6: I²C module

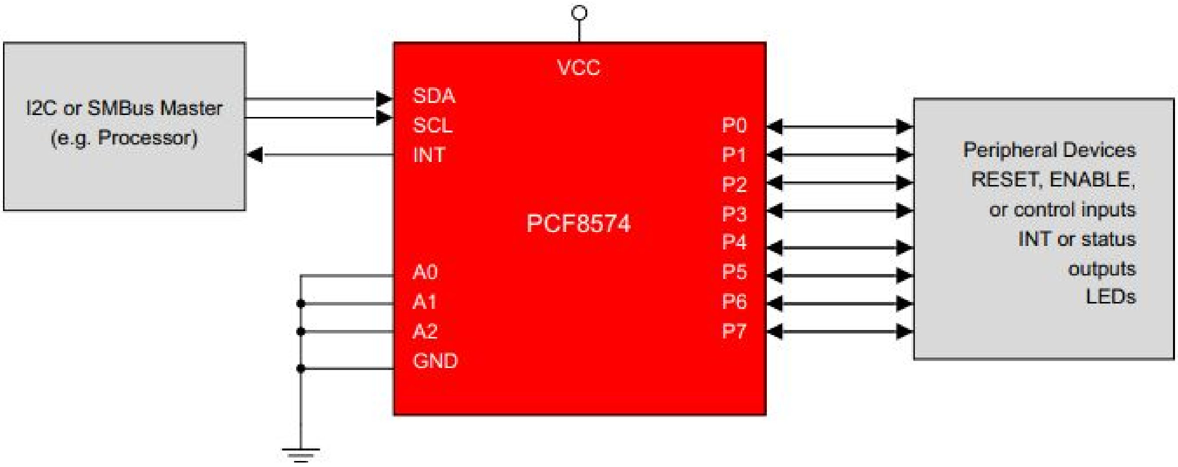


Fig 4.7: I²C Architecture

4.2.6 Push Buttons

The pushbutton is a component that connects two points in a circuit when we press it. When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton. When the button is closed (pressed), it makes a connection between its two legs.

4.2.7 KPX-G1203UB Piezo Speaker

A piezo speaker or buzzer is an audio signalling device. The KPX-G1203UB is a piezoelectric speaker and operates between 2V and 5V. The Min SPL is 85dB/3VDC/10cm and the Resonant Frequency is 2.3kHz±0.5kHz. It is a miniature size buzzer, ideal for small electronic circuits.

III. Design and Development

1. Setup

As rephrased previously, the system consists of two units: Acknowledgment Unit, and Voting Machine. All the major parts namely, the fingerprint module, EEPROMs and the I²C LCD module are connected to the Acknowledgement Unit. The Acknowledge primarily consists of an Arduino Nano in addition to these parts. The other parts include circuitry components like resistors, wires, LEDs, and buttons. The EEPROMs are connected by an abstraction of a memory module. Also, all I²C lines are joined together in an I²C module which contains the power circuitry as well.

1.1 Voting Machine

The Voting Machine consists of *n* buttons corresponding to *n* candidates. Its brain is an Arduino Nano board. To it are connected in addition to the buttons are, *n* LEDs corresponding to each button, a piezo buzzer, and another LED to notify if the device is active.

1.2 Control Unit

The Acknowledgement Unit is the master device for all others. Its brain too, as stated above, is an Arduino Nano board. The other circuitry elements include *4* LEDs to notify voter verification status, and *4* push buttons: *start verification*, *activate voting machine*, *reset*, and *display*. Note that, the display feature isn't a feature that should be present on final devices. It is used here just for demonstrating candidate votes. The Acknowledge Unit is stated previously is connected to a memory module and an I²C communication hub module, described next.

1.3 Memory Module

The memory module consists of two EEPROM ICs connected to common *Vcc*, *Gnd*, *SDA*, *SCL* lines. One of them is write protected.

