DESIGN AND IMPLEMENTATION OF SECURED E-VOTING MACHINE USING OPENCV WITH DEEP LEARNING

A PROJECT REPORT

Submitted by

DHYALAN.C **(611819106009)**

KARTHICK.S **(611819106016)**

SATHISH.G **(611819106046)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

P.S.V. COLLEGE OF ENGINEERING AND TECHNOLOGY

(Accredited by the NAAC with 'A' Grade)
(An ISO 9001:2015 Certified Institution)
KRISHNAGIRI

ANNA UNIVERSITY:CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"DESIGN AND IMPLEMENTATION OF SECURED E-VOTING MACHINE USING OPENCV WITH DEEP LEARNING"**is the bonafide work of "**DHYALAN.C (611819106009), KARTHICK.S (611819106016), SATHISH.G (611819106046)"** who carried out the project work under my supervision.

SIGNATURE
Prof. C. THAMILARASI, M.E.,
HOD/ Associate Professor,
Department of Electronics and Communication Engineering,
P.S.V College of Engineering and Technology,
Krishnagiri - 635 108.

SIGNATURE
Prof. L. PRAKASAM, M.E.,
Supervisor/ Assistant Professor,
Department of Electronics and Communication Engineering,
P.S.V College of Engineering and Technology,
Krishnagiri - 635 108.

Submitted for the project Viva-Voce held on …….. at P.S.V College of Engineering and Technology,Krishnagiri.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We hereby declare that the project entitled **"DESIGN AND IMPLEMENTATION OF SECURED E-VOTING MACHINE USING OPENCV WITH DEEP LEARNING"** is an authentic record of **DHYALAN.C (611819106009), KARTHICK.S (611819106016), SATHISH.G (611819106046)** who carried out work as requirements for the award of degree of B.E (Electronics and Communications) at P.S.V college of Engineering and Technology, Krishnagiri (Anna University, Chennai).

C. DHYALAN (611819106009)

S. KARTHICK (611819106016)

G. SATHISH (611819106046)

We certify that the declaration made above by candidates is true.

SIGNATURE
Prof. C. THAMILARASI, M.E.,
Head of the department,
Department of ECE,
P.S.V. College of
Engineering & Technology
Krishnagiri – 635 108.

SIGNATURE
Prof. L. PRAKASAM, M.E.,
Supervisor/Assistant Professor,
Department of ECE,
P.S.V. College of
Engineering & Technology
Krishnagiri – 635 108.

ACKNOWLEDGEMENT

At the very outset, We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our most sincere salutations go to Anna University, Chennai that gave us an opportunity to have sound base of Electronics and Communication Engineering.

We acknowledge our profound intellect of gratitude to our beloved chairman **Dr.P.SELVAM, M.A., B.Ed., M.Phil., Ph.D.,** for his support and coordination.

We also express our gratitude to our honourable principal **Dr.P.LAWRENCE, M.E., Ph.D.,** for his timely advice and his encouragement.

We offer our sincere thanks to Prof. C. THAMILARASI, M.E., Head of the Department of Electronics and Communication Engineering for giving this prospect and her full support.

We consider it as a great privilege to place a record of our deep sense of gratitude to our Internal Guide Prof. L. PRAKASAM, M.E., Assistant Professor, Electronics and Communication Engineering for being the great inspiration to us.

We also express our thanks to our Department Faculty members, our parents and our beloved friends.

DHYALAN.C

KARTHICK.S

SATHISH.G

ABSTRACT

E-voting systems are increasingly becoming popular due to their ability to provide a fast, efficient, and convenient voting process. However, security concerns such as tampering, manipulation, and fraud have been a significant challenge in the adoption of electronic voting systems.

In this project, we propose the design and implementation of a Secured E-Voting Machine that uses RFID, fingerprint sensing, face recognition, and GSM technologies to provide a secure and reliable voting system.

The proposed system is designed to be user-friendly, with a straightforward and intuitive interface that allows voters to cast their votes easily and securely. The RFID technology is used to authenticate the voter's identity, while fingerprint sensing and face recognition are used to verify the authenticity of the voter. The system also uses GSM technology to transmit voting data securely to the central server, providing real-time results while ensuring the confidentiality of the voting process.

To ensure the privacy, accuracy, and integrity of the voting process, the proposed system incorporates several security features, including encryption, authentication, and audit trails.

The system's architecture provides redundancy and fault tolerance, ensuring that the voting process continues even in the event of a technical failure or security breach.

Overall, the Secured E-Voting Machine using RFID, fingerprint sensing, face recognition, and GSM technologies provides a reliable, efficient, and secure solution for the democratic process.

The system enhances transparency, trustworthiness, and public confidence in the electoral system, thereby promoting democracy and good governance.

KEYWORDS:

Secured E-Voting Machine, RFID, Fingerprint Sensing, Face Recognition, GSM

ABSTRACT (TAMIL)–சுருக்கம்

வாக்குப்பதிவு முறைகளை ஏற்றுக்கொள்வதில் சேதப்படுத்துதல், கையாளுதல் மற்றும் மோசடி போன்ற பாதுகாப்பு கவலைகள் குறிப்பிடத்தக்க சவாலாக உள்ளன. இந்தத் திட்டத்தில், பாதுகாப்பான மற்றும் நம்பகமான வாக்களிக்கும் முறையை வழங்குவதற்காக ஆர்எஃப்ஐடி, கைரேகை உணர்தல், முகம் அடையாளம் காணுதல் மற்றும் ஜிஎஸ்எம் தொழில்நுட்பங்களைப் பயன்படுத்தும் பாதுகாப்பான மின்-வாக்களிப்பு இயந்திரத்தை வடிவமைத்து செயல்படுத்துவதை நாங்கள் முன்மொழிகிறோம்.முன்மொழியப்பட்ட அமைப்பு பயனர்களுக்கு ஏற்றதாக வடிவமைக்கப்பட்டுள்ளது, இது நேரடியாக வாக்காளர்களை எளிதாகவும் பாதுகாப்பாகவும் வாக்களிக்க அனுமதிக்கிறது.

வாக்காளரின் அடையாளத்தை அங்கீகரிக்க ஆர்எஃப்ஐடி தொழில்நுட்பம் பயன்படுத்தப்படுகிறது, அதே நேரத்தில் வாக்காளரின் நம்பகத்தன்மையை சரிபார்க்க கைரேகை உணர்தல் மற்றும் முகம் அங்கீகாரம் ஆகியவை பயன்படுத்தப்படுகின்றன. இந்த அமைப்பு ஜிஎஸ்எம் தொழில்நுட்பத்தைப் பயன்படுத்தி வாக்களிக்கும் தரவை மையச் சேவையகத்திற்குப் பாதுகாப்பாக அனுப்புகிறது, அதே நேரத்தில் வாக்களிக்கும் செயல்முறையின் இரகசியத்தன்மையை உறுதி செய்யும் போது நிகழ்நேர முடிவுகளை வழங்குகிறது.

வாக்களிக்கும் செயல்முறையின் தனியுரிமை, துல்லியம் மற்றும் ஒருமைப்பாடு ஆகியவற்றை உறுதிப்படுத்த, முன்மொழியப்பட்ட அமைப்பு அங்கீகாரம் மற்றும் தணிக்கை தடங்கள் உட்பட பல பாதுகாப்பு அம்சங்களை உள்ளடக்கியது. கணினியின் கட்டமைப்பு பணிநீக்கம் மற்றும் தவறு சகிப்புத்தன்மையை வழங்குகிறது, இது தொழில்நுட்பக் கோளாறு அல்லது பாதுகாப்பு மீறல் ஏற்பட்டாலும் வாக்களிக்கும் செயல்முறை தொடர்வதை உறுதி செய்கிறது.

ஒட்டுமொத்தமாக, ஆர்எஃப்ஐடி, கைரேகை உணர்தல், முகம் அடையாளம் காணுதல் மற்றும் ஜிஎஸ்எம் தொழில்நுட்பங்களைப் பயன்படுத்தி பாதுகாப்பான மின்-வாக்களிக்கும் இயந்திரம் ஜனநாயக செயல்முறைக்கு நம்பகமான, திறமையான மற்றும் பாதுகாப்பான தீர்வை வழங்குகிறது. தேர்தல் முறையில் வெளிப்படைத் தன்மை, நம்பகத்தன்மை மற்றும் பொதுமக்களின் நம்பிக்கையை இந்த அமைப்பு மேம்படுத்துகிறது, இதன் மூலம் ஜனநாயகம் மற்றும் நல்லாட்சியை மேம்படுத்துகிறது.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

LCD	LIQUID CRYSTAL DISPLAY
IOT	INTERNET OF THINGS
RFID	RADIO-FREQUENCY IDENTIFICATION
SMS	SHORT MESSAGE SERVICE
ID	IDENTIFICATION
MYSQL	MY STRUCTURED QUERY LANGUAGE
ASIC	APPLICATION SPECIFIC INTEGRATED CIRCUITS
VVPAT	VOTER-VERIFIED PAPER AUDIT TRAIL
EVM	ELECTRONIC VOTING MACHINE
OTP	ONE TIME PASSWORD
GSM	GLOBAL SYSTEM FOR MOBILE COMMUNICATION
GPRS	GENERAL PACKET RADIO SERVICES
I/O	INPUT/OUTPUT
3D	THREE DIMENSIONAL
UART	UNIVERSAL ASYNCHRONOUS RECEIVER-TRANSMITTER

ICSP	IN-CIRCUIT SERIAL PROGRAMMING
DC	DIRECT CURRENT
AC	ALTERNATE CURRENT
USB	UNIVERSAL SERIAL BUS
TTL	TRANSISTOR–TRANSISTOR LOGIC
ISP	IN-SYSTEM PROGRAMMING
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT
DFU IN ARDUINO	DEVICE FIRMWARE UPDATE
GND	GROUND
PWM	PULSE WIDTH MODULATION
SPI	SERIAL PERIPHERAL INTERFACE
LED	LIGHT EMITTING DIODE
TWI	TWO-WIRE INTERFACE
RX	RECEIVER
TX	TRANSMITTER
RAM	RANDOM ACCESS MEMORY
EPROM	ELECTRICALLY ERASABLE PROGRAMMABLE READ-ONLY MEMORY

TCP/IP	TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL
DDR	DOUBLE DATA RATE
LVDS	LOW VOLTAGE DIFFER- ENTIAL SIGNALING
PLL	PRECAUTIONARY AND LIQUIDITY LINE
PCB	PRINTED CIRCUIT BOARD
DAC	DIGITAL TO ANALOG
ADC	ANALOG TO DIGITAL
GPIO	GENERAL PURPOSE INPUT/OUTPUT
URL	UNIFORM RESOURCE LOCATOR
HTML	HYPERTEXT MARKUP LANGUAGE
DNN	DEEP NEURAL NETWORKS

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

The voting system is the cornerstone of any democratic society, and it's crucial that the process is secure, reliable, and accessible to all eligible voters. However, traditional voting systems face a range of challenges, including security breaches, vote manipulation, and difficulties in ensuring accessibility for all voters. To address these challenges, we propose a secure and user-friendly voting system that leverages advanced technologies such as IoT, RFID authentication, and face recognition technology. Our system is designed to ensure the integrity of the election process while maintaining the highest standards of security and accuracy.

Our system consists of a webpage used by the election officer to initiate the election process, an Arduino MEGA microcontroller that manages the voting process, and a range of authentication and verification mechanisms that ensure the identity of voters and polling officers. The system is also user-friendly, with a keypad for casting votes, an LCD display to provide real-time updates on the voting process, and an SMS notification system to confirm that votes have been successfully cast.

Overall, our proposed system represents a significant step forward in ensuring the security, accuracy, and accessibility of the election process. By leveraging advanced technologies and design principles, our system provides a robust and reliable solution that can help to restore public confidence in the democratic process.

1.2 SCOPE OF THE PROJECT

The main aim of the project is to eliminate the security compromises in the existing voting system by introducing various authentication steps (Biometric, IOT and RFID). And to provide secure voting process.

1.3 EXISTING SYSTEM

Votes are casted by just identification and casting process which lacks security. It does not have security features as the votes can be casted by any other and lot of excess polls are recorded. Due to these disadvantages this model is proposed.

1.3.1 EXISTING SYSTEM DISADVANTAGES

- Casting false votes are possible.
- Finding Non-eligible candidates is difficult.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE SURVEY

Title 1: Design and Implementation of Arduino Based Voting Machine.

Authors: Santosh Kumar Shaw, Sashank Poddar, Vivek Singh, Sudip Dogra.

Year: 2018

Description:

India being the largest democracy faces a lot of issues during elections. Lot of controversies are reported about voting system, voting machines, authentication of voting, corruptions during elections etc. In our paper we have described a secured system that can eliminate such controversies involving elections in our country. In our present work we have developed a prototype and tested successfully an Arduino UNO based Aadhar facilitated electronic voting machine possessing a Two-Tier fingerprint security. The main purpose of this system is to give a straight and fair elections and to curb all other factors that affect it, this goal has been achieved by providing dual verification of the voters based on their fingerprint and unique id. In this System all the relevant information are taken from the voters and are stored in the database, then they are provided with unique ID. The process of verification involves matching of this id and fingerprint from the database. This is a faster and more secured way of holding elections. Our system is secured, reliable and also cost- effective.

Title 2: e-Voting kiosk: A Network Architecture School-based Registration and Voting System

Authors: Giovanni N, Jessie N, Lorna

Year: 2020

Description:

The e-voting kiosk: A Network Architecture School-Based Registration and Voting System features an election management system that provides a secure, clean, and honest Supreme Student Government election for Eastern Visayas State University-Main Campus. The study is driven to design and develop a voting system that is equipped with a network-based software architecture that provides computing resources to multiple voting kiosks where voters can register and cast their votes securely. The study employs a quantitative descriptive research design to present the respondents' perception of the current manual election system compared to the e-voting system. A waterfall model was employed in the sequential process of developing the e-voting system. Researchers utilized specialized software applications such as C# and MySQL in creating a secured database for the robust development of the system. The findings of the study present a unique e-voting system powered by an election management system that features an intelligent system algorithm. The e- voting kiosk features interactive ballots that allow voters to easily cast their votes securely. The system provides secured access to voters' identity and process real-time generation of election results. The observed system performance resulted to a significant rating of 4.04 interpreted as “Very Efficient” and its software characteristics was rated 95% for its contribution in the election proceedings. Finally, the electoral board unanimously approved the implementation and use of e-voting system which provides fast, effective, efficient, and reliable election management system.

Title 3: Prototyping of Indian Electronic Voting Machine

Authors: Tushar Puri, Jaspreet Singh, Hemant Kaushal.

