USER FRIENDLY AI HARDWARE FOR EV CHARGING USING RFID

A PROJECT REPORT

Submitted by

BALA SRINIVAS S	**111719106017**
YOGAVARDHAN C	**111719106026**
CHINNI SUMAN CHAITANYA	**111719106027**
OMMINA MASTHAN MURALI	**111719106106**

In partial fulfillment for the award of

the degree of

BACHELOR OF ENGINEERING

in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

R.M.K. ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601206.

ANNAUNIVERSITY: CHENNAI-600025

MARCH 2023

R.M.K. ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601206.

BONAFIDE CERTIFICATE

Certified that this project report **"USER FRIENDLY AI HARDWARE FOR EV CHARGING USING RFID"** "is the bonafide work of" **SUMANCHAITANYA CHINNI (111719106027), OMMINA MASTHAN MURALI (111719106106), BALA SRINIVAS.S (111719106017), CHINNAGURAVAGARI YOGAVARDHAN (111719106026)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.T.Suresh,M.E.,Ph.D.,	**Dr.Sumitra V,M.E.,Ph.D.**
Professor & **Head of The Department**	**Supervisor &Associate Professor**
Electronics & Communication Engineering	Electronics & Communication Engineering
R.M.K. Engineering College	R.M.K. Engineering College
Kavaraipettai - 601206.	Kavaraipettai - 601206.

Submitted for the Project Viva –Voce held on……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We would like to express our heartfelt thanks to the Almighty, our beloved parents for their blessings and wishes for successfully doing this project.

We convey our thanks to Chairman **Thiru R.S. Munirathinam** and Vice-Chairman **Thiru R.M. Kishore** who took keen interest on us and encouraged throughout the course of study and for their kind attention and valuable suggestions offered to us. We express our sincere gratitude to our Principal. **Dr. K.A. Mohamed Junaid M.E., Ph.D** for fostering an excellent climate to excel.

We are extremely thankful to **Dr. T. Suresh M.E., Ph.D,** Professor and Head, Department of Electronics and Communication Engineering, for having permitted us to carry out this project effectively.

We convey our sincere thanks to our mentor, skillful and efficient supervisor **Dr.Sumitra.V,M.E.,Ph.D.** Professor for her extremely valuable guidance through out the course of project.

We are grateful to our Project Co-coordinators and all the department staff members for their intense support.

ABSTRACT:

The transition to electric mobility is a promising global strategy for de carbonizing the transport sector. India is among a handful of countries that support the global EV30@30 campaign, which targets to have at least 30% new vehicle sales be electric by 2030. The Government of India has instituted various enabling policies to promote the development of the charging infrastructure network. The primary audience for this handbook includes public and private sector stakeholders that are responsible for charging infrastructure implementation, such as electricity distribution companies, municipal corporations, urban development authorities, and charge point solutions providers and operators. This project offers the improvement and validation of the charging station to be access easily using currency based architecture. The existing method EV station give an option to use the station through only the application. And the major problem of existing system is in the process of charging time we cannot disconnect the charger for the emergency purpose. If we disconnect the charger, we lost the particular amount of money. So in this project we proposed currency based return policy EV station to rectify the problems.

KEYWORDS: Deep learning, RFID, Keypad, EV charging.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

ABBREVATION	EXPANSION
IOT	Internet of Things
WSN	Wireless Sensor Network
LCD	Liquid Crystal Display
GSM	Global System for Mobile
SMPS	Switched Mode Power Supply
LED	Light Emitting Diode
RFID	Radio Frequency Identification

CHAPTER: 1 - INTRODUCTION

1.1 GENERAL

The usage of electric vehicles is increasing in many folds. Hence, charging the electric vehicles has become a greater task these days. Battery power is the main concern when buying new electric vehicles. The main purpose of charger is to reduce the wastage of electrical power which often arises due to negligence of the user. Once the currency is inserted, the currency acceptor detects whether the currency is valid or not. For each unit of price, the power is available only for a limited period. The arduino can calculate the time based on the number of currencies inserted.

1.2 SCOPE OF THE PROJECT:

The aim of our project is a currency based EV charging station.

1.3 EXISTING SYSTEM

- In this system, there is the only option to access the device is using card.
- In the process of charging time, we cannot disconnect the charger for the emergency purpose.
- If we disconnect the charger, we lost the particular amount of money.
- The cards are only identifying in previous system.

1.3.1 EXISTING SYSTEM DISADVANTAGES

- There is no option to use physical currency.
- Loss of Money.
- No return policy for the unused charging time.

1.4 LITERATURE SURVEY

Title 1: Deployment of Secure EV Charging System Using Open Charge Point Protocol (2018)

Authors: Binod Vaidya, Hussein T. Mouftah

Description:

A large-scale deployment of Electric Vehicles (EVs) not only provides mobility paradigm shift but also demands new requirements in the information and control components of the electric power grid. Some of barriers to the EV adoption could be lowered if Smart cities make available sufficient public EV Charging systems such that they facilitate EVs to be charged in those charging systems. In order to provide interoperability and reduce costly maintenance, open standards such as Open Charge Point Protocol (OCPP) are desired for communication in EV Charging systems. In this paper, we depict a smart management system for community-wide and public EV charging infrastructures in the Smart city known as Sec Charge System as well as its implementation using OCPP.

Title 2: IOT Based PV assisted EV Charging Station for Confronting Duck Curve (2018)

Authors: Badrinath Kulkarni, Devaji Patil, Prof Rahul.G. Suryavanshi

Description:

In this paper, we study the effects of electricity production from a solar power plant on the load curve also known as duck curve and provide an alternative by charging electric vehicles in the IOT integrated multi-level charging stations. An effort is made to improvise the load curve by neutralizing the dip and sudden rise in the duck curve by creating alternative loading on the powergrid. It also supports for the promotion of EV utilization by improvising the charging technologies with the help of IOT interface and providing incentives to customers to utilize

EVCS at the workplace. It also talks about an idea of replacing fossil fuel based revenue system with centralized EV charging taxation that would boost the idea of green mobility.

Title 3: Overview of Storage System and EV Aggregation Charging Behavior in Energy Market (2018)

Authors: Salman, Ai Xin, Arsalan Masood, Mishkat Ullah Jan

Description:

The Electrical vehicle (EV) technology is gaining popularity; it is because of many factors like reduction in carbon and green gas and usage of vehicle batteries as a storage element in the power system. However, the increasing usage of auxiliary loads i.e. EV and heat pumps creates adverse effects on the grid. Flexibility of power consumption due to uncertainty in modern day grid requires some storage elements. In this paper an overview of modern day grid is presented. Moreover, the charging scheduling of Aggregators EVs battery is addressed. The mathematical formulation for generalization of EV, battery energy storage system (BESS) is presented with the constraints. Finally, the case study for multi aggregator charging schedule generation is presented which can also be extended to a generalized modelling of storage system with different new energy sources in the future. The required results and the proposed mathematical modelling show the effectiveness of the solution towards a more modern day grid friendly environment.