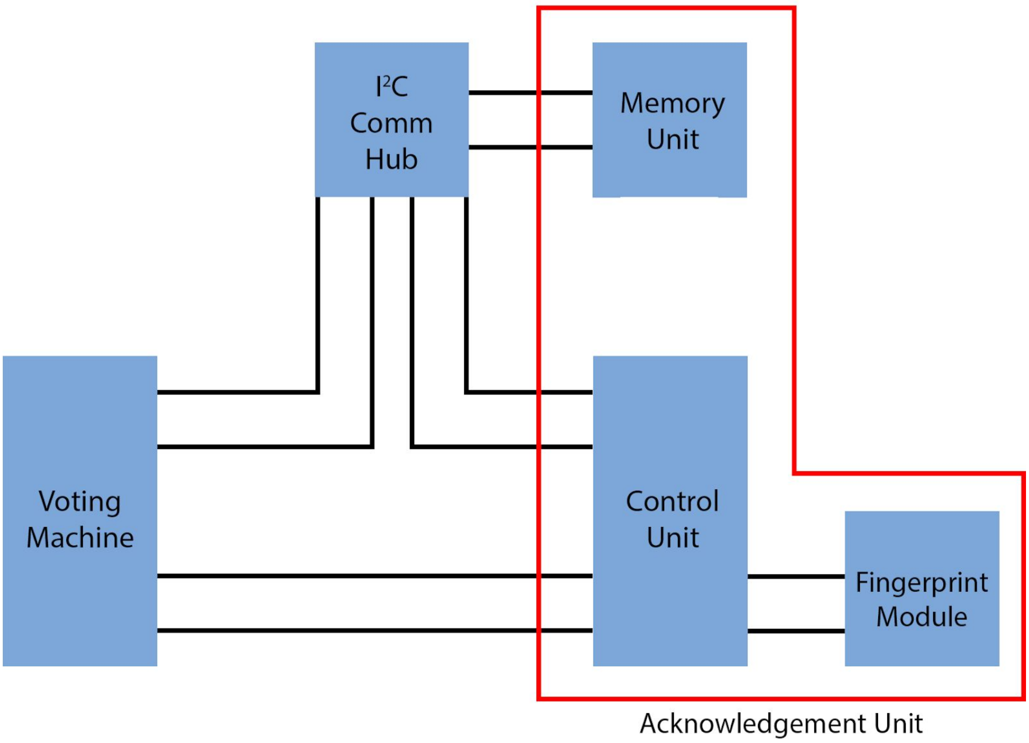
1.4 I²C Communication Hub

The communication hub comprises of the power lines: *9V*, *Gnd*, at one end connected via a switch. In addition there are *2* common *SDA* (Serial Data), *SCL* (Serial Clock) lines. All devices connect their *SDA* and *SCL* to these lines. The power line is connected to the *Vin* pin of both Arduinos to power them. All devices share a common ground. Other than the Arduinos themselves, other components draw power from the 5V power lines of the Arduinos, as it is regulated and steady.