Year: 2017

Description:

The Voting System of a country consists of certain regulations which define how the preference of people is collected and how outcome of the polling process is indicating the will of people. To implement such a system in the largest democracy in the world is a cumbersome task. An indigenous Electronic Voting Machine was introduced by the Election Commission of India to overcome the issues with manual voting which was slower and inefficient. In this paper the Indian Electronic Voting Machine's Protocol for voting is implemented on a field programmable gate array. The ASIC based design is known to be faster than a microcontroller based design. Furthermore the use of an ASIC based design will make the Electronic Voting Machine a more reliable and tamper resistant machine. The new Voter-verified paper audit trail (VVPAT) system could also be interfaced with the ASIC based design. The protocol of Indian Electronic Voting Machine has been successfully implemented on a Basys 2 board using Verilog HDL. The FPGA based implementation gets half the job done for ASIC based EVM. The tool used for simulation and implementation is Xilinx ISE Design Suite with ISim as a Simulator.

Title 4: Smart Online Voting System

Authors: Ganesh Prabhu S, Nizarahammed.A, Prabu.S, Raghul.S, R.R.Thirrunavukkarasu, P. Jayarajan

Year: 2021

Description:

Our country, India is the largest democratic country in the world. So it is essential to make sure that the governing body is elected through a fair election. India has only offline voting system which is not effective and upto the mark as it requires large man force and it also requires more time to process and publish the results. Therefore, to be made effective, the system needs a change, which overcomes these disadvantages. The new method does not force the person's physical appearance to vote, which makes the things easier. This paper focusses on a system where the user can vote remotely from anywhere using his/her computer or mobile phone and doesn’t require the voter to got to the polling station through two step authentication of face recognition and OTP system. This project also allows the user to vote offline as well if he/she feels that is comfortable. The face scanning system is used to record the voters face prior to the election and is useful at the time of voting. The offline voting system is improvised with the help of RFID tags instead of voter id. This system also enables the user the citizens to see the results anytime which can avoid situations that pave way to vote tampering

Title 5: Election Voting Machine - A Review

Authors: Sanket M. Gawade, Ninad S. Mandavkar, Sanket S. Mane, Chinmayee N. Manjarekar.

Year: 2017

Description:

Electronic Voting Machine (EVM) is a simple electronic machine used to record votes in place of ballot papers and boxes which were used earlier in conventional voting system. The problem with conventional voting machine is unreliable use of ballot papers which makes process time consuming and costlier. These problems solved by using biometrics identifier which is more reliable for person identification than traditional voting technology methods. The focus of this paper is to summarize the emerging election voting machine, presents these new election voting machine and survey the recent trends for EVM for advance voting machine. This paper also discusses the issues and comparison among the different voting machines.

2.2 PROPOSED SYSTEM

In this proposed system the voting system is improved with security features to avoid various security issues like illegal voting's. The proposed system consists of various security authentications by various election officers followed by the implementation of aadhar with voters ID which is for the voter's identification and the votes are casted by the voters using their fingerprints. The voters can get a notification of their casting as an alert SMS. This intimates the voters and they can crosscheck whether they have voted or not. This can be

implemented and can provide safe and secure voting. The image processing are used to identify the face ID of that voter ID.

2.2.1 PROPOSED SYSTEM ADVANTAGES

- No chance of casting false votes.
- Only the eligible candidates can caste vote.
- Votes are cast using fingerprints.
- Biometric security feature is implemented and uploaded data to the cloud using GSM module.

CHAPTER 3

PROJECT DESCRIPTION

3.1 GENERAL

The security in the existing voting system is not to the expected level. So in the proposed model security is enhanced by adding an IOT based authentication and followed by RFID identification and the Biometric authentication of the voting system which increases security of the model. This can be implemented to avoid lags in security.

3.2 BLOCK DIAGRAM

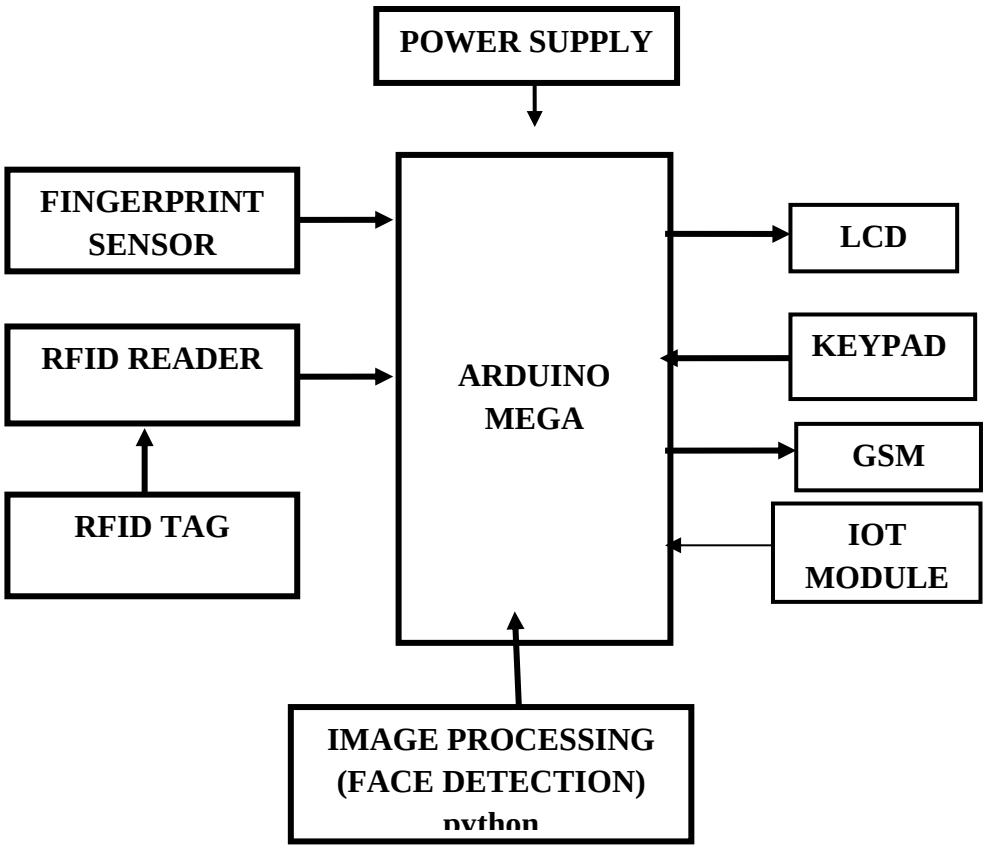


Fig.no. 3.2: BLOCK DIAGARM

3.3 MODULES NAME

- ❖ IOT Authentication
- ❖ RFID Authentication
- ❖ Biometric Authentication
- ❖ Vote casting

3.4 MODULE DESCRIPTION

3.4.1 IOT AUTHENTICATION

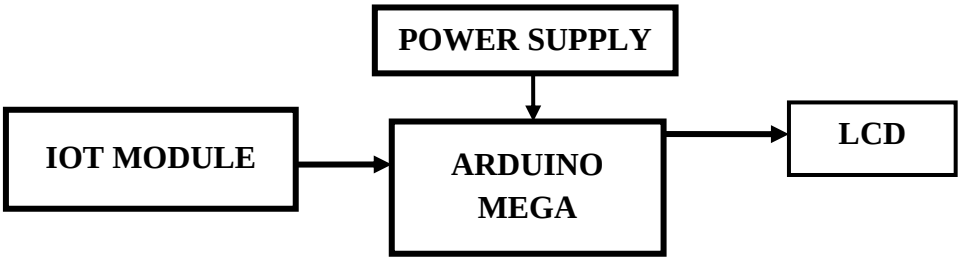


Fig 3.4.1 IOT AUTHENTICATION

In this voting system, the first module is the authentication of the election officers by an IOT webpage which increases the security of the system. Through the webpage the election officer initiates the start of the election. The prototype provided is activated by the command from the IOT webpage which is given by the officer. The command from the webpage is received by the microcontroller (Arduino MEGA) and the voting process is initiated, this process is followed by the RFID authentication.

3.4.2 RFID AUTHENTICATION

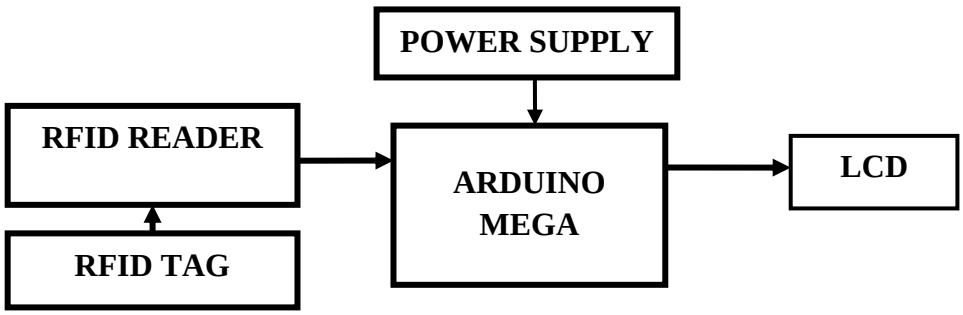


Fig.3.4.2: RFID AUTHENTICATION

The second module which is followed by the IOT authentication is the authentication by the polling officers who are incharge of the polling booth. The polling officers with their RFID tag ensures their identity and they are asked for their biometric for security reasons. The voter ID which is linked with the aadhar is a RFID tag by which the voters can ensure their identification in the polling booths. The LCD attached shows the process of the model.

3.4.3 FACE AUTHENTICATION

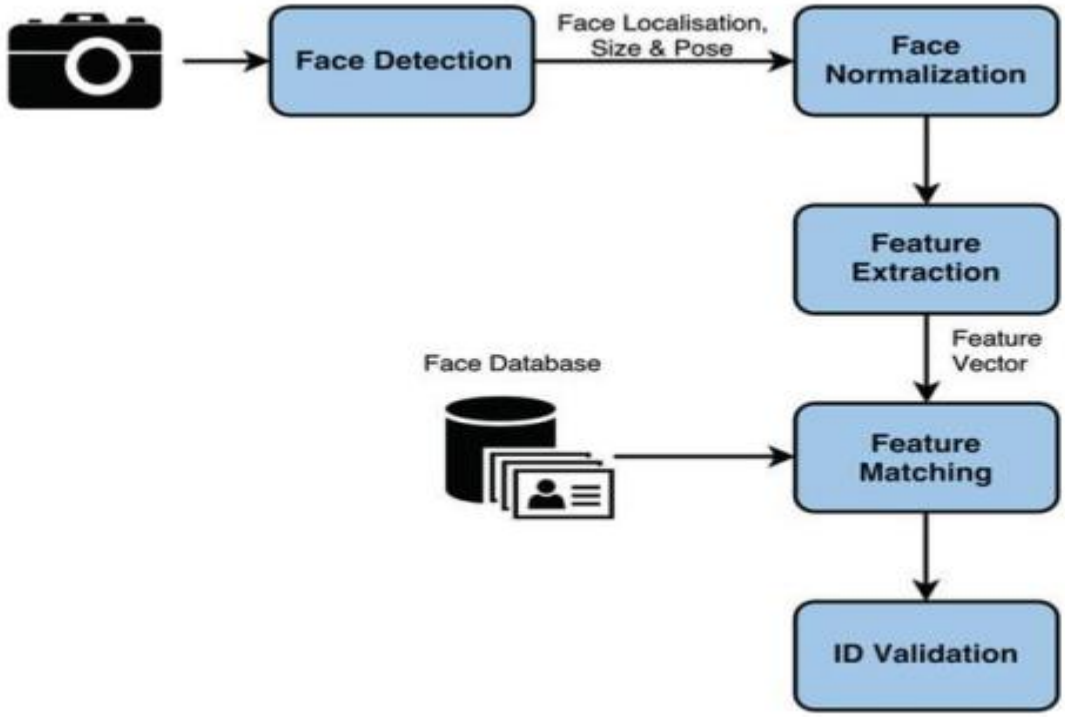


Fig. 3.4.3: FACE AUTHENTICATION

The face authentication process in the e-voting machine uses a camera placed in the machine to capture the face of the voter. The image is then processed using OpenCV and face recognition library packages to detect and identify the face.

The system compares the face data obtained from the captured image with the face data stored in the database. If the system finds a match, the voter's identity is verified, and they are allowed to proceed with the voting process. If there is no match, the voter is denied access to the e-voting machine.

The face authentication process plays a crucial role in preventing identity theft and other forms of voter fraud, ensuring that only authorized individuals are allowed to cast their vote. It provides an additional layer of security to the e-voting machine, enhancing the integrity and accuracy of the voting process.

3.4.4 BIOMETRIC AUTHENTICATION

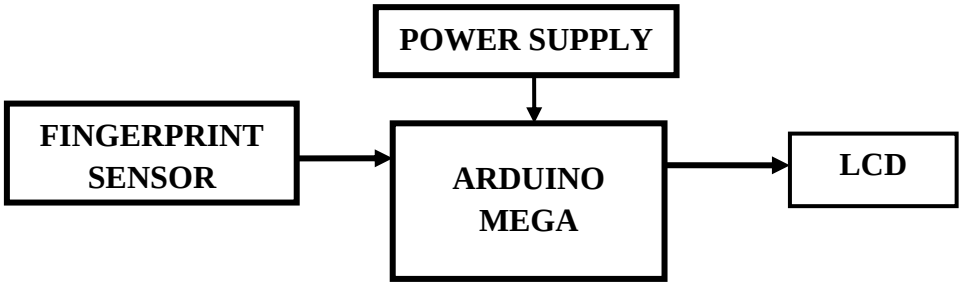


Fig. 3.4.4 : BIOMETRIC AUTHENTICATION

The biometric authentication is one of the feature where the security is enhanced. The polling officer in the booth ensures their identity by RFID followed by their biometric when again activates the prototype after the IOT authentication by the election officer. The voters also ensures their identity by the RFID and followed by their biometric to cast their votes.

3.4.5 VOTE CASTING

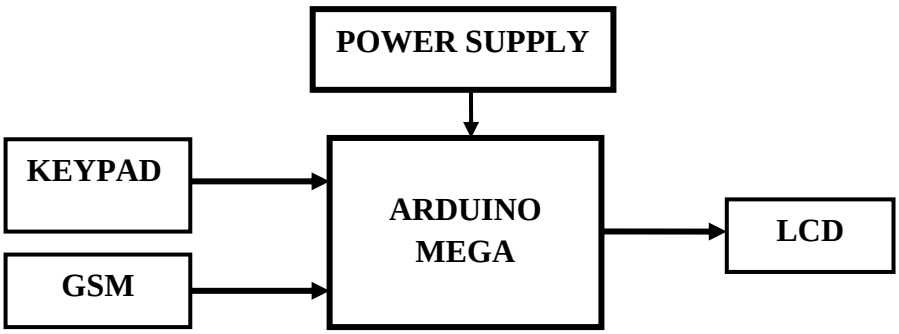


Fig .3.4.5: VOTE CASTING

After the verification of the biometric and votes are casted by the voters using the Keypad and a SMS is sent to the voters. All these processes are displayed in the LCD.