Title 4: Deployment of Secure EV Charging System Using Open Charge Point Protocol (2019)

Authors: Binod Vaidya, Hussein T. Mouftah

Description:

A large-scale deployment of Electric Vehicles (EVs) not only provides mobility paradigm shift but also demands new requirements in the information and control components of the

electric power grid. Some of barriers to the EV adoption could be lowered if Smart cities make available sufficient public EV Charging systems such that they facilitate EVs to be charged in those charging systems. In order to provide interoperability and reduce costly maintenance, open standards such as Open Charge Point Protocol (OCPP) are desired for communication in EV Charging systems. In this paper, we depict a smart management system for community-wide and public EV charging infrastructures in the Smart city known as Sec Charge System as well as its implementation using OCPP.

Title 5: Currency Based Mobile Battery Charger With High Security (2019)

Authors: Aliae Squalli Houssaini, Hassan Qjidaa

Description:

Recently, driver hypo vigilance (drowsiness and fatigue) becomes one of the principal causes of traffic crashes, it can prompt many deaths, wounds and many economic losses. Therefore, the use of a system that takes into account the driver's level of vigilance can play an important role in preventing accidents and saving human lives. In this work, we propose a non-intrusive driver hypo vigilance detection system in real-time. This system makes it possible to detect drowsiness by the identification of Micro Sleep corresponding to a sleepiness of more than 2 seconds through the analysis of eye-closure, and to identify fatigue by the analysis of the movement of the mouth to detect yawning. In case of drowsiness or fatigue, an alert is launched to make the driver vigilant and thus definitely avoid road accidents, decrease the percentage of murders and injuries caused by driver hypo vigilance, then save many human lives. Experiments were conducted in real time to evaluate the proposed approach.

1.5 PROPOSED SYSTEM

- In this system, we can provide the support for the card and physical currency (CURRENCY).
- Also given the return policy for the unused charging time in the previous usage time.

1.5.1 PROPOSED SYSTEM ADVANTAGES:

Option to use card and carless, No Loss Money, Return policy for un used charging.

CHAPTER: 2

PROJECT DESCRIPTION

2.1 GENERAL

The currency based mobile charging system charges the electric vehicles when the currency is inserted. This system is used by shop owners, rural people and can be implemented in the public places like railway stations , bus stands to provide mobile charging facilities. So, the currency acceptor recognizes valid currencies and then signals the arduino for further action. If a valid currency is found, it signals the arduino and then the arduino starts the mobile charging mechanism providing a 5V supply through a power supply section to the mobile phone. The arduino starts a reverse countdown timer to display the charging time for that mobile phone.

2.2 BLOCK DIAGRAM

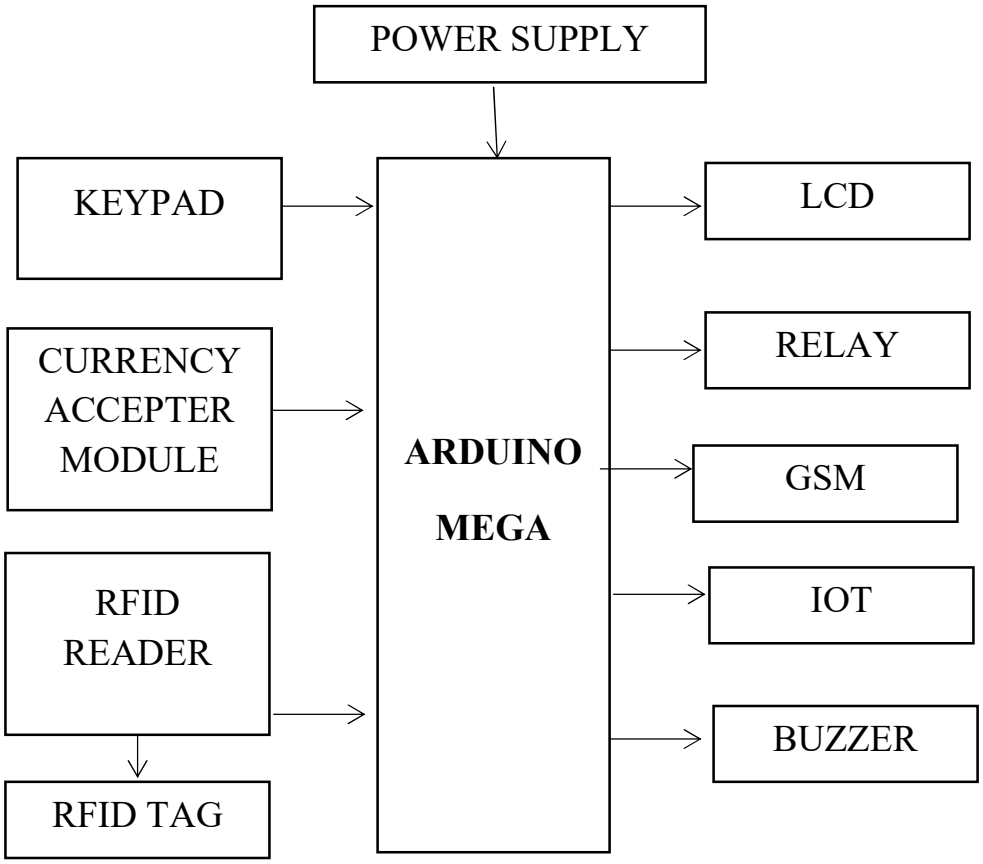


Fig 2.2 BLOCK DIAGRAM

2.3 MODULES NAME

- CURRENCY ACCEPTER MODULE WORKING
- IDENTIFICATION

2.4 MODULE DESCRIPTION

2.4.1CURRENCYACCEPTER MODULE WORKING

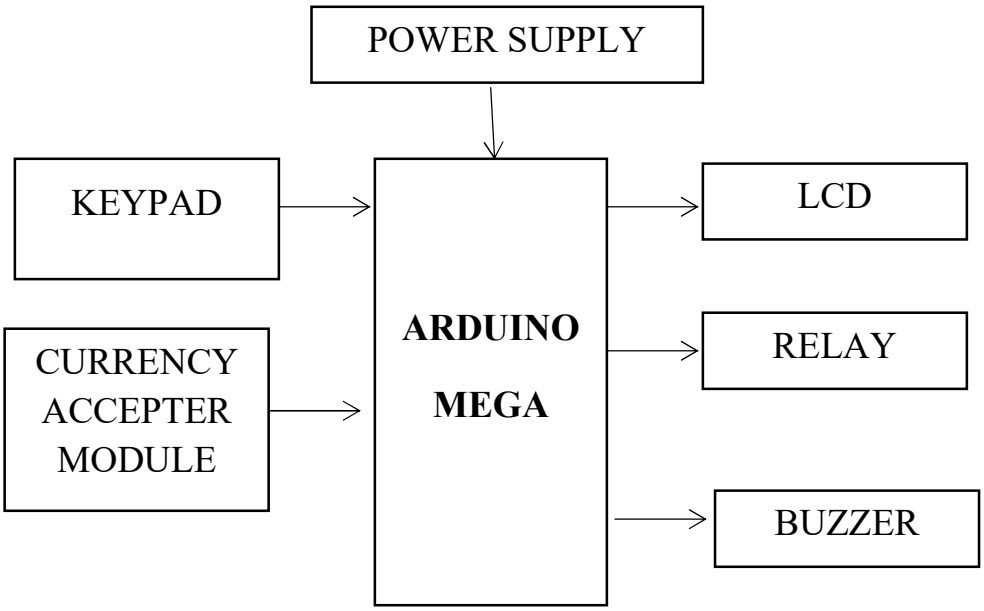


Fig 2.4.1 CURRENCY ACCEPTER MODULE

This Currency Acceptor works by comparing the currency that already inserted in this sensor (blue color at sensor) with the currency that is inserted at the front hole. If the currency is the same, the currency will be accepted, but if currency is different, the currency will be rejected. Keypad used set timing for charging

2.4.2 IDENTIFICATION

Using Rfid to identify the next time you charge, every time remaining amount will be saved Rfid card every time. So, we use that money as alternate time for charging.