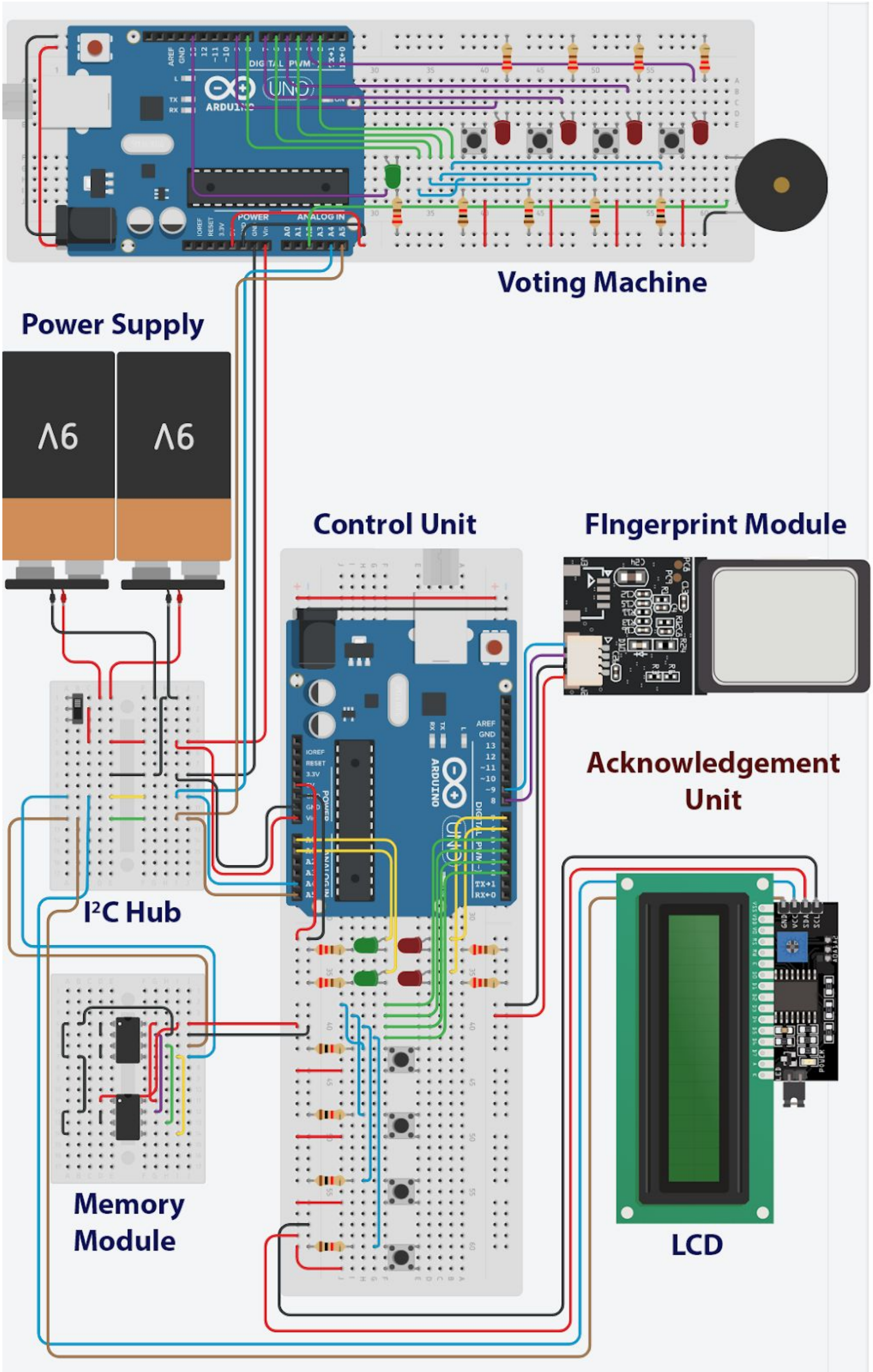
2. Schematic

The Acknowledgement Unit is composed of the Control Unit which comprises of the Arduino Nano and other circuitry. The Memory Module is included too, however, there is no direct communication between the two. The Fingerprint Module included, communicates with the Acknowledgement Unit over a software serial.

All communication takes place through the I2C Communication Hub. The Acknowledgement Unit communicates with the Voting Machine for the votes. The Acknowledgement also communicates with the Memory Module.