3.5 BLOCK DIAGRAM DESCRIPTION AND WORKING

In our system, we have Arduino mega microcontroller which acts as brain of our system, hence entire system program is stored in it. Initially each voter has unique ID i.e. RFID which is accessed by reader, then he/she has to place thumb on fingerprint sensor if that ID and finger matched and the image processing used to detect the voters face ID and confirmed the voter identification means the voter will receive conformation message to his/her registered mobile number via GSM, they have to type it through keypad, then they can vote for favorable candidate by keypad, switch is used to send the vote count to the controller via GSM. The IoT module is used to monitor the voting status and control the voting machine.

CHAPTER 4

HARDWARE AND SOFTWARE DESCRIPTION

4.1HARDWARE DESCRIPTION

4.1.1 ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

4.1.1.1 ARDUINO MEGA

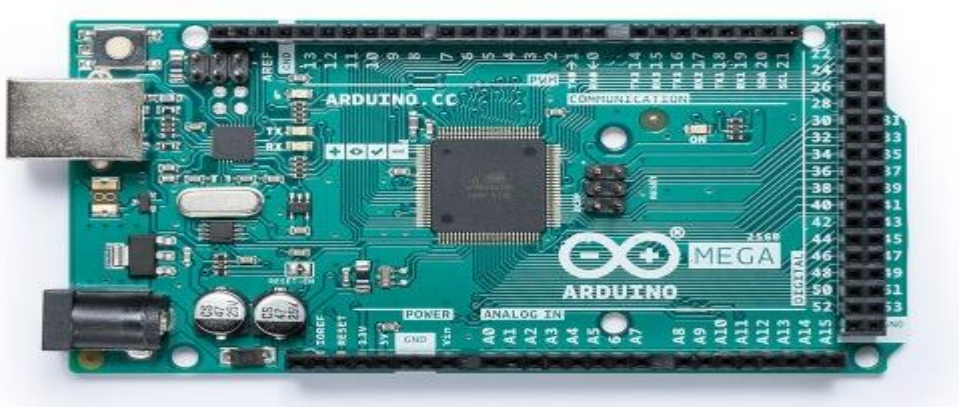


Fig .4.1.1.1 : ARDUINO MEGA

The MEGA 2560 is designed for more complex projects. With 54 digital I/O pins, 16 analog inputs and a larger space for your sketch it is the

recommended board for 3D printers and robotics projects. This gives your projects plenty of room and opportunities.

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

TECHNICAL SPECIFICATIONS

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by boot loader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Table .4.1.1: TECHNICAL SPECIFICATION

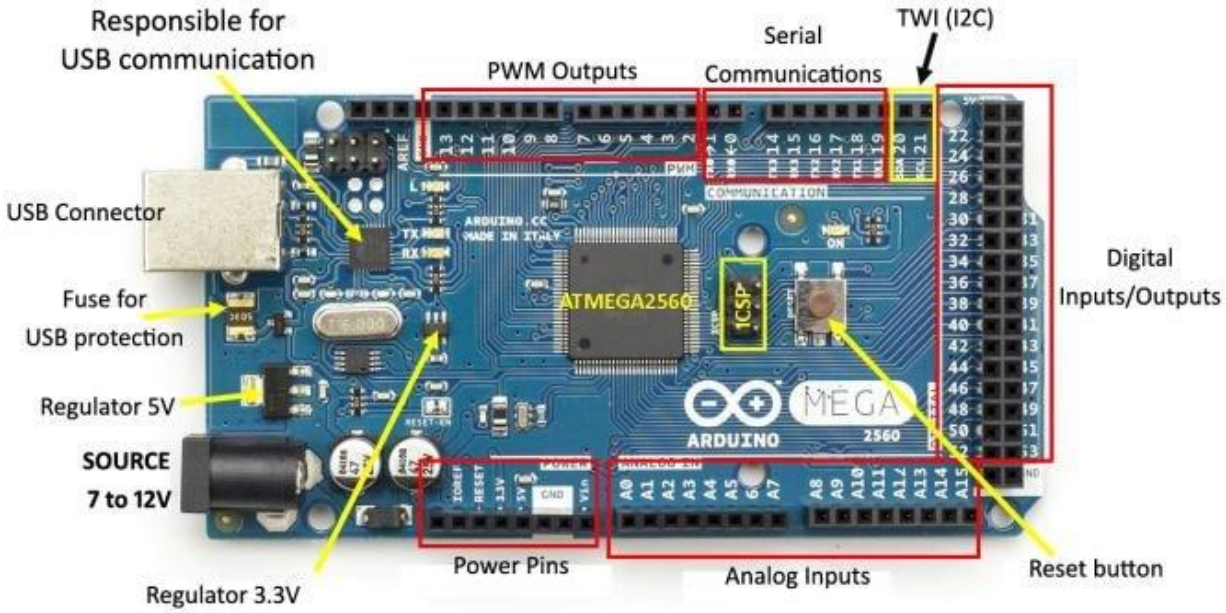


Fig 4.1.1.2: SPECIFICATIONS OF ARDUINO MEGA

HARDWARE

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name Arduino to

be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in -Arduino. An early Arduino board with an RS-232 serial interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are at the top, the 6 analog input pins at the lower right, and the power connector at the lower left.

Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the LilyPad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino UNO is the optiboot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model

Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Arduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) is used. An official Arduino Uno R2 with descriptions of the I/O locations

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Bo Arduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

PROGRAMMING

The Mega 2560 board can be programmed with the Arduino Software (IDE). For details, see the reference and tutorials.

The ATmega2560 on the Mega 2560 comes preprogrammed with a boot loader 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 boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

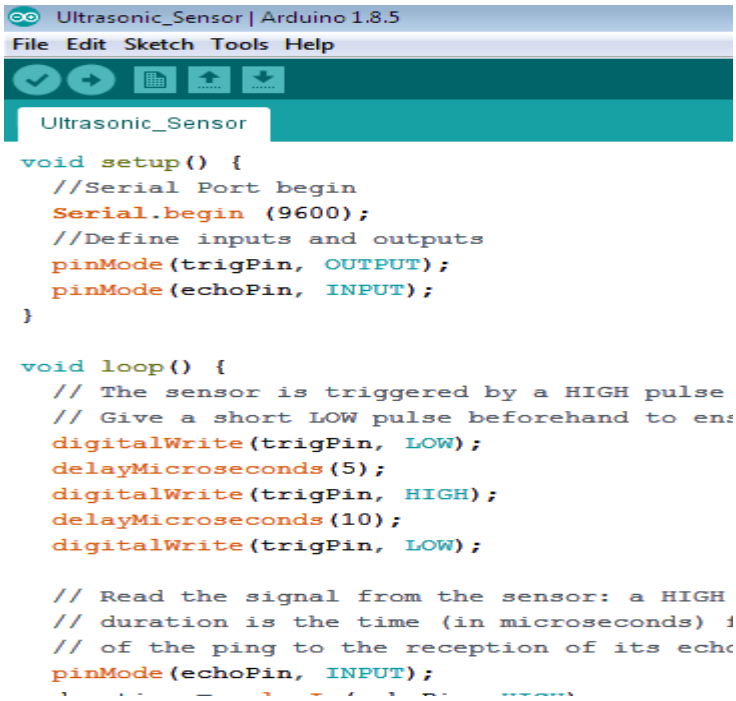


Fig 4.1.1.3: ARDUINO IDE

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

- **On Rev1 boards:** connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- **On Rev2 or later boards:** there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into

DFU mode. 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 boot loader). See this user-contributed tutorial for more information.

POWER

The Mega 2560 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 become 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 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**. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the

DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

- **3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**. Ground pins.
- **IOREF**. This pin on the board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

MEMORY

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

INPUT AND OUTPUT

ATmega2560-Arduino Pin Mapping

Below is the pin mapping for the Atmega2560. The chip used in Arduino 2560. There are pin mappings to Atmega8 and Atmega 168/328 as well. Each of the 54 digital pins on the Mega 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 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by

default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller.

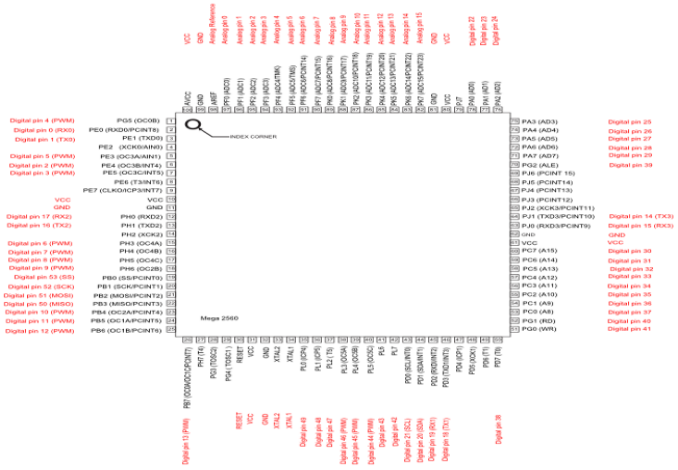


Fig 4.1.1.4: ARDUINO MEGA 2560 PIN DIAGRAM

In addition, some pins have specialized functions:

- **Serial**: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.
- **External Interrupts**: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the attachInterrupt() function for details.
- **PWM**: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analogWrite() function.
- **SPI**: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using theSPI library. The SPI pins are also

broken out on the ICSP header, which is physically compatible with the Arduino /Genuino Uno and the old Duemilanove and Diecimila Arduino boards.

- **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.
- **TWI**: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the old Duemilanove or Diecimila Arduino boards.

See also the mapping Arduino Mega 2560 PIN diagram.

The Mega 2560 has 16 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 analogReference() function. 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.

4.1.1 POWER SUPPLY

This section describes how to generate +5V DC power supply

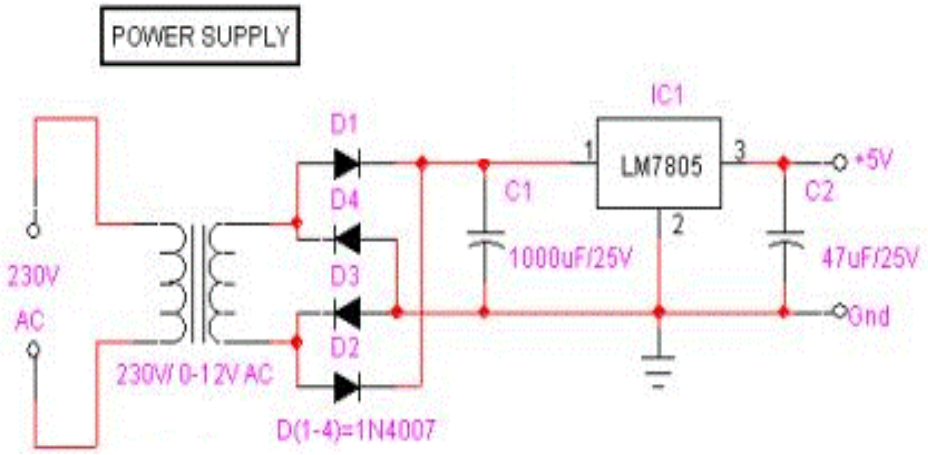


Fig 4.1.2: CIRCUIT DIAGRAM OF DC POWER SUPPLY +5V

The power supply section is the important one. It should deliver constant output regulated power supply for successful working of the project. A 0-12V/1 mA transformer is used for this purpose. The primary of this transformer is connected in to main supply through on/off switch& fuse for protecting from overload and short circuit protection. The secondary is connected to the diodes to convert 12V AC to 12V DC voltage. And filtered by the capacitors, which is further regulated to +5v, by using IC 7805.

4.1.2 GSM

Fig 4.1.3 : LINKSPRIT GSM/GPRS MODULE

A GSM modem is a wireless modem that works with a GSM wireless network. A

wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves. The working of GSM modem is based on commands, the commands always start with AT (which means ATtention) and finish with a <CR> character. For example, the dialing command is ATD<number>; ATD3314629080; here the dialing command ends with semicolon.

The AT commands are given to the GSM modem with the help of PC or controller. The GSM modem is serially interfaced with the controller with the help of MAX 232. Here max 232 acts as driver which converts TTL levels to the RS 232 levels. For serial interface GSM modem requires the signal based on RS 232 levels. The T1_OUT and R1_IN pin of MAX 232 is connected to the TX and RX pin of GSM modem

OBJECT OF THE DOCUMENT

This document gives an overview of the LinkSprite GSM/GPRS module: a miniature, single-side board, quad-band GSM 850/EGSM 900/DCS 1800/PCS 1900 module, ready for integration in various kinds of Fix wireless phones and other wireless devices.

SYSTEM BLOCK DIAGRAM

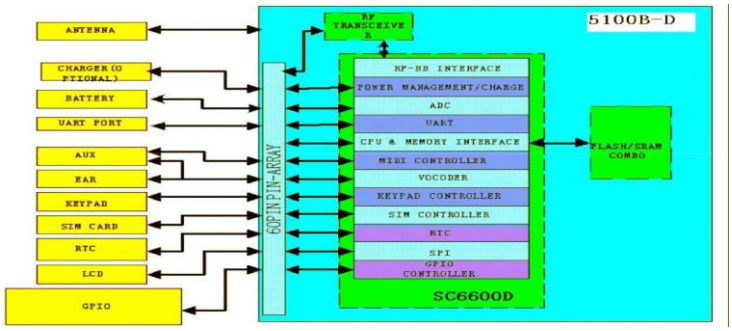


Fig 4.1.3.2: SYSTEM BLOCK DIAGRAM

4.1.3 RFID READER AND TAG

An RFID reader is a device that is used to interrogate an RFID tag. The reader has an antenna that emits radio waves; the tag responds by sending back its data.

An RFID tag is a microchip combined with an antenna in a compact package; the packaging is structured to allow the RFID tag to be attached to an object to be tracked. "RFID" stands for Radio Frequency Identification. The tag's antenna picks up signals from an RFID reader or scanner and then returns the signal, usually with some additional data (like a unique serial number or other customized information).

A passive tag is an RFID tag that does not contain a battery; the power is supplied by the reader. When radio waves from the reader are encountered by a passive rfid tag, the coiled antenna within the tag forms a magnetic field. The tag draws power from it, energizing the circuits in the tag. The tag then sends the information encoded in the tag's memory.

The RX and TX pins of RFID reader connected to Tx and Rx pins of 8051 Microcontroller respectively. Then the reader senses the data from the Tag and transmits the sensed data to microcontroller via serial port.

Fig 4.1.4.1 : RFID Reader

Fig 4.1.4.2 : RFID Card

The EM-18 RFID Reader module operating at 125kHz is an inexpensive solution for your RFID based application. The Reader module comes with an on-chip antenna and can be powered up with a 5V power supply. Power-up the module and connect the transmit pin of the module to receive pin of your microcontroller. Show your card within the reading distance and the card number is thrown at the output. Optionally the module can be configured for also a and output.