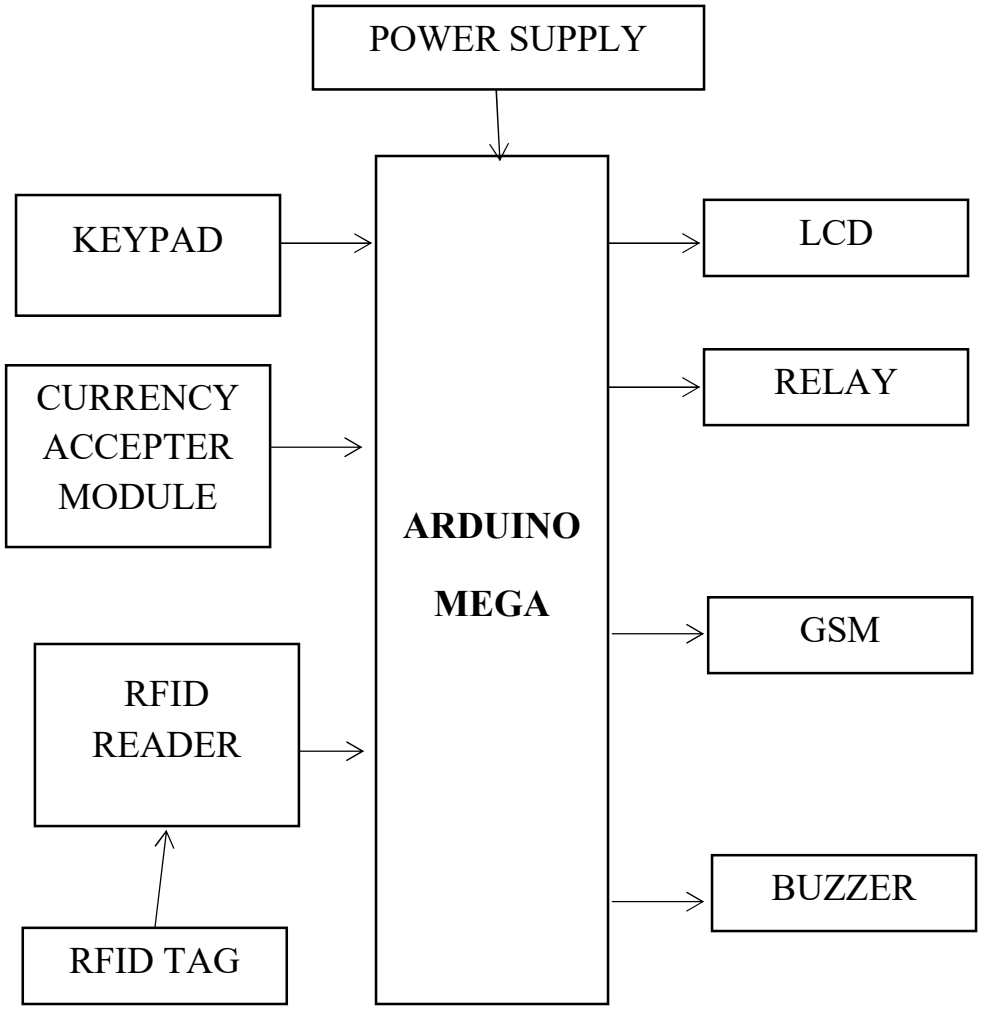


FIG 2.4.2 IDENTIFICATION MODULE

2.5 BLOCK DIAGRAM WORKING

In this project, we have used an arduino uno micro controller which acts as the brain of our system; hence the entire system program is stored in it. Here the use of the RFID and currency accepter module can give the two options for the payment purpose. The RFID is used to make a payment using a card and the currency accepter module is used to make a payment card less. The keypad is used to give the access for the owner of the power station and another option is to change the output timing for a currency. GSM module is used to update payment information for the user. Buzzer gives the alert for the start and end of the charging time. Whenever the person completed the payment relay will be on and charging the vehicle. The LCD is used to update all the information of the device. And update it into the IOT webpage.

CHAPTER: 3

HARDWARE AND SOFTWARE DESCRIPTION

3.1 HARDWARE DESCRIPTION:

3.1.1 ARDUINO MICROCONTROLLER

3.1.1.1 INTRODUCTION TO 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.

3.1.1.2 ARDUINO MEGA:

The UNO 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 uno 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.

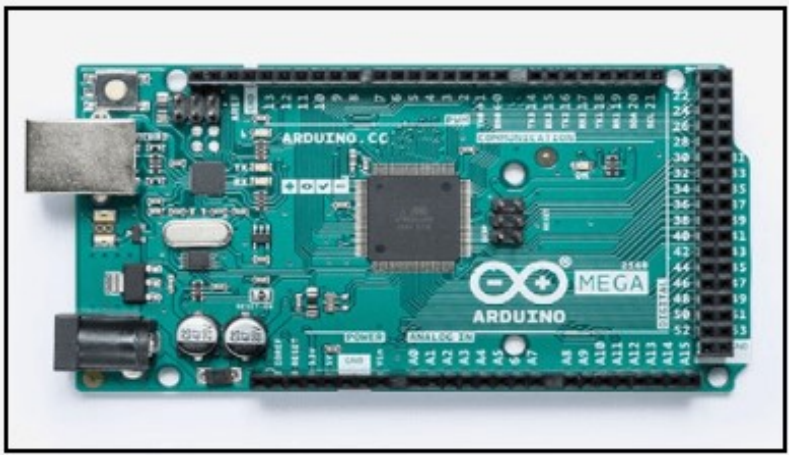


Fig 3.1.1.2 Arduino Mega

3.1.1.3 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

3.1.1.4 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.

Most arduino boards consist of an Atmel 8-bit AVR micro controller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. 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.

3.1.1.5 PHYSICAL CHARACTERISTICS AND SHIELD COMPATIBILITY:

The maximum length and width of the Mega 2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega 2560 is designed to be compatible with most shields designed for the Uno and the older Diecimila or Duemilanove Arduino boards. Digital pins 0 to13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations.

3.1.1.6 APPLICATIONS:

- Arduboy, a handheld game console based on Arduino.
- Arduinome, a MIDI controller device that mimics the Monome.
- Ardupilot, drone software and hardware.