3. Circuit Diagram

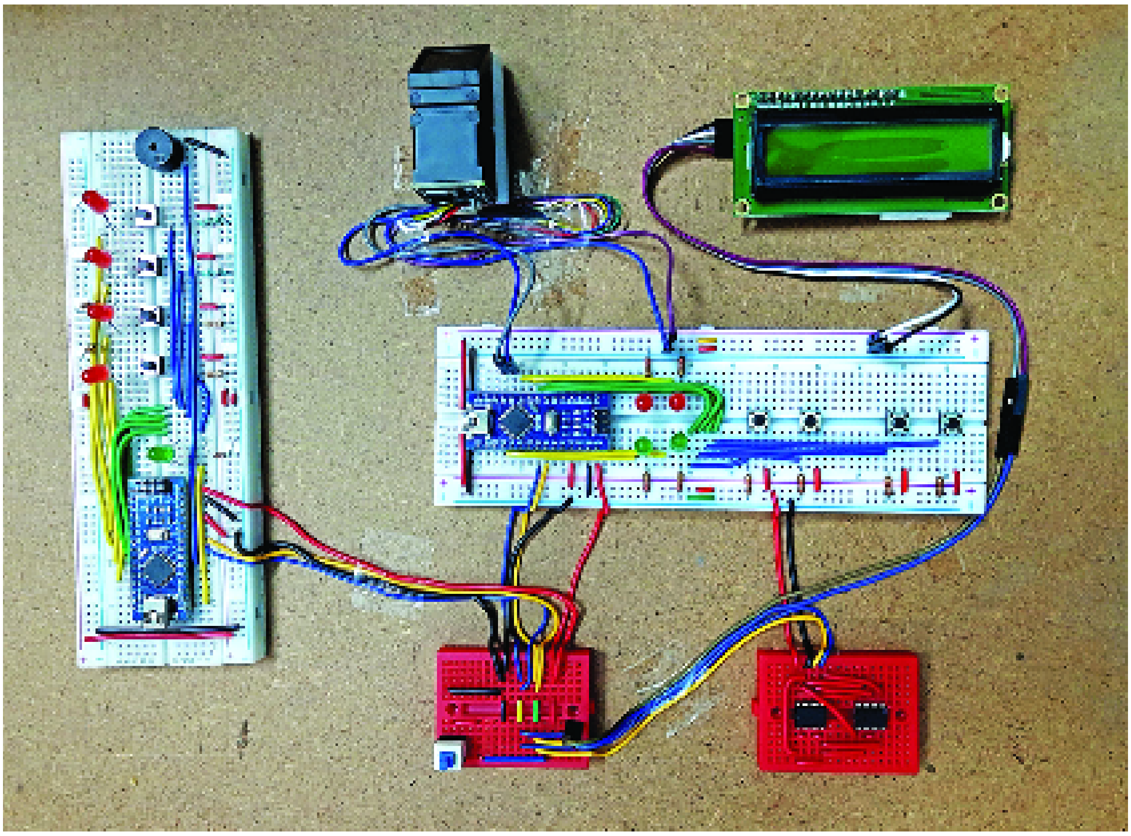


IV. Results and Discussion

1. Modus Operandi

After the device is powered on and enters ready state, the polling staff starts the verification process by pressing the verify button. The fingerprint module is activated and the voter scans his finger. The process of verification follows. On invalidity, appropriate message is notified. In the case of validity, the polling staff then activates the Voting Machine by pressing the activate button on the Control Unit. The voter then casts the vote by pressing the appropriate button on the Voting Machine. After the vote is cast, the writing operations, as mentioned earlier, are concluded. To abort a voting transaction halfway in, the polling staff can press the reset button. This particular setup also has a display button which shows the vote statistics per candidate. However, such a feature shouldn't be available on a finished device.

2. Prototype Design



3. Discussions

The aim of this project was the development of a proof of concept for a prototype of a fingerprint based voting system. Hence, it is different from a practical model. Firstly, the microcontroller board will be stripped down to meet only the requirements needed for the model, thus cutting power usage and space. The efficiency is thus increased.

Also, the circuitry will be embedded on a Printed Circuit Board (PCB), and connecting wires will be shrunk down to traces. Other miniaturization and stripping would be performed on the different modules and circuitry elements to make it more robust, reliable, and efficient.

This project has a major shortcoming. Since, the project is based on a microcontroller, the hardware interaction is pretty reliable, however, the device is not intelligent enough. There can occur issues with security, and faulty communication. Thus, a possible upgrade to a microprocessor setup, with microcontrollers to support the hardware interactions, is viable; such a setup could better handle these issues. Also, the use of a microprocessor allows more possibilities, e.g., monitoring voting behaviour to figure anomalies, and notify them.

REFERENCES

1. **Adafruit Fingerprint Sensor Library**

   https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library

2. **Arduino Wire Library**

   https://www.arduino.cc/en/Reference/Wire

3. **Liquid Crystal I2C Library**

   https://github.com/johnrickman/LiquidCrystal_I2C

4. **Forked library for Fingerprint Template Upload**

   https://github.com/hanifizzudinrahman/Module-Fingerprint-DY50-FPM10A