Typical Applications

- E-Payment
- E-Toll Road Pricing
- E-Ticketing for Events
- E-Ticketing for Public Transport
- Access Control
- PC Access
- Authentication
- Printer / Production Equipment

Features	
RF Transmit Frequency	125kHz
Supported Standards	EM4001 64-bit RFID tag compatible
Communications Interface	TTL Serial Interface, Wiegand output
Communications Protocol	Specific ASCII
Communications Parameter	9600 bps, 8, N, 1
Power Supply	4.6V - 5.5VDC ± 10% regulated
Current Consumption	50 mA < 10mA at power down mode.
Reading distance	Up to 100mm, depending on tag
Antenna	Integrated
Size (LxWxH)	32 x 32 x 8mm

Table 4.1.4 : RFID Features

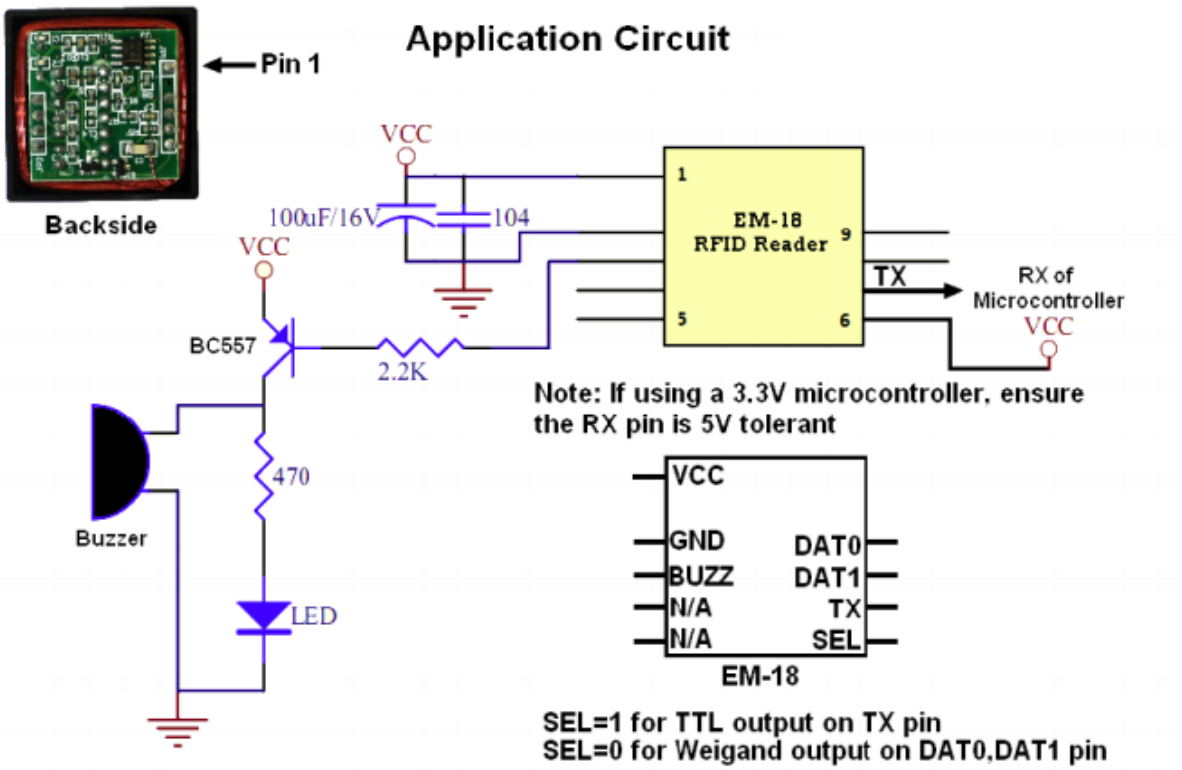


Fig 4.1.4.3: CIRCUIT DIAGRAM OF RFID READER

4.1.4 KEYPAD

Fig 4.1.5.1 : 4*4 matrix KEYPAD

At the lowest level, keyboards are organized in a matrix of rows and columns. The CPU accesses both rows and column through ports; therefore, with two 8-bit ports, an 8*8 matrix of keys can be connected to a microprocessor. When a key pressed, a row and column make a connect otherwise, there is no connection between row and column. In IBM PC keyboards, a single microcontroller (consisting of microprocessor, RAM and EPROM, and several ports all on a single chip) takes care of software and hardware interfacing of keyboard. In such systems it is the function of programs stored in the EPROM of microcontroller to scan the keys continuously, identify which one has been activated, and present it to the motherboard.

The rows are connected to an output port and the columns are connected to an input port. If no key has been pressed, reading the input port will yield 1s for all columns since they are all connected to high (Vcc) If all the rows are grounded and a key is pressed, one of the columns will have 0 since the key pressed provides the path to ground. It is the function of the microcontroller to scan the keyboard continuously to detect and identify the key pressed.

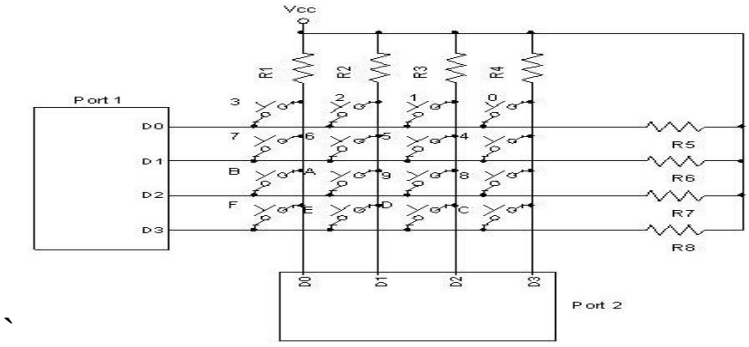


Fig 4.1.5.2: CIRCUIT DIAGRAM OF KEYPAD

4.1.5 LCD

LCD screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

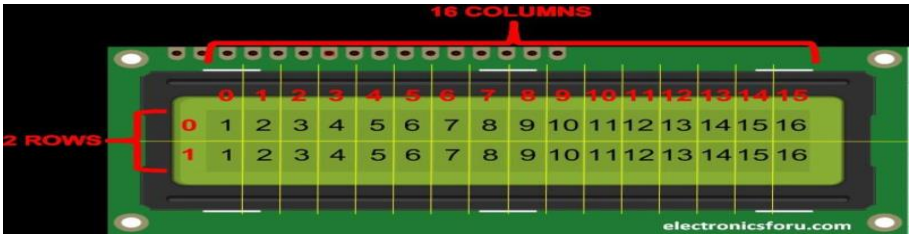


Fig 4.1.6.1: 16x2 LCD display

We come across LCD displays everywhere around us. Computers, calculators, television sets, mobile phones, digital watches use some kind of display to display the time. An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in projects. The 16×2 translates to a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix.

16X2 LCD PINOUT DIAGRAM

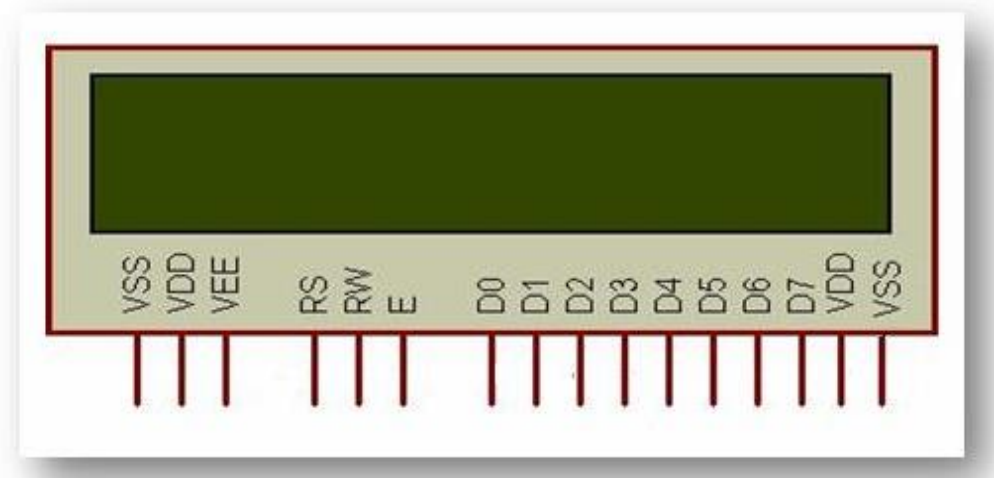


Fig 4.1.6.2 **:** 16x2 LCD display ports

PIN NO.	FUNCTION	NAME
1	Ground (0V)	Ground

2	Supply voltage; 5V (4.7V – 5.3V)	VCC
3	Contrast adjustment; the best way is to use a variable resistor such as a potentiometer. The output of the potentiometer is connected to this pin. Rotate the potentiometer knob forward and backwards to adjust the LCD contrast.	Vo / VEE
4	Selects command register when low, and data register when high	RS (Register Select)
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given; Extra voltage push is required to execute the instruction and EN(enable) signal is used for this purpose. Usually, we make it en=0 and when we want to execute the instruction we make it high en=1 for some milliseconds. After this we again make it ground that is, en=0.	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight VCC (5V)	Led+
16	Backlight Ground (0V)	Led-

Table 4.1.6 : FUNCTIONS AND NAMES OF PINS IN LCD DISPLAY

4.1.6 FINGERPRINT SENSOR

R305 Fingerprint Module consists of optical fingerprint sensor, high-speed DSP processor, high-performance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition, stable performance, simple structure, with fingerprint entry, image processing, fingerprint matching, search and template storage and other functions.

Fig 4.1.7 : FINGERPRINT MODULE R305

Fingerprint Module R305

R305 is a finger print sensor module with TTL UART interface. The user can store the finger print data in the module and can configure it in 1:1 or 1: N mode for identifying the person. The FP module can directly interface with 3v Microcontroller. A level converter (like MAX232) is required for interfacing with PC.

The R305 fingerprint module has two interface TTL UART and USB2.0, USB2.0 interface can be connected to the computer; RS232 interface is a TTL level, the default baud rate is 57600 , can be changed, refer to a communication protocol ; can And microcontroller, such as ARM, DSP and other serial devices with a connection, 3.3V 5V microcontroller can be connected directly. Needs to connect the computer level conversion, level conversion note, embodiments such as a MAX232 circuit.

4.1.7 INTERNET OF THINGS (IOT)

The **Internet Of Things** (**IoT**) is the network of physical devices, vehicles, buildings and other items embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data. In 2013 the Global Standards Initiative on Internet of Things (IoT-GSI) defined the IoT as "the infrastructure of the information society. The IoT

allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit.

ESP-12E BASED NODEMCU

The ESP8266 is the name of a micro controller designed by Espressif Systems. The ESP8266 itself is a self-contained Wi-Fi networking solution offering as a bridge from existing micro controller to Wi-Fi and is also capable of running self-contained applications. This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.

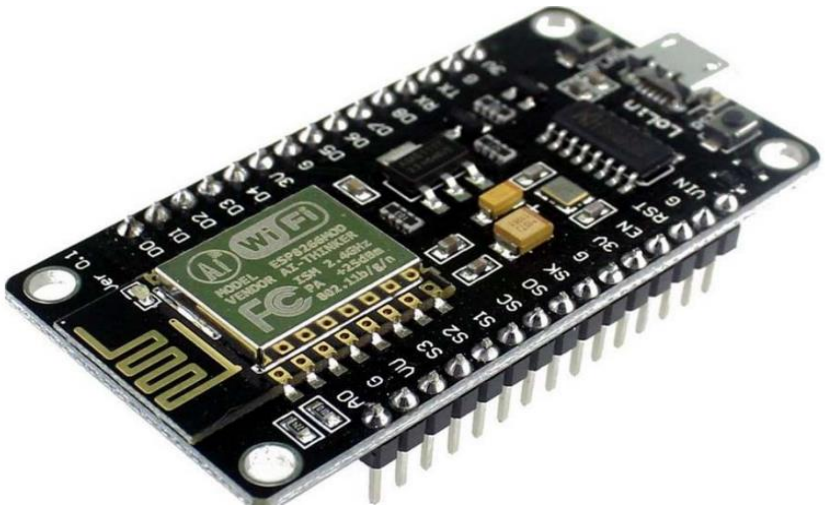


Fig 4.1.8.1 : ESP-12E BASED NODEMCU

ESP-12E Wi-Fi module is developed by Ai-thinker Team. core processor ESP8266 in smaller sizes of the module encapsulates Tensilica L106 integrates industry-leading ultra-low power 32-bit MCU micro, with the 16-bit short mode, Clock speed support 80 MHz, 160 MHz, supports the RTOS, integrated Wi-Fi MAC/BB/RF/PA/LNA, on-board antenna. The module supports standard IEEE802.11 b/g/n agreement, complete TCP/IP protocol stack. Users can use the add modules to an existing device networking, or building a separate network controller. ESP8266 is high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

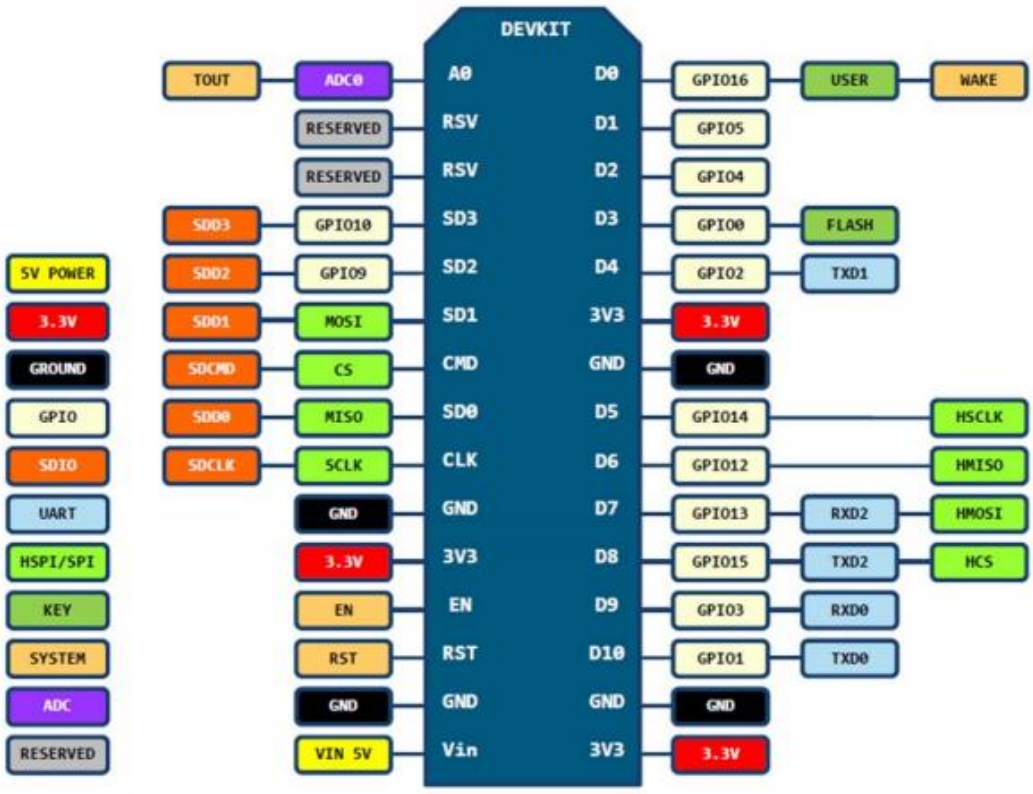


Fig 4.1.8.2 : NODEMCU PIN CONFIGURATION

ESP8266EX offers a complete and self-contained Wi-Fi networking solution; it can be used to host the application or to offload Wi-Fi networking functions from another application processor. When ESP8266EX hosts the

application, it boots up directly from an external flash. In has integrated cache to improve the performance of the system in such applications. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any micro controller based design with simple connectivity (SPI/SDIO or I2C/UART interface). ESP8266EX is among the most integrated Wi-Fi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