3.1.2 POWER SUPPLY:

3.1.2.1 GENERAL DESCRIPTION:

An adapter is a device that converts attributes of one electrical device or system to those of an otherwise incompatible device or system. Some modify power or signal attributes, while others merely adapt the physical form of one electrical connector to another. In a computer, an adapter is often built into a card that can be inserted into a slot on the computer's motherboard. The card adapts information that is exchanged between the computer's microprocessor and the devices that the card supports.

3.1.2.2 PRODUCT DESCRIPTION

An electrical Power adapter may enable connection of a power plug, sometimes called, used in on region to a AC power socket used in another, by offering connections for the disperate contact arrangements, while not changing the voltage. An AC adapter, also called a “recharger”, is a small power supply that changes household electric current from distribution voltage to low voltage DC suitable for consumer electronics. Some modify power signal or attributes, while others merely adapt the physical form of one electrical connector to another. For computers and related items, one kind of serial port adapter enables connections.

3.1.2.3 FEATURES

- Output current: 1A
- Supply voltage: 220-230VAC
- Output voltage: 12VDC
- Reduced costs
- Increased value across front-office and back-office functions
- Access to current, accurate, and consistent data
- It generates adapter meta data as WSDL files with J2CA extension.

Fig 3.1.2.2 Power Supply

3.1.2.4 APPLICATIONS:

- Back-end systems which need to send purchase order data to oracle applications send it to the integration service via integration server client.
- SMPS applications.

3.1.3 LCD (Liquid Crystal Display)

3.1.3.1 INTRODUCTION

LCD (Liquid Crystal Display) is a type of flat panel display which uses liquid crystals in its primary form of operation. LEDs have a large and varying set of use cases for consumers and businesses, as they can be commonly found in smart phones, televisions, computer monitors and instrument panel. Pin diagram as shown below.

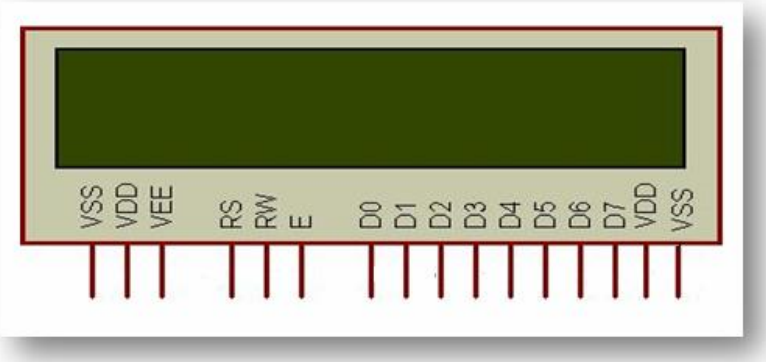


Fig 3.1.3 LCD

3.1.3.2 APPLICATIONS

- LCD module display is economical.
- It has no limitation when it comes to displaying special and custom characters. It is easily programmable.
- It is easily programmable.
- It can be used for high resolution animation.

- It is suitable for environments that brightly lit.
- It is energy efficient.

3.1.4 Keypad Interfacing with the Microcontrollers:

Keyboards are organized in a matrix of rows and columns, and the CPU accesses both rows and columns through ports. With two 8-bit ports, an 8*8 matrix of keys can be connected to a microprocessor. In IBM PC keyboards, a single microcontroller takes care of software and hardware interfacing, and programs stored in the EPROM scan the keys continuously to identify which one has been activated and present it to the motherboard.

Figure shows a 4*4 matrix connected to two ports. 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 3.1.4 Keypad

3.1.5 RELAY:

Relays are the primary protection as well as switching devices in most of the control processes or equipment. All the relays respond to one or more electrical quantities like voltage or current such that they open or close the contacts or circuits. A relay is a switching device as it works to isolate or change the state of an electric circuit from one state to another.

Fig 3.1.5.1 RELAY

Classification or the types of relays depend on the function for which they are used. Some of the categories include protective, reclosing, regulating, auxiliary and monitoring relays.

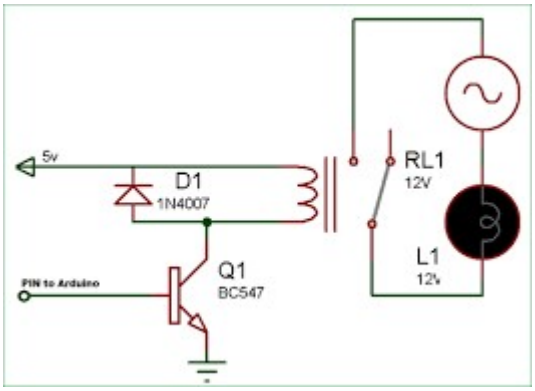


Fig 3.1.5.2 RELAY CIRCUIT

3.1.6 RFID (RADIO-FREQUENCY IDENTIFICATION)

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically stored information. Passive tags collect energy from a nearby RFID reader's interrogating radio waves.

RFID is similar to bar coding in that data from a tag or label are captured by a device that stores the data in a database. It has several advantages over barcode asset tracking software, such as being able to read outside the line-of-sight and being embedded in the tracked object.

RFID tags are used in many industries and can be attached to cash, clothing, and possessions, or implanted in animals and people. However, the possibility of reading personally-linked information without consent has raised privacy and security concerns, leading to standard specifications development addressing privacy and security issues.

An RFID reader is a device used to interrogate an RFID tag. The reader emits radio waves and the tag responds by sending back its data. An RFID tag is a microchip combined with an antenna in a compact package. The tag's antenna picks up signals from an RFID reader or scanner and returns the signal, usually with some additional data.

A passive tag does not contain a battery, but 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, energizing the circuits in the tag and sending the information encoded in the tag's memory.

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

The EM-18 RFID Reader module operating at 125 kHz 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 an output.

3.1.6.1 TAGS:

Fig 3.1.6.1 Tags

A radio-frequency identification system uses tags, or labels attached to objects, to be identified. Two-way radio transmitter-receivers called interrogators or readers send a signal to the tag and read its response.

RFID tags can be passive, active or battery-assisted passive. Passive tags use the radio energy transmitted by the reader, but must be illuminated with a power level a thousand times stronger than for signal transmission.

RFID tags contain an integrated circuit that stores and processes information and modulates and demodulates radio-frequency (RF) signals, a means of collecting DC power from the incident reader signal, and an antenna for receiving and transmitting the signal.

The tag information is stored in a non-volatile memory, and an RFID reader transmits an encoded radio signal to interrogate the tag. The tag receives the message and responds with its identification and other information, such as a unique tag serial number or product-related information.

The RFID system design can discriminate among several tags that might be within the range of the RFID reader and read them simultaneously.

3.1.6.2 READERS:

Fig 3.1.6.2 READERS:

RFID systems can be classified by the type of tag and reader. A Passive Reader Active Tag (PRAT) system has a passive reader which only receives radio signals from active tags (battery operated, transmit only). The reception range of a PRAT system reader can be adjusted from 1–2,000 feet (0–600 m), allowing flexibility in applications such as asset protection and supervision.

An Active Reader Passive Tag (ARPT) system has an active reader, which transmits interrogator signals and also receives authentication replies from passive tags.

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

3.1.6.3 FREQUENCY BANDS:

BAND	REGULATIONS	RANGE	DATA SPEED	REMARKS
120–150 kHz (LF)	Unregulated	10 cm	Low	Animal identification, factory data collection
13.56 MHz (HF)	ISM band worldwide	10 cm–1 m	Low to moderate	Smart cards, ISO-non-compliant memory cards
433 MHz (UHF)	Short range devices	1–100 m	Moderate	Defense applications, with active tags
865–868 MHz (Europe) 902–928 MHz (North America) UHF	ISM band	1–12 m	Moderate to high	EAN, various standards; used by railroads
2450–5800 MHz (microwave)	ISM band	1–2 m	High	802.11 WLAN, Bluetooth standards
3.1–10 GHz (microwave)	Ultra-wide band	Up to 200 m	High	Requires semi-active or active tags

3.1.6.4 SIGNALING:

Signaling between the reader and the tag is done in several different incompatible ways, depending on the frequency band used by the tag. Tags operating on LF and HF bands are, in terms of radio wavelength, very close to the reader antenna because they are only a small percentage of a wavelength away. In this near field region, the tag is closely coupled electrically with the transmitter in the reader. The tag can modulate the field produced by the reader by changing the electrical loading the tag represents. By switching between lower and higher relative loads, the tag produces a change that the reader can detect.

3.1.6.5 USES:

The RFID tag can be affixed to an object and used to track and manage inventory, assets, people, etc. For example, it can be affixed to cars, computer equipment, books, electric vehicles, etc.

RFID can be used in a variety of applications, such as:

- Access management.
- Tracking of goods.
- Tracking of persons and animals.
- Toll collection and contactless payment.
- Machine readable travel documents.
- Smart dust (for massively distributed sensor networks).
- Airport baggage tracking logistics.
- Timing sporting events.
- Tracking and billing processes.

3.1.7 Summary of the Currency Detector and Counter

3.1.7.1 INTRODUCTION

- The system of the design allows voltage differences in the output signal from the receiving coil to be measured as the primary or sole basis for currency identification. Because of the novel geometry of the coils, such voltage differences are indicative of the conductance of the whole currency not just the conductivity of the alloy composition used to fabricate the currency.

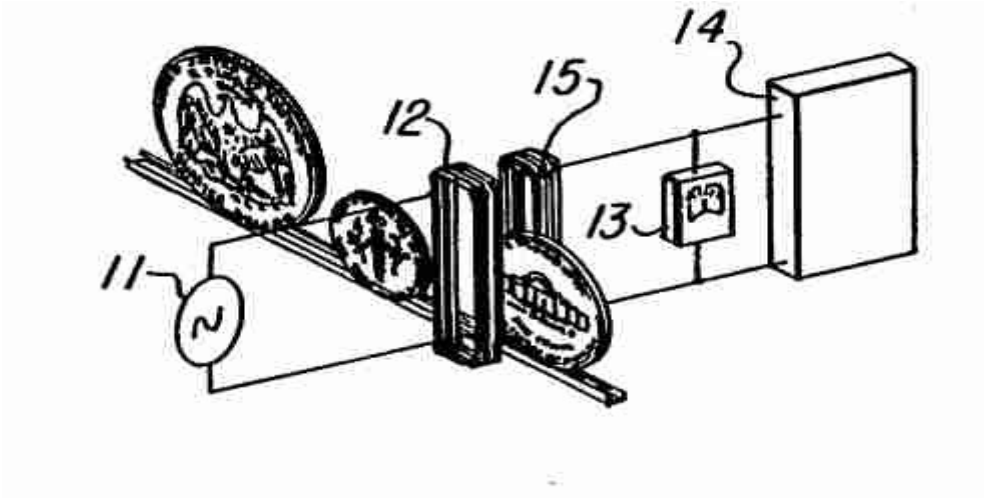


Fig 3.1.7 Currency Detector and Counter

3.1.7.2 Description of the Currency Detector and Counter

- The simplest example of the design consists of an a.c. source 11 connected to transmitter coil 12, and a volt meter 13 and/or an oscilloscope 14 connected to receiver coil 15.
- The currency path between coils 12 and 15 cuts across the width of the coils, i.e., at an angle to the plane of the paper. The distance between the coils must be slightly greater than the thickness of the thickest currency to be counted.
- Preferably the distance between the coils is three to five times greater than the minimum required, so that the chance of jamming the slot is minimized.

- Each coil is wound on an oblong base, which may be either oval-shaped or rectangular. The dimensions of each coil are selected such that the diameter of the smallest currency to be counted, a dime for example, is slightly greater than the width of each coil, and the diameter of the largest currency, a half dollar for example, is slightly less than the length of each coil.
- The coils are coaxial and aligned such that all corresponding sides are parallel to each other. An assortment of currencies to be counted is passed one at a time between the coils. No control of currency speed or position is requiring except that the full diameter of each currency must pass between the coils.

3.1.8 GLOBAL SYSTEM FOR MOBILE COMMUNICATION (GSM)

3.1.8.1 GENERAL DESCRIPTION

A GSM modem or GSM module is a hardware device that uses GSM mobile telephone technology to provide a data link to a remote network. From the view of the mobile phone network, they are essentially identical to an ordinary mobile phone, including the need for a SIM to identify them to the network.

Fig 3.1.8.1 GSM module

3.1.8.2 PRODUCT DESCRIPTION

GSM is a mobile communication modem; it stands for global system for mobile communication (GSM). The idea of GSM was developed at Bell Laboratories in 1970. It is a widely used mobile communication system in the world. GSM is an open and digital cellular technology used for transmitting mobile voice and data services operate at the 850MHz, 900MHz, 1800MHz, and 1900MHz frequency bands. GSM technology was developed as a digital system using the time division 26 multiple access (TDMA) technique for communication purposes. A GSM digitizes and reduces the data, then sends it down through a channel with two different streams of client data, each in its own time slot. The digital system can carry 64 kbps to 120 Mbps of data rates.

3.1.8.3 APPLICATIONS

- They can feature all the functionalities of a mobile phone through computer like making and receiving calls, SMS, MMS etc. These are mainly employed for computer-based SMS and MMS services.
- The GSM/GPRS module demonstrates the use of AT commands. They can feature all the functionalities of a mobile phone through computer like making and receiving calls, SMS, MMS etc. These are mainly employed for computer-based SMS and MMS service.

3.1.9 SIM 900 GSM/GPRS MODULE:

GSM/GPRS Modem-RS232 is built with Dual Band GSM/GPRS engine- SIM900, works on frequencies 900/ 1800 MHz. The Modem is coming with RS232 interface, which allows you connect PC as well as microcontroller with RS232 Chip (MAX232). The baud rate is configurable from 9600-115200 through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable you to connect with internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer application in M2M interface. The onboard Regulated Power supply allows you to connect wide range unregulated power supply. Using this modem, you

can make audio calls, SMS, Read SMS attend the incoming calls and internet through simple AT commands.