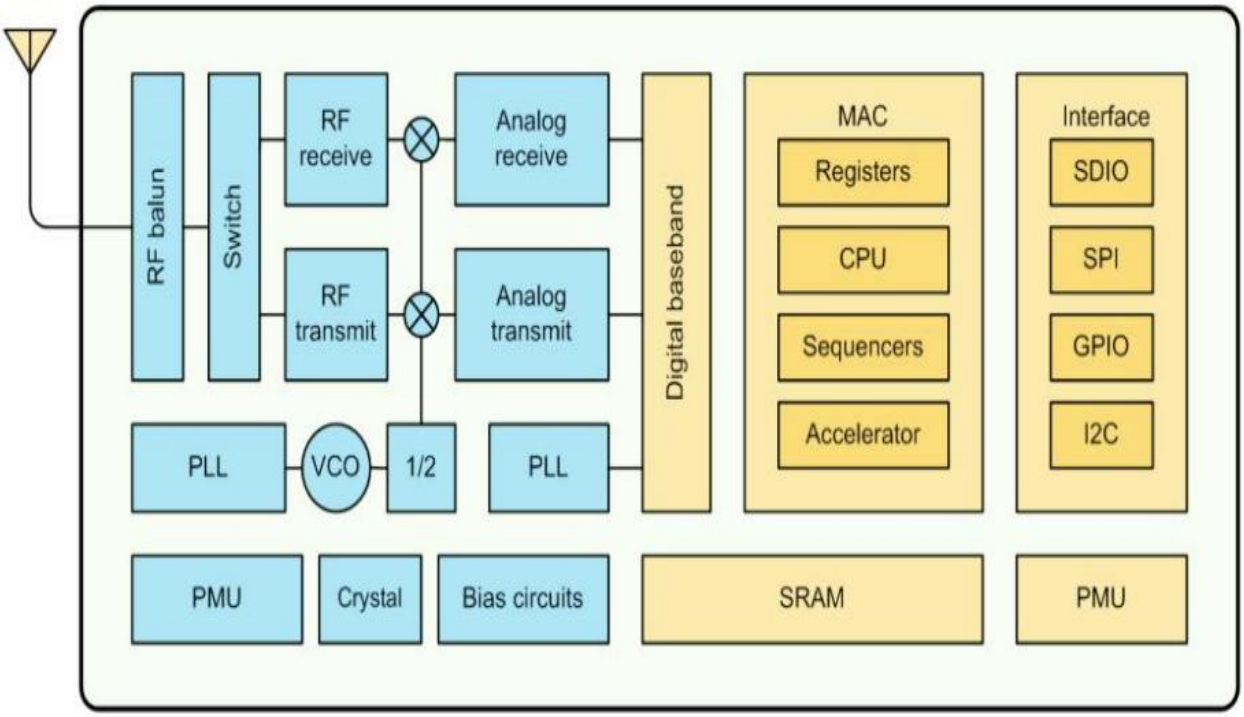


Fig 4.1.8.3 : BLOCK DIAGRAM OF IOT MODULE

ESP-12E ARCHITECTURE

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the Wi-Fi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; codes for such applications are

provided in examples in the SDK. Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing. For low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- Wi-Fi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IRDA, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation and 0.4s guard interval
- Deep sleep power < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C

4.2SOFTWARE REQUIREMENTS

4.2.1 EMBEDDED C

Embedded C is most popular programming language in software field for developing electronic gadgets. Each processor used in electronic system is associated with embedded software.

Embedded C programming plays a key role in performing specific function by the processor. In day-to-day life we used many electronic devices such as mobile phone, washing machine, digital camera, etc. These all device working is based on microcontroller that are programmed by embedded C.

Let's see the block diagram representation of embedded system programming:

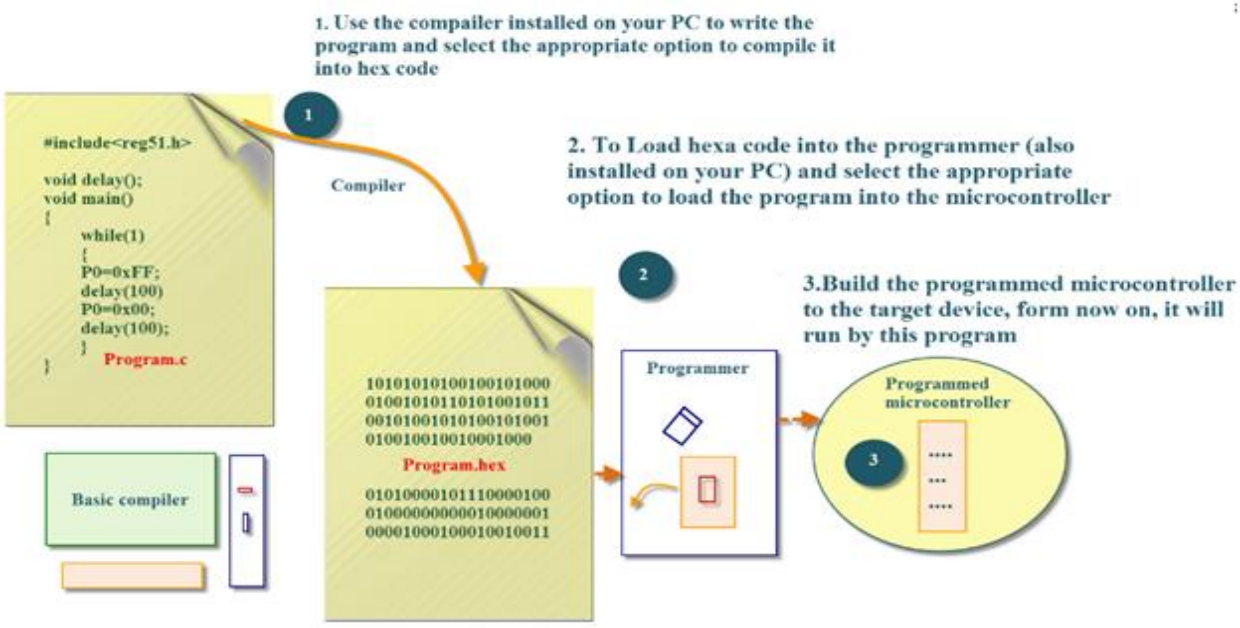


Fig 4.2.1.1 : BLOCK DIAGRAM REPRESENTATION OF EMBEDDED C

The Embedded C code written in above block diagram is used for blinking the LED connected with Port0 of microcontroller.

In embedded system programming C code is preferred over other language. Due to the following reasons:

- Easy to understand
- High Reliability
- Portability
- Scalability

EMBEDDED SYSTEM PROGRAMMING

Basic Declaration

Let's see the block diagram of Embedded C Programming development:

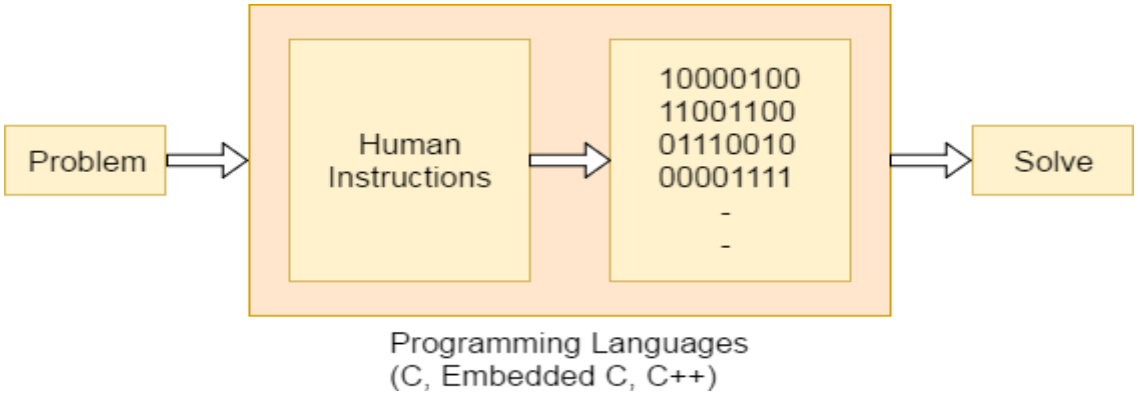


Fig 4.2.1.2 : BLOCK DIAGRAM OF EMBEDDED C PROGRAMMING DEVELOPMENT

Function is a collection of statements that is used for performing a specific task and a collection of one or more functions is called a programming language. Every language is consisting of basic elements and grammatical rules. The C language programming is designed for function with variables, character set, data types, keywords, expression and so on are used for writing a C program.

The extension in C language is known as embedded C programming language. As compared to above the embedded programming in C is also have some additional features like data types, keywords and header file etc is represented by #include<**microcontroller** name.h>.

BASIC EMBEDDED C PROGRAMMING STEPS

Let's see the block diagram representation of Embedded C Programming Steps:

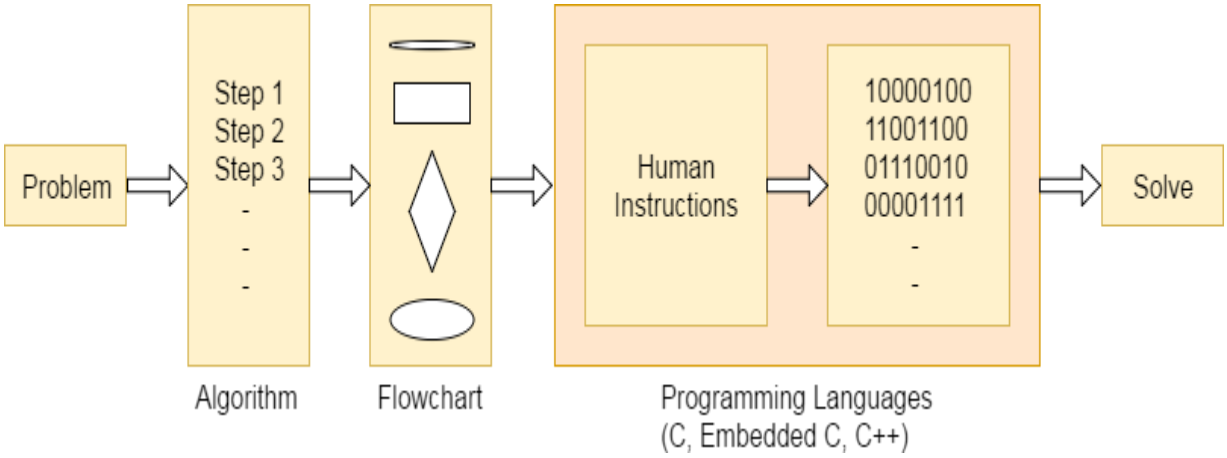


Fig 4.2.1.3 : BLOCK DIAGRAM REPRESENTATION OF EMBEDDED C PROGRAMMING STEPS

The microcontroller programming is different for each type of operating system. Even though there are many operating system are exist such as Windows, Linux, RTOS, etc but RTOS has several advantage for embedded system development.

EMBEDDED SYSTEMS

Embedded System is a system composed of hardware, application software and real time operating system. It can be small independent system or large combinational system.

Our Embedded System tutorial includes all topics of Embedded System such as characteristics, designing, processors, microcontrollers, tools, addressing modes, assembly language, interrupts, embedded c programming, led blinking, serial communication, lcd programming, keyboard programming, project implementation etc.

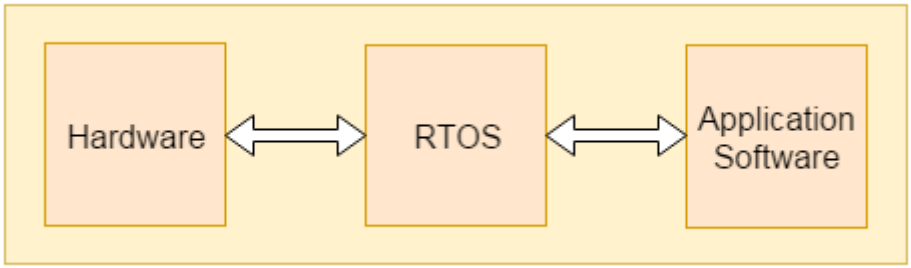


Fig 4.2.1.4 : EMBEDDED SYSTEM

CHARACTERISTICS OF EMBEDDED SYSTEM

- An embedded system is software embedded into computer hardware that makes a system dedicated to be used for variety of application.
- Embedded system generally used for do specific task that provide real-time output on the basis of various characteristics of an embedded system.
- Embedded system may contain a smaller part within a larger device that used for serving the more specific application to perform variety of task using hardware-software intermixing configuration.
- It provides high reliability and real-time computation ability.

Advantages

- Same hardware can be used in variety of application.
- Lesser power requirement
- Lower operational cost of system
- Provide high performance and efficiency

Disadvantages

- Developing a system required more time. Due to functional complexity.

- Skilled engineers required because one mistake may result in destroying of complete project.

Designing of an embedded system

BASIC STRUCTURE OF AN EMBEDDED SYSTEM

Let's see the block diagram shows the basic structure of an embedded system.