3.1.9.1 FEATURES:

- Dual-Band GSM/GPRS 900/ 1800 MHz
- RS232 interface for direct communication with computer or MCU kit.
- Configurable baud rate.
- Power controlled using 29302WU IC.
- ESD Compliance.
- Enable with MIC and Speaker socket.
- With slid in SIM card tray.
- With Stub antenna and SMA connector.
- Input Voltage: 12V DC.

3.1.9.2 PIN SPECIFICATIONS:

PIN	NAME	DETAILS
1	GND	Power supply ground
2	TX	Transmitter
3	RX	Receiver
4	Line_r&Line_l	Line input
5	Spk_p&spk_n	Speaker positive & negative
6	Mic_p&mic_n	Mic positive & negative
7	DTR	Data terminal ready
8	CTS	Clear to send
9	RTS	Request to send

3.1.9.3 WORKING:

A GSM modem does not have a keypad and display, so it accepts certain commands through a serial interface and acknowledges them. These commands are called AT commands and start with "AT". When a ten-digit mobile number is provided, the program instructs the modem to send the text message using a sequence of AT commands.

3.2 BUZZER:

3.2.1 GENERAL DESCRIPTION

Buzzer is a kind of voice device that converts audio model into sound signal. It is mainly used to prompt or alarm. According to different design and application, it can produce music sound, flute sound, buzzer, alarm sound, electric bell and other different sounds.

3.2.2 PRODUCT DESCRIPTION:

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

Fig 3.2.2 Buzzer

3.2.3 FEATURES:

- Color is black.
- The frequency range is 3,300Hz
- Operating Temperature ranges from – 20° C to +60°C
- Operating voltage ranges from 3V to 24V DC
- The sound pressure level is 85dBA or 10cm
- The supply current is below 15mA 3.9.4

3.2.4 APPLICATIONS

- Communication Devices
- Electronics used in Automobiles
- Alarm Circuits
- Portable Devices
- Security Systems

3.3 NODE MCU:

3.3.1 FEATURES:

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa.
- Operating Voltage: 3.3V.
- Input Voltage: 7-12v.
- Digital I/O Pins (DIO): 16.
- Analog Input Pins (ADC): 1.

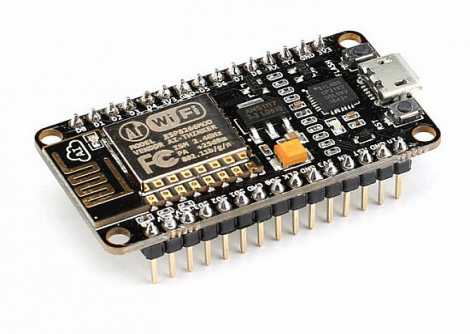


Fig 3.3.1 Node MCU

3.3.2 APPLICATIONS:

- It is used in VR tracker.
- It is used in Security Alarms.

3.4 INTERNET OF THINGS

The Internet of things (IOT) is the network of everyday objects — physical things embedded with electronics, software, sensors, and connectivity enabling data exchange. Basically, a little networked computer is attached to a thing, allowing information exchange to and from that thing. Be it light bulbs, toasters, refrigerators, flower pots, watches, fans, planes, trains, automobiles, or anything else around you, a little networked computer can be combined with it to accept input (especially object control) or to gather and generate informational output (typically object status or other sensory data).

This means computers will be permeating everything around us — ubiquitous embedded computing devices, uniquely identifiable, interconnected across the Internet. Because of low-cost, networkable microcontroller modules, the Internet of things is really starting to take off we are using the ADAFRUIT.IO.

3.4.1 ADAFRUIT.IO

- Adafruit.IO with the MQTT protocol is used on the arduino, because you can use the Adafruit MQTT Arduino library to control it easily.
- This is a general-purpose MQTT library for Arduino that is built to use as few resources as possible so that it can work with platforms like the Arduino Uno.
- The Adafruit.IO Arduino library is a simple library for sending and receiving the latest value for a feed.
- Adafruit.IO also has a web interface as seen below.

Uploading your data to Adafruit.IO is just the beginning. Interacting with that data is where the real magic is. Quickly set up a device that sends you an email if the temperature drops below a certain level. Monitor how much moisture is in your soil and have Adafruit.IO send you an SMS when it needs to be watered. Log the air quality outside and get a notification when it is time to close your windows. This is just the tip of the iceberg. When it comes to all the things you can do with Adafruit.IO, the possibilities really are endless!

Fig 3.4.1 ADAFRUIT

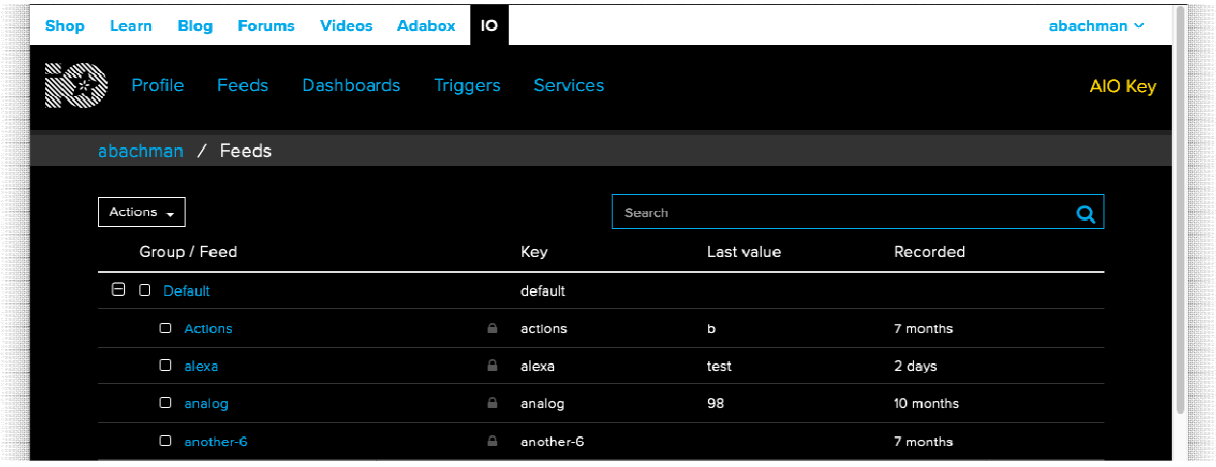


Fig 3.4.2 ADAFRUIT

3.5 SOFTWARE REQUIREMENTS:

3.5.1 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. The device working is based on microcontroller that are programmedby embedded.

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

Here, we need a specific compiler that can help in generating micro-controller based results. Famous compilers used in embedded C are Bi POM Electronic, Green Hill Software and more. Embedded C is generally used to develop microcontroller-based application.