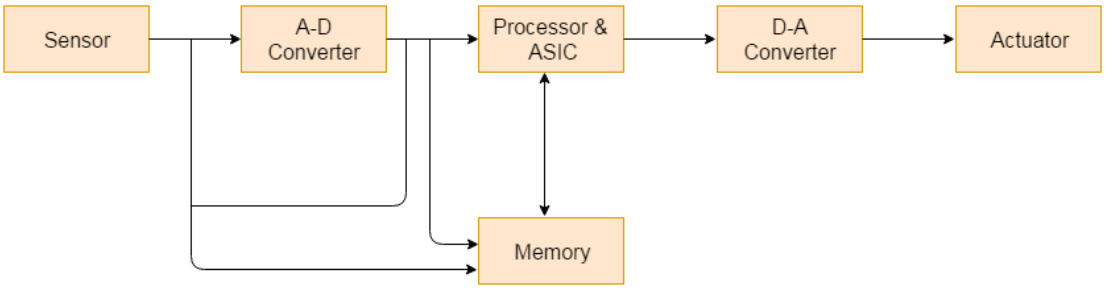


Fig 4.2.1.5 **:** BASIC STRUCTURE OF AN EMBEDDED SYSTEM

- **Sensor**: Sensor used for sensing the change in environment condition and it generate the electric signal on the basis of change in environment condition. Therefore it is also called as transducers for providing electric input signal on the basis of change in environment condition.
- **A-D Converter**: An analog-to-digital converter is a device that converts analog electric input signal into its equivalent digital signal for further processing in an embedded system.
- **Processor & ASICs**: Processor used for processing the signal and data to execute desired set of instructions with high-speed of operation.Application specific integrated circuit (ASIC) is an integrated circuit designed to perform task specific operation inside an embedded system.

- **D-A Converter**: A digital-to-analog converter is a device that converts digital electric input signal into its equivalent analog signal for further processing in an embedded system.
- **Actuators**: Actuators is a comparator used for comparing the analog input signal level to desired output signal level for providing the error free output from the system.

DESIGN STEPS REQUIRED FOR THE DEVELOPMENT OF EMBEDDED SYSTEM

Designing steps required for embedded system are different from the design process of another electronic system.

Let's see a flow chart represent the design steps required in the development of an embedded system:

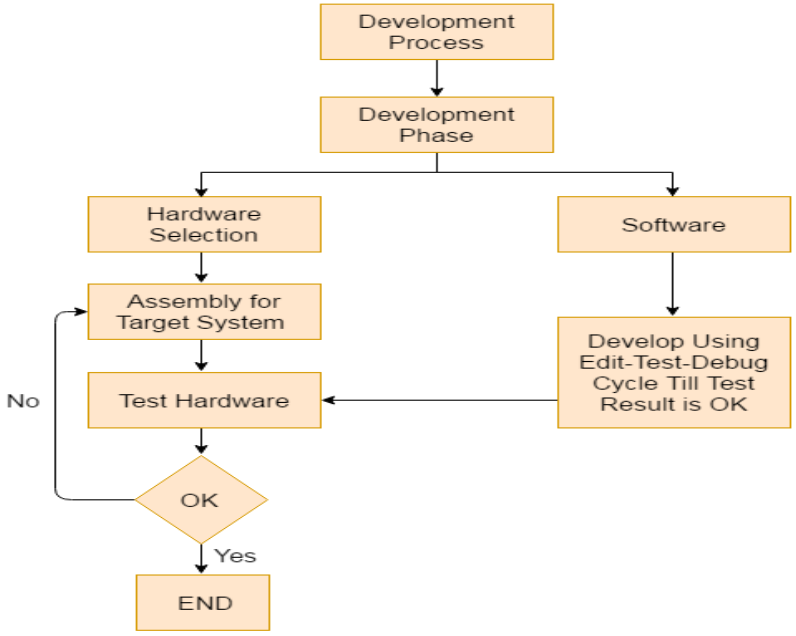


Fig 4.2.1.6: FLOW CHART REPRESENTATION OF DESIGN STEPS

EMBEDDED SYSTEM TOOLS AND PERIPHERALS

Compiler

Compiler is used for converting the source code from a high-level programming language to a low-level programming language. It converts the code written in high level programming language into assembly or machine code. The main reason for conversion is to develop an executable program.

Let's see the operations performed by compiler are:

- Code generation
- Code optimization
- Parsing
- Syntax direct translation
- Preprocessing

Cross-Compiler

If a program compiled is run on a computer having different operating system and hardware configuration than the computer system on which a compiler compiled the program, that compiler is known as cross-compiler.

Decompiler

A tool used for translating a program from a low-level language to high-level language is called a decompiler. It is used for conversion of assembly or machine code to high-level programming language.

Assembler

Assembler is embedded system tool used for translating a computer instruction written in assembly language into a pattern of bits which is used by the computer processor for performing its basic operations. Assembler creates an object code by translating assembly language instruction into set of mnemonics for representing each low-level machine operation.

DEBUGGING TOOLS IN AN EMBEDDED SYSTEM

Debugging is a tool used for reducing the number of error or bugs inside a computer program or an assembled electronic hardware.

Let's see the different debugging tools used in embedded system are

Simulators

Simulator is a tool used for simulation of an embedded system. Code tested for microcontroller unit by simulating code on the host computer. Simulator is used for model the behavior of the complete microcontroller in software.

MICROCONTROLLER STARTER KIT

For developing an embedded system based project a complete microcontroller starter kit is required. The major advantage of this kit over simulator is that they work in real-time operating condition. Therefore it allows the easy input/output functional verification. Consider a microcontroller starter kit consists of:-

- Hardware Printed Circuit Board (PCB)
- In-System Programmer (ISP)
- Some embedded system tools like compiler, assembler, linker, etc
- Sometimes, there is a requirement of an Integrated Development Environment (IDE)

The above component available in microcontroller starter kit is completely enough and the cheapest option available for developing simple microcontroller projects.

Emulators

An emulator is a software program or a hardware kit which emulates the functions of one computer system into another computer system. Emulators have an ability to support closer connection to an authenticity of the digital object.

PERIPHERAL DEVICES IN EMBEDDED SYSTEMS

Communication of an embedded system with an outside environment is done by using different peripheral devices as a combination with microcontroller.

Let's see the different peripheral devices in embedded system are:-

- Universal Serial Bus (USB)
- Networks like Ethernet, Local Area Network(LAN) etc
- Multi Media Cards (SD Cards, Flash memory, etc)
- Serial Communication Interface (SCI) like RS-232, RS-485, RS-422, etc
- Synchronous Serial Communication Interface like SPI, SSC and ESSI
- Digital to Analog/ Analog to Digital (DAC/ADC)
- General Purpose Input/Output (GPIO)
- Debugging like In System Programming (ISP), In Circuit Serial Programming (ICSP), BDM Port, etc

4.2.2 ARDUINO SOFTWARE IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Fig 4.2.2.1: ARDUINO IDE SOFTWARE

WRITING SKETCHES

Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension **.ino**. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

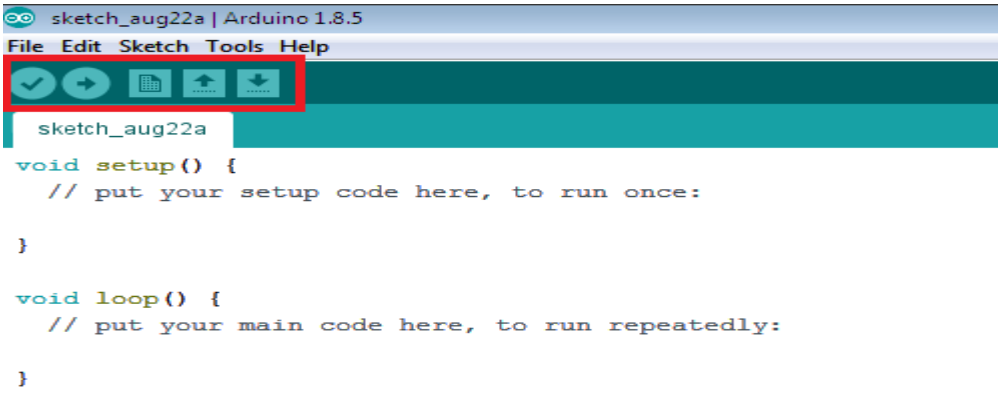


Fig 4.2.2.2: ARDUINO IDE SOFTWARE WRITING SKETCH

	*Verify* Checks your code for errors compiling it.
	*Upload* Compiles your code and uploads it to the configured board. See uploading below for details. Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"
	*New* Creates a new sketch.
	*Open* Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content. Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the **File | Sketchbook**menu instead.

	Save Saves your sketch.
	Serial Monitor Opens the serial monitor.

Table 4.2.1: ARDUINO IDE SOFTWARE WRITING SKETCH FUNCTION

Additional commands are found within the five menus: **File**, **Edit**, **Sketch**, **Tools**, and help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

FILE

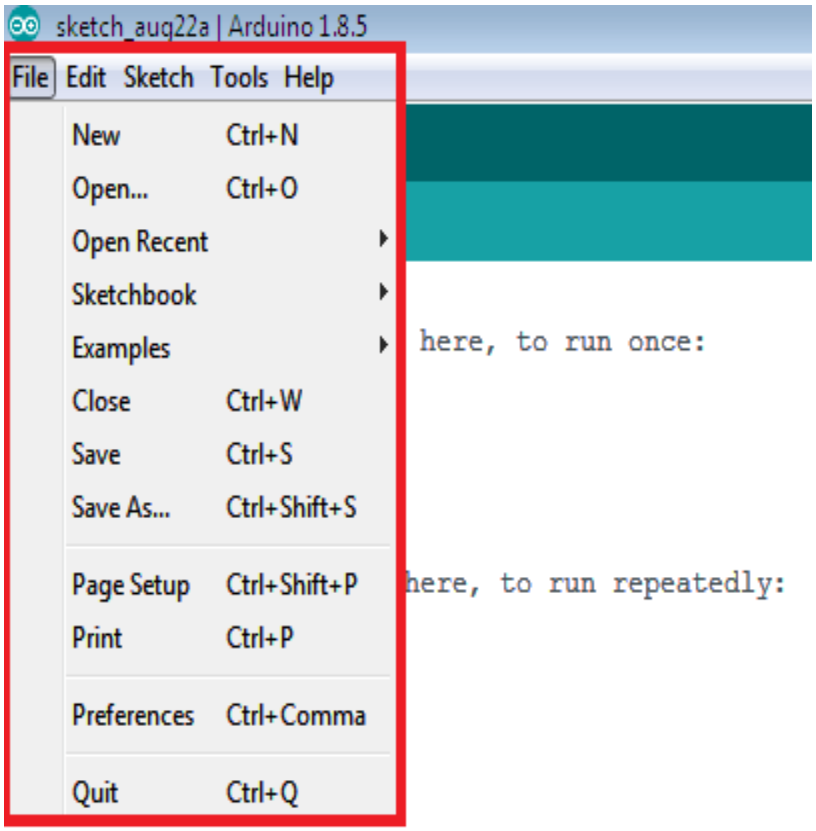


Fig 4.2.2.3: ARDUINO IDE SOFTWARE FILE

- *New*

  Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- *Open*

  Allows loading a sketch file browsing through the computer drives and folders.
- *OpenRecent*

  Provides a short list of the most recent sketches, ready to be opened.
- *Sketchbook*

  Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- *Examples*

  Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- *Close*

  Closes the instance of the Arduino Software from which it is clicked.
- *Save*

  Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
- *Saveas...*

  Allows saving the current sketch with a different name.
- *PageSetup*

  It shows the Page Setup window for printing.
- *Print*

  Sends the current sketch to the printer according to the settings defined in Page Setup.

- *Preferences*

  Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.

- *Quit*

  Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

EDIT

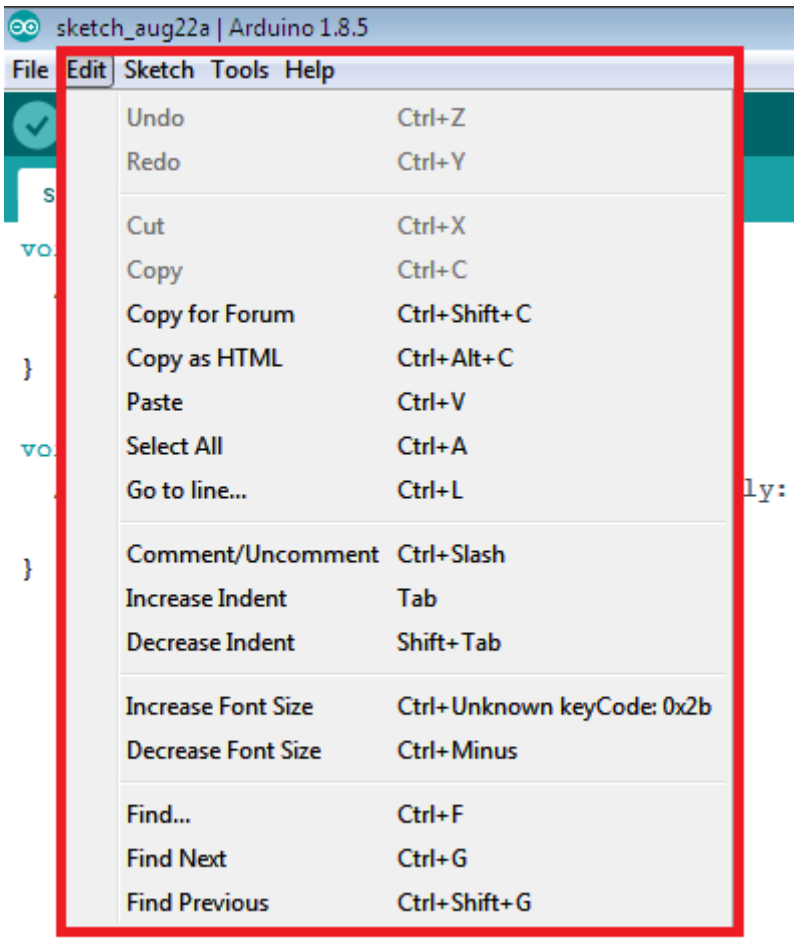


Fig 4.2.2.4: ARDUINO IDE SOFTWARE EDIT

- *Undo/Redo*

  Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.