3.5.1.1 EMBEDDED SYSTEM PROGRAMMING:

Basic Declaration

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

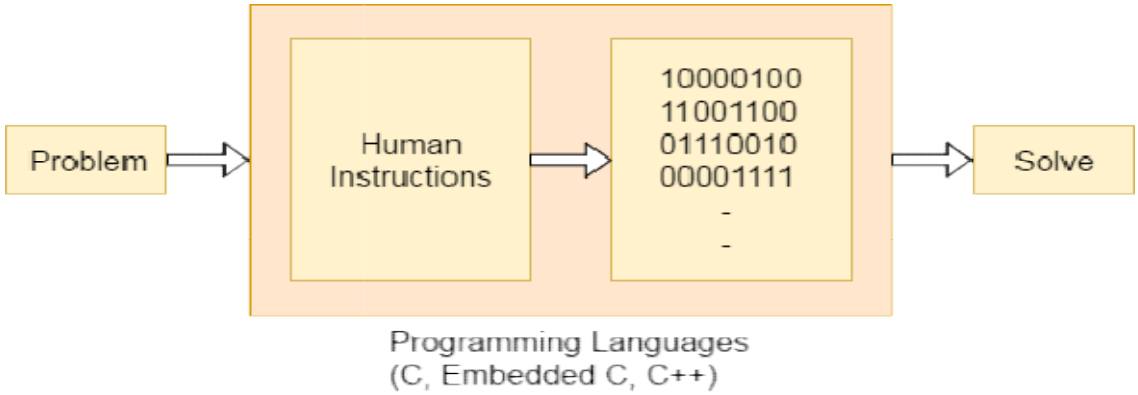


Fig 3.5.1.1 C Programming development

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.

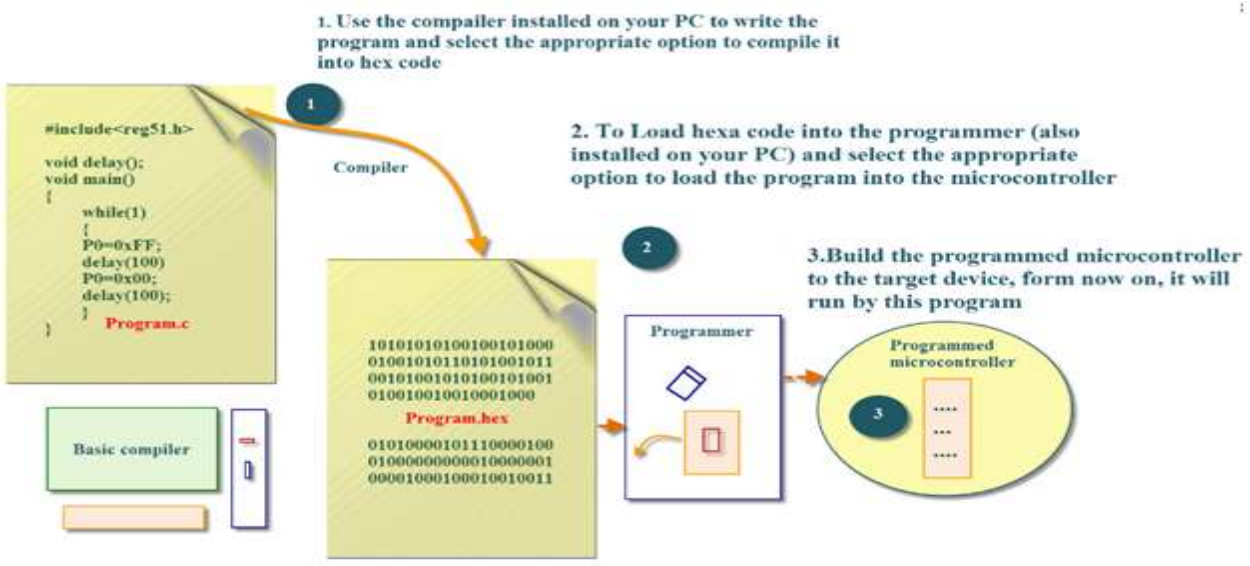


FIG3.5.1.1BLOCKDIAGRAM

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>.

3.5.1.2 BASIC EMBEDDED C PROGRAMMING STEPS:

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

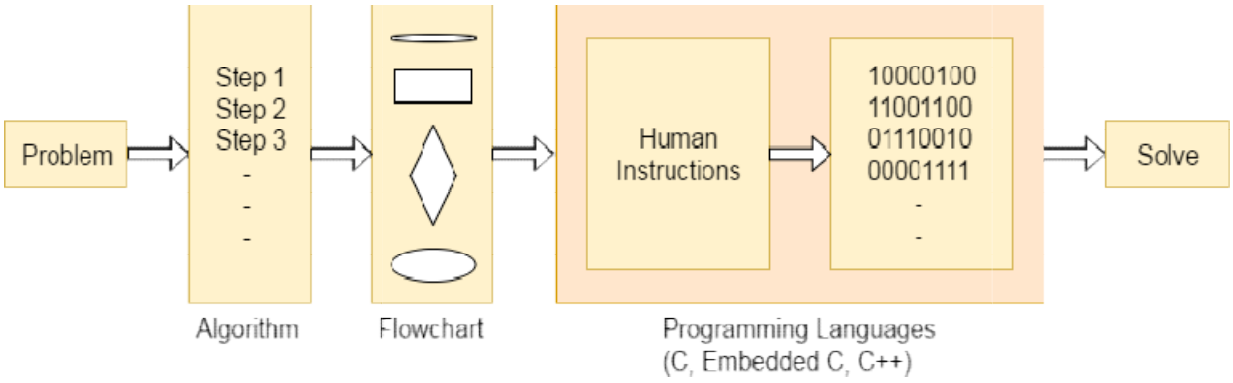


Fig 3.5.1.2 BASIC EMBEDDED C PROGRAMMING STEPS

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.

3.5.2 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.

SYSTEM:

Systems are a way of working, organizing or performing tasks according to a fixed set of rules, program or plan. An Embedded System is a system that has software embedded into computer-hardware, which can be a small independent system or a large combinational system. It is a microcontroller-based control system used to perform a specific task of operation. An embedded system is a combination of three major components:

- **Hardware:** Hardware is physically used component that is physically connected with an embedded system.
- **Application software:** Application software allows the user to perform varieties of application to be run on an embedded system by changing the code installed in an embedded system.
- **Real Time Operating system (RTOS):** RTOS supervises the way an embedded system work. It act as an interface between hardware and application software which supervises the application software and provide

mechanism to let the processor run on the basis of scheduling for controlling the effect of latencies.

Fig 3.5.2 Embedded System

3.5.2.1 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

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.

3.5.3 Designing of an embedded system

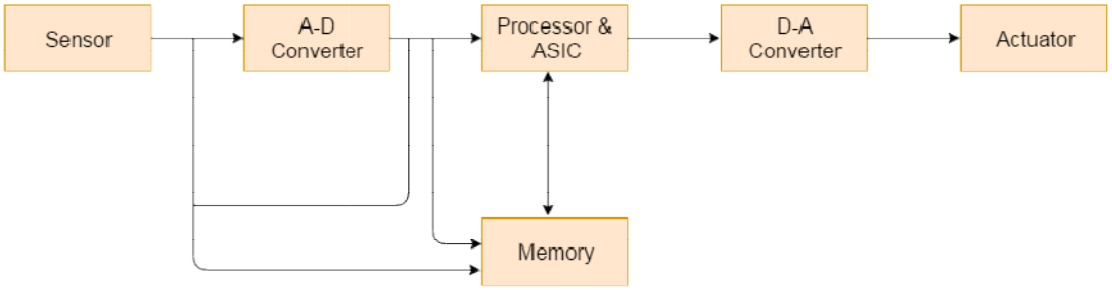


Fig 3.5.3 Designing of an embedded system

BASIC STRUCTURE OF AN EMBEDDED SYSTEM:

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

- **Sensor**: Sensor used for sensing the change in environment condition and it generate the electric 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.
- **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.