- *Cut*

  Removes the selected text from the editor and places it into the clipboard.
- *Copy*

  Duplicates the selected text in the editor and places it into the clipboard.
- *CopyforForum*

  Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- *CopyasHTML*

  Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- *Paste*

  Puts the contents of the clipboard at the cursor position, in the editor.
- *SelectAll*

  Selects and highlights the whole content of the editor.
- *Comment/Uncomment*

  Puts or removes the // comment marker at the beginning of each selected line.
- *Increase/DecreaseIndent*

  Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- *Find*

  Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

- *FindNext*

  Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

- *FindPrevious*

  Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

SKETCH

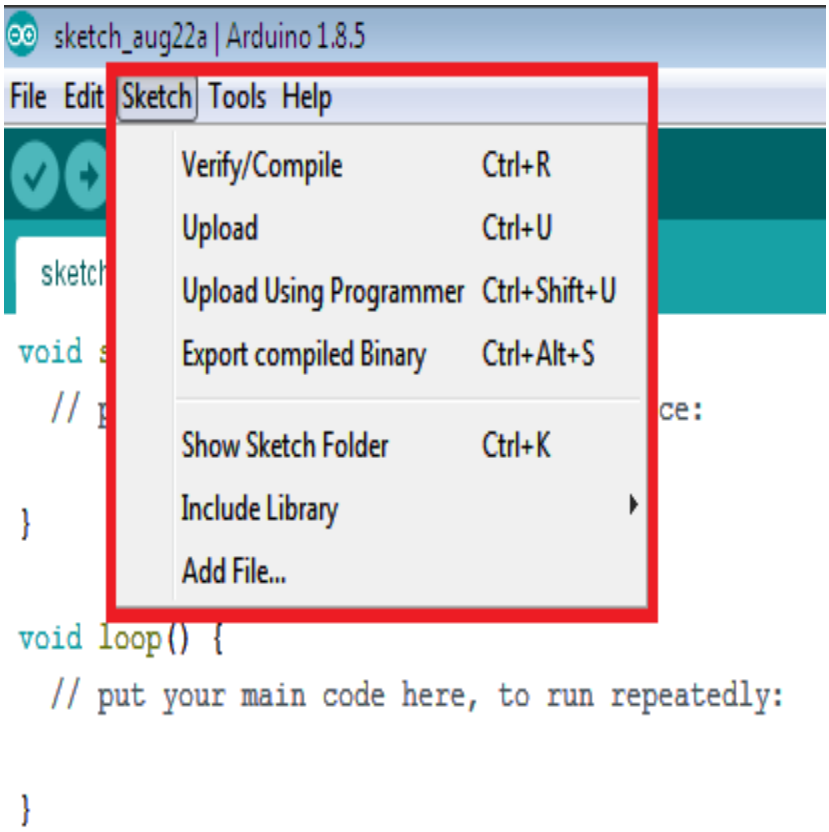


Fig 4.2.2.5: ARDUINO IDE SOFTWARE SKETCH

- *Verify/Compile*

  Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

- *Upload*

  Compiles and loads the binary file onto the configured board through the configured Port.

- *Upload Using Programmer*

  This will overwrite the boot loader on the board; you will need to use Tools > Burn Boot loader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a *Tools -> Burn Bootloader* command must be executed.

- *Export Compiled Binary*

  Saves a .hex file that may be kept as archive or sent to the board using other tools.

- *Show Sketch Folder*

  Opens the current sketch folder.

- *Include Library*

  Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

- *Add File...*

  Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

TOOLS

- *Archive Sketch*

  Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

- *Auto Format*

  This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.

- *Fix Encoding & Reload*

  Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

- *Serial Monitor*

  Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

- *Board*

  Select the board that you're using. See below for descriptions of the various boards.

- *Port*

  This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

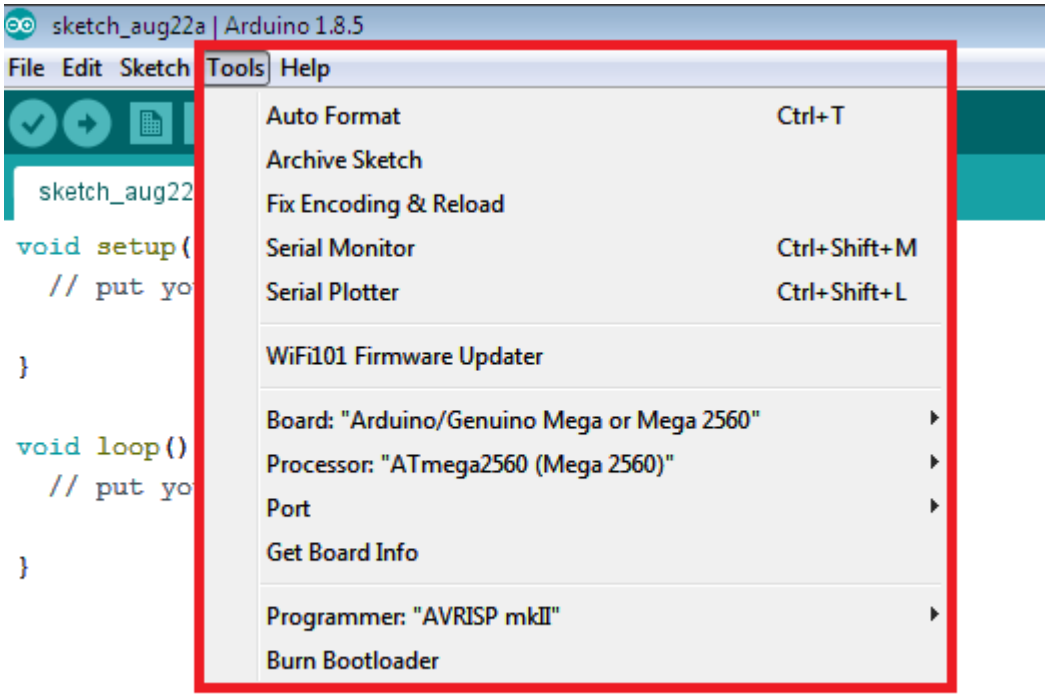


Fig 4.2.2.6: ARDUINO IDE SOFTWARE TOOLS

- *Programmer*

  For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a boot loader to a new microcontroller, you will use this.

- *Burn Boot loader*

  The items in this menu allow you to burn a boot loader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally comes without a boot loader). Ensure that you've selected the correct board from the **Boards** menu before burning the boot loader on the target board. This command also set the right fuses.

Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

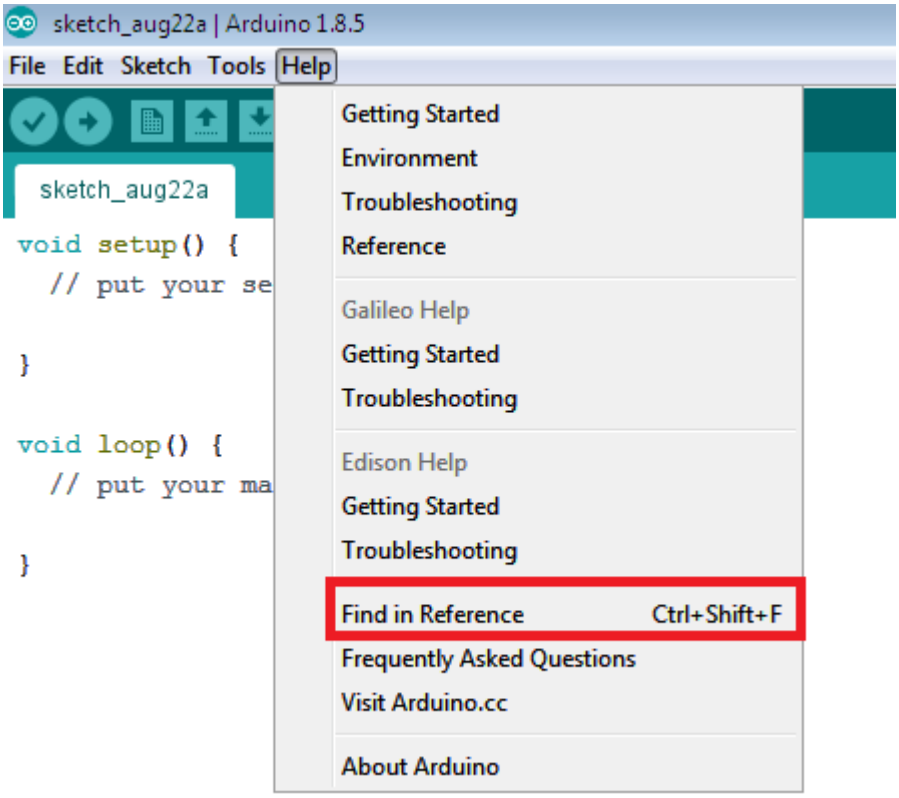


Fig 4.2.2.7: ARDUINO IDE SOFTWARE HELP

- *FindinReference*

  This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

SKETCHBOOK

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the **File > Sketchbook** menu or from the **Open** button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the **Preferences** dialog.

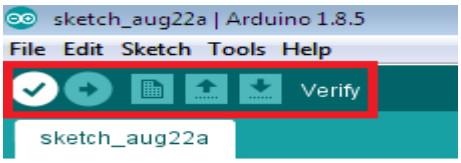


Fig 4.2.2.8: ARDUINO IDE SOFTWARE SKETCHBOOK

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

TABS, MULTIPLE FILES, AND COMPILATION:

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

UPLOADING

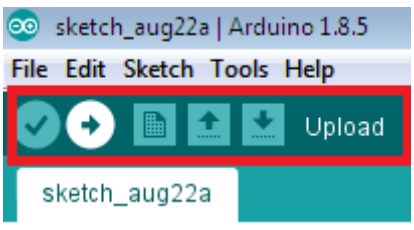


Fig 4.2.2.9: ARDUINO IDE SOFTWARE FILE UPLOADS

Before uploading your sketch, you need to select the correct items from the **Tools > Board** and **Tools > Port** menus. The boards are described below. On the Mac, the serial port is probably something like **/dev/tty.usbmodem241** (for an Uno or Mega2560 or Leonardo) or **/dev/tty.usbserial-1B1** (for a Duemilanove or earlier USB board), or **/dev/**tty.USA19QW1b1P1**.1** (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be **/dev/ttyACMx, /dev/ttyUSBx** or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the **Upload** item from the **Sketch** menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino **bootloader**, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The boot loader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

LIBRARIES

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the **Sketch > Import Library** menu. This will insert one or

more **#include** statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its **#include** statements from the top of your code.

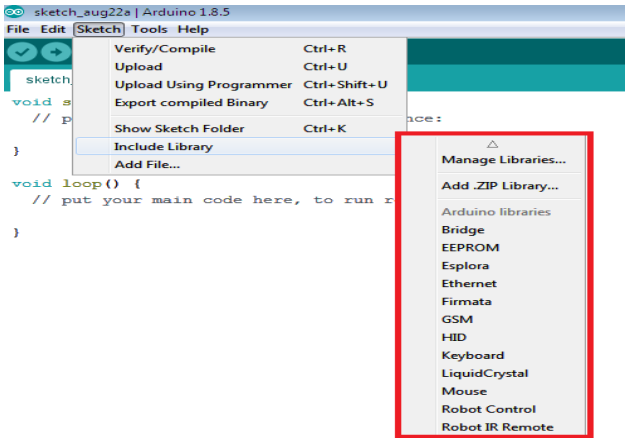


Fig 4.2.2.10: ARDUINO IDE SOFTWARE LIBRARIES

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

THIRD-PARTY HARDWARE

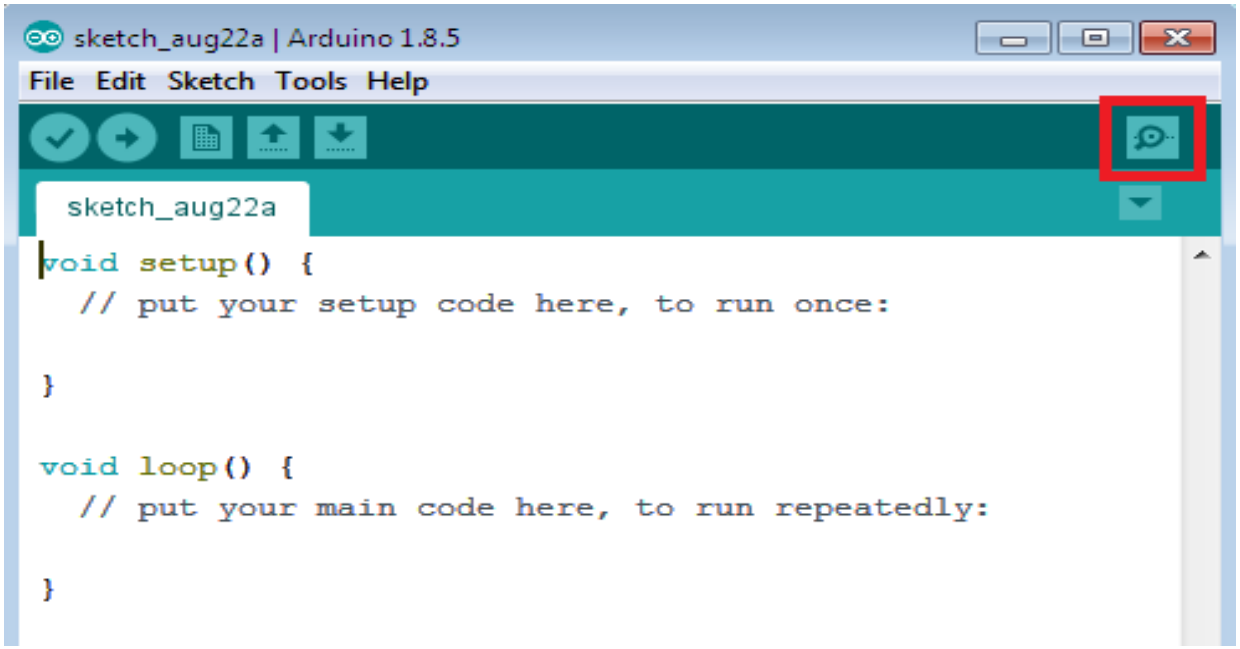
Support for third-party hardware can be added to the **hardware** directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the **hardware** directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

SERIAL MONITOR

This displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to **Serial.begin** in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).



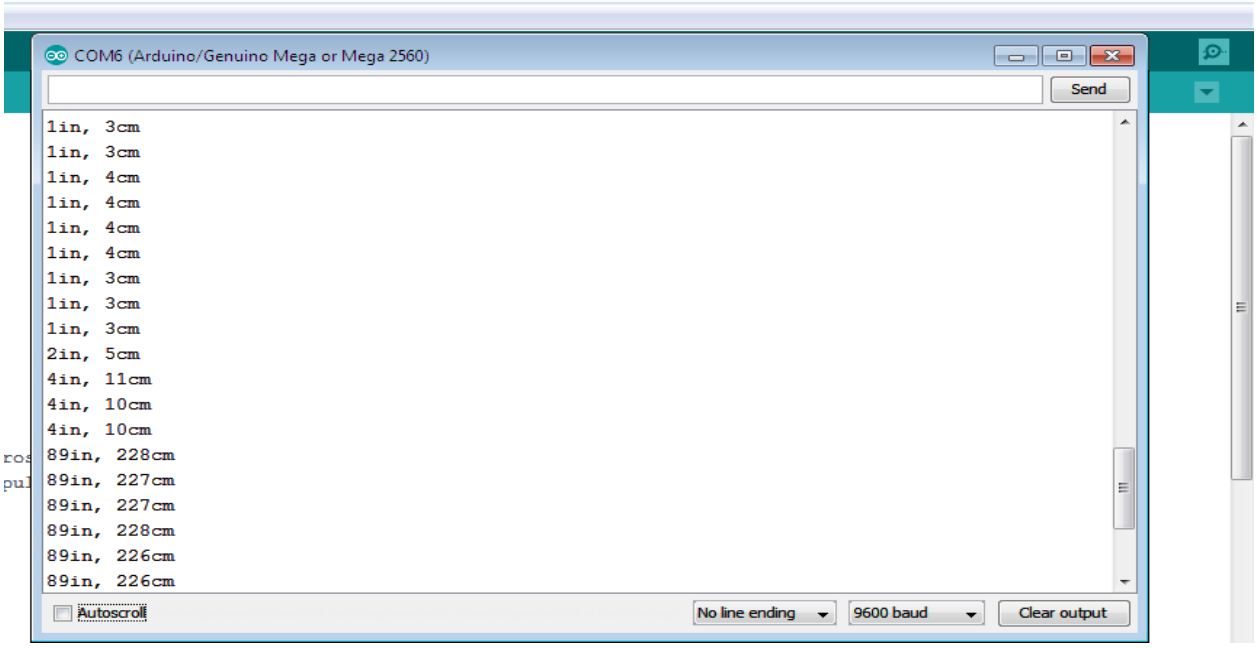


Fig 4.2.2.11: ARDUINO IDE SOFTWARE CODE AND IT EXECUTION

PREFERENCES

Some preferences can be set in the preferences dialog (found under the **Arduino** menu on the Mac, or **File** on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

LANGUAGE SUPPORT

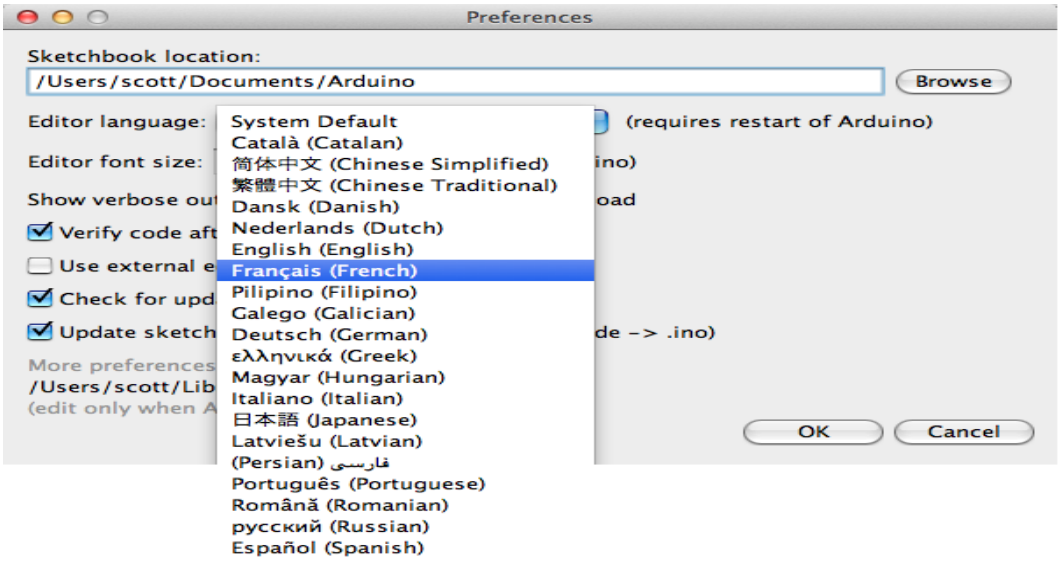


Fig 4.2.2.12: ARDUINO IDE SOFTWARE LANGUAGE SUPPORT

Since version 1.0.1, the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.)

If you would like to change the language manually, start the Arduino Software (IDE) and open the **Preferences** window. Next to the **Editor Language** there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English.

You can return the software to its default setting of selecting its language based on your operating system by selecting **System Default** from the **Editor Language** drop-down. This setting will take effect when you restart the Arduino Software (IDE). Similarly, after changing your operating system's settings, you must restart the Arduino Software (IDE) to update it to the new default language.

4.2.3 JUPYTER NOTEBOOK

The Jupyter Notebook is an incredibly powerful tool for interactively developing and presenting data science projects. This article will walk you through how to use Jupyter Notebooks for data science projects and how to set it up on your local machine.

RUNNING JUPYTER

Fig 4.2.3 :JUPYTER NOTEBOOK INTERFACE

On Windows, you can run Jupyter via the shortcut Anaconda adds to your start menu, which will open a new tab in your default web browser that should look something like the following screenshot.

This isn't a notebook just yet, but don't panic! There's not much to it. This is the Notebook Dashboard, specifically designed for managing your Jupyter Notebooks. Think of it as the launchpad for exploring, editing and creating your notebooks. Be aware that the dashboard will give you access only to the files and sub-folders contained within Jupyter's start-up directory (i.e., where Jupyter or Anaconda is installed). However, the start-up directory can be changed. It is also possible to start the dashboard on any system via the command prompt (or terminal on Unix systems) by entering the command jupyter notebook; in this case, the current working directory will be the start-up directory.

With Jupyter Notebook open in your browser, you may have noticed that the URL for the dashboard is something like https://localhost:8888/tree. Localhost is not a website, but indicates that the content is being served from your local machine: your own computer.

COMPONENTS OF JUPYTER NOTEBOOK

There are the following three components of Jupyter Notebook –

1. The notebook web application: It is an interactive web application for writing and running the code.

    - The notebook web application allows users to:Edit code in the browser with automatic syntax highlighting and indentation.
    - Run code on the browser.
    - See results of computations with media representations, such as HTML, LaTex, png, pdf, etc.
    - Create and use JavaScript widgets.
    - Includes mathematical equations using Markdown cells.

2. Kernels: Kernels are the separate processes started by the notebook web application that is used to run a user's code in the given language and return output to the notebook web application.

    In Jupyter notebook kernel is available in the following languages:

    - Python
    - Julia
    - Ruby
    - R
    - Scala
    - node.js
    - Go

3. Notebook documents: Notebook document contains a representation of all content which is visible in the notebook web application, including inputs and outputs of the computations, text, mathematical equations, graphs, and images.

4.3.3.1 FACE DETECTION

OpenCV is a popular computer vision library that supports deep learning-based tasks, and it provides a new module called DNN (Deep Neural Networks) that facilitates the integration of pre-trained deep learning models. The face recognition library is an extension of OpenCV that enables the recognition of human faces.

The face recognition process involves several steps, including face detection, face alignment, and face recognition. The first step is to detect the face in the image, which involves identifying the location of the facial features such as eyes, nose, and mouth. The second step is to align the face to a standard pose, which involves transforming the image to a fixed coordinate system. The final step is to recognize the face by comparing it to the face data stored in the database.

The face data stored in the database is obtained during the registration process, where the face of the voter is captured and stored in the database alongside other biometric information such as fingerprints and RFID data. During the voting process, the captured face image is compared to the face data stored in the database to ensure that the voter is authorized to cast their vote.

Overall, the use of face authentication in the e-voting machine provides an additional layer of security and prevents fraudulent voting, ensuring the integrity and accuracy of the voting process.

CHAPTER 5

RESULT

5.1 OUTPUT

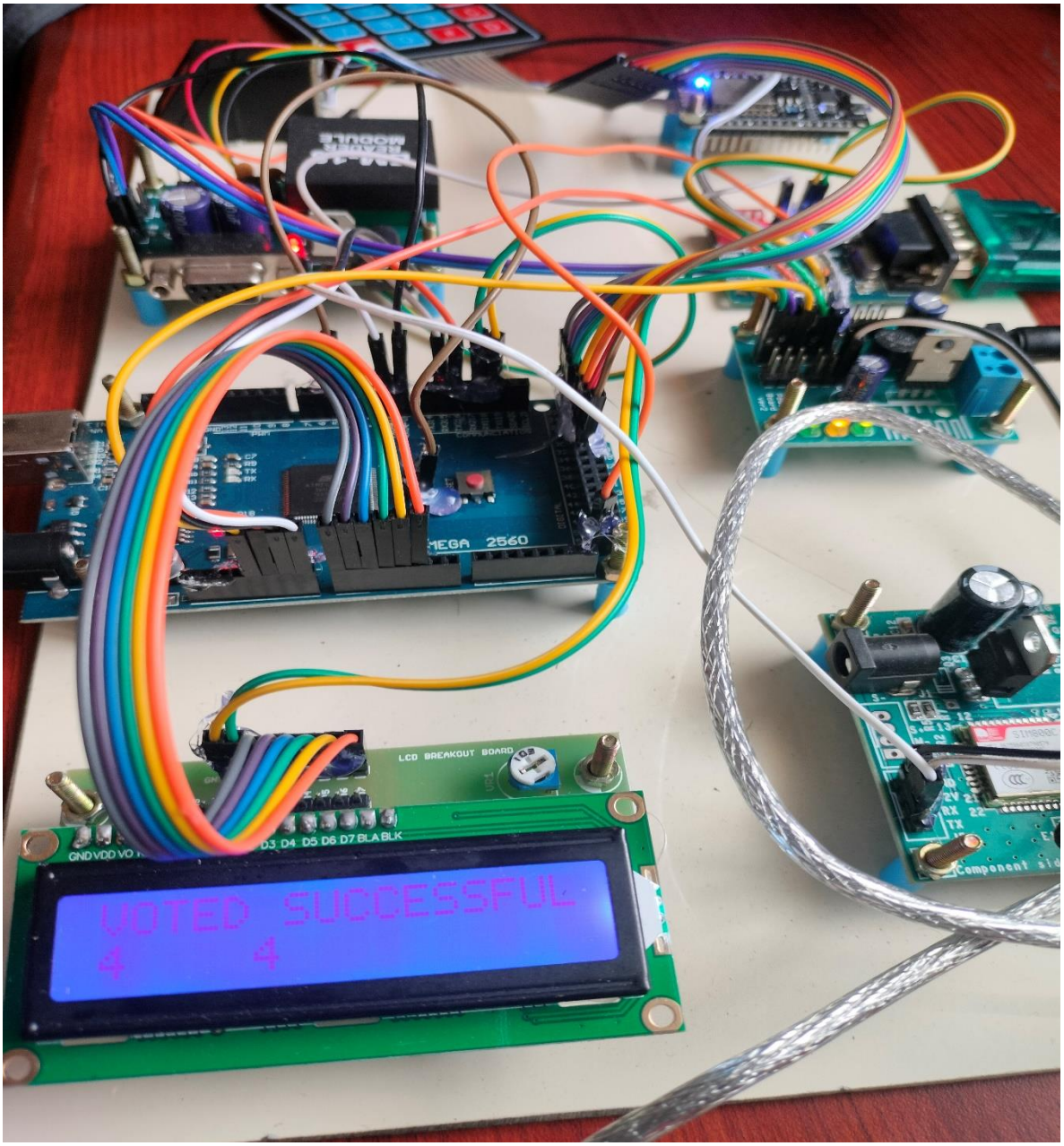


Fig 5.1: OUTPUT OF DESIGN AND IMPLEMENTATION OF SECURED E-VOTING MACHINE USING OPENCV WITH DEEP LEARNING

The authentication processes used in the e-voting machine, including RFID, fingerprint sensing, face recognition, and IoT authentication, play a crucial role in ensuring the integrity and accuracy of the voting process. By verifying the identity of the voter before allowing them to cast their vote, the system can prevent fraudulent voting and ensure that each vote is counted accurately.

In particular, face authentication plays a crucial role in enhancing the security of the voting process. By comparing the face of the voter to the face data stored in the database, the system can verify that the person attempting to cast their vote is authorized to do so. This helps to prevent instances of identity theft and other forms of voter fraud.

Similarly, the use of RFID and fingerprint sensing ensures that only authorized individuals are allowed to access the e-voting machine, preventing unauthorized access and ensuring the security of the system.

Once the voter has cast their vote, the use of GSM technology to send an SMS confirmation message provides an additional layer of transparency and accountability. By receiving a confirmation message on their registered mobile number, the voter can be confident that their vote has been cast successfully, and the system provides a reliable way of tracking the voting process.

Overall, the authentication processes and the use of GSM technology in the e-voting machine are essential components of the system, ensuring the integrity and accuracy of the voting process and providing transparency and accountability to the voters.

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

The main outcome of this project is to improve the security in the voting system. The security improvement is attained by the various authentication processes and finally polling by the biometric of the voters. As said the votes of the individual can only be casted by them and it can't be illegally casted by others as the biometric plays a vital role. The election officer and the polling officer are given separate authentication to start the polling process. By all the above processes the security has been increased in the voting system

6.2 FUTURE ENHANCEMENT

- This system can be connected in cloud, so that all the polling booths operation are monitored through authorised personnel's.
- In future polling can be done online by implementing various security measures which can be convenient for the voters.

CHAPTER 7

REFERENCES

7.1 REFERENCES

[1]. D. Ashok Kumar, T. Ummal Sariba Begum, Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering, March 21-23, 2012. "Electronic Voting Machine – A Review."

[2]. Ankita R. kasliwal, Jaya S. Gadekar, Manjiri A. Lavadkar, Pallavi K. Thorat, Dr. Prapti Desmukh,"Aadhar Based Election Voting Syestem" ,IOSR- Journal of Computer and Engineering),[p-ISSN2394-9333],PP 18-21.

[3]. Thiruthanigesan Kanagasabai et al,"Fingerprint voting system using Arduino,Middle-East Journal of Scientific Research,25 (8): 1793-1802,2017.

[4]. N.N.Nagamma and et al. "Aadhar based fingerprint EVM system", International Journal of Electronics Engineering Research,ISSN 0975- 6450 vol.9,No.6(2017) pp. 923-930.

[5]. Deepika Iswarya, Rathna Prabha, Trini Xavier, "A Survey on E-Voting System Using Arduino Software", *International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization)*, vol. 5, no. 2, pp. 687-690, February 2016.

[6]. D. Bowen. \Top-to-Bottom" Review (TTBR) of voting machines certi_ed for use in California. California Secretary of State, Aug. 2007. http://sos.ca.gov/elections/elections vsr.htm.

[7]. J. Brunner. Evaluation & Validation of Election-Related Equipment, Standards & Testing (EVEREST). Ohio Secretary of State, Dec. 2007. http://www.sos.state.oh.us/SOS/Text.aspx?page=4512.

[8]. Bundesverfassungsgericht (German Constitutional Court). Judgment [...] 2 BvC 3/07, 2 BvC 4/07, o_cial English translation. Mar. 3, 2009. http://www.bverfg.de/ entscheidungen/rs20090303 2bvc000307en.html.

[9]. K. Butler, W. Enck, H. Hursti, S. McLaughlin, P. Traynor, and P. McDaniel. Systemic issues in the Hart InterCivic and Premier voting systems: Reections on Project EVEREST. In Proc. EVT, San Jose, CA, July 2008.

[10]. J. A. Calandrino, A. J. Feldman, J. A. Halderman, D. Wagner, H. Yu, and W. P. Zeller. Source code review of the Diebold voting system. Part of California TTBR, Aug. 2007.