3.5.4 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:

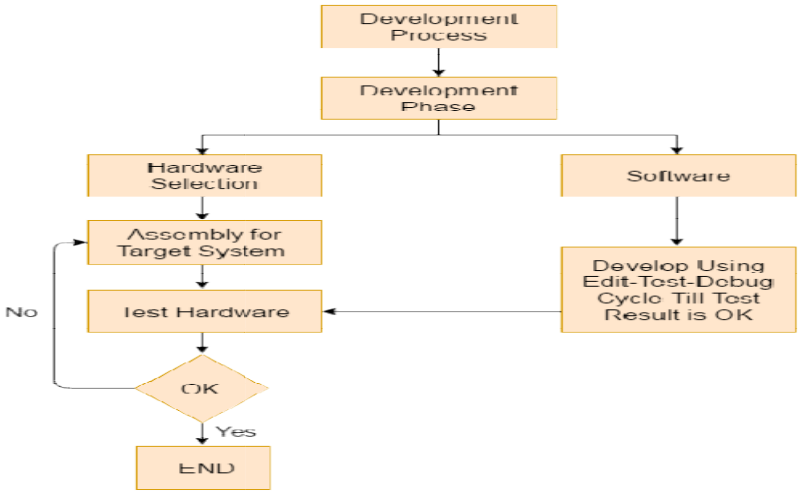


Fig 3.5.4 DEVELOPMENT OF EMBEDDED SYSTEM:

3.5.5 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. The operations performed by compiler are code generation, code optimization, parsing Syntax, etc.

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.

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.

Debugging of a compact subsystem is difficult because a small change in one subsystem can create bugs in another system. The debugging used inside embedded system differs in terms of their development time and debugging features.

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.

Functions of simulators:

Let's see the functions performed by simulator are:

- It defines the processing or processor device family with various version of target system.
- It monitors the detailed information of a source code and symbolic arguments as the execution goes for each single step of operation.
- It simulates the ports of target system for each single step of execution.
- It provides the working status of RAM.
- It monitors the response of system and determines the throughput.
- It provides the complete meaning of the present command.
- It monitors the detailed information of the simulator commands entered from the keyboard or selected from the menu.
- It facilitates synchronization of internal peripherals and delays.

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.

It can also be defined as the ability of a computer program in electronic device to emulate another program or device. It focusing on recreating the original computer environment and helps a user to work on any type of application or operating system.

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.
- 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.

CRITERIA FOR CHOOSING MICROCONTROLLER:

Choosing a microcontroller is essential process in designing of embedded system. While selecting a microcontroller, make sure that it meets the system need and it must be cost effective. We need to decide whether an 8-bit, 16-bit or 32-bit microcontroller is best suitable for the computing needs of a task.

In addition to above, the following points need to be kept in mind while selecting a microcontroller: -

- Speed
- Packaging
- RAM and ROM
- Count of I/O pins
- Cost per unit
- Power consumption

3.6 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 3.6.1 Arduino Software

WRITING SKETCHES:

Sketches written using Arduino Software (IDE) are written in a text editor and saved with the file extension .ino. The editor has features for cutting/pasting and searching/replacing text. The message area gives feedback while saving and exporting, and the console displays text output. 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.

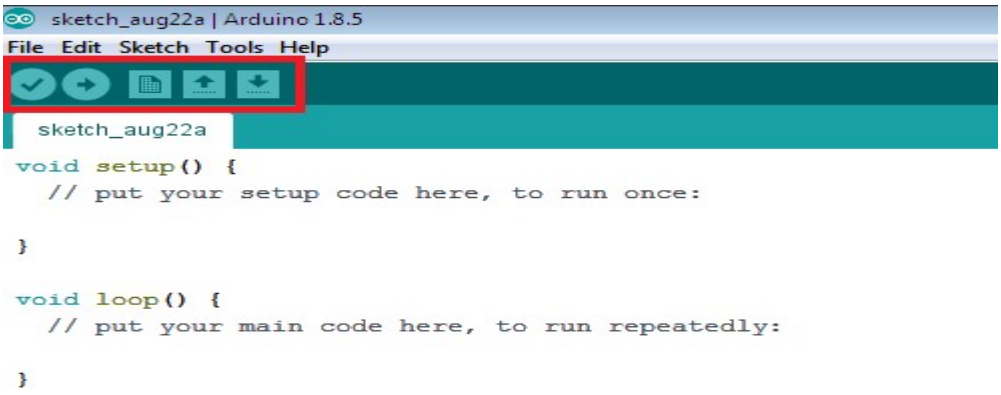


Fig 3.6.2 Arduino Software Sketches

	***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.
	***Save*** Saves your sketch.
	***SerialMonitor*** Opens the serial monitor.

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.

SKETCH BOOK:

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.

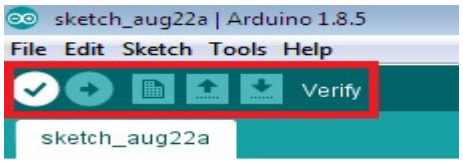


Fig 3.6.3 Ardunio Software Sketch ebook

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:

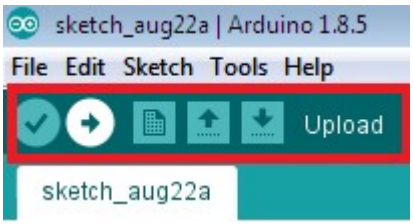


Fig 3.6.4 Arduino Software Upload

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 **boot loader**, 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 boot loader 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.

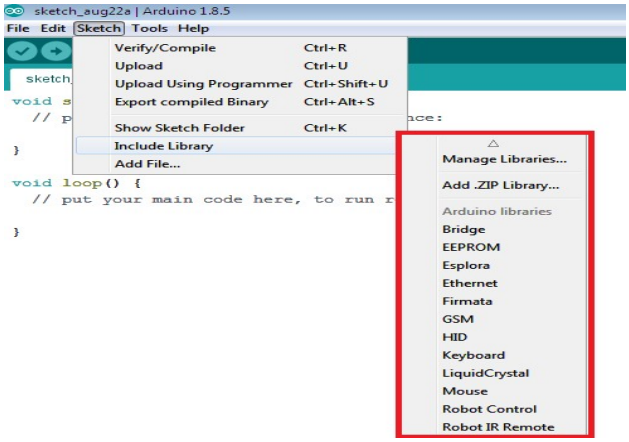


Fig 3.6.5 Ardunio Software LIBRARIES

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, boot loaders, 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:

The Serial Monitor 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. On Windows, Mac or Linux, the board will reset when connected with the serial monitor. If your sketch needs a complete management of the serial communication with control characters, 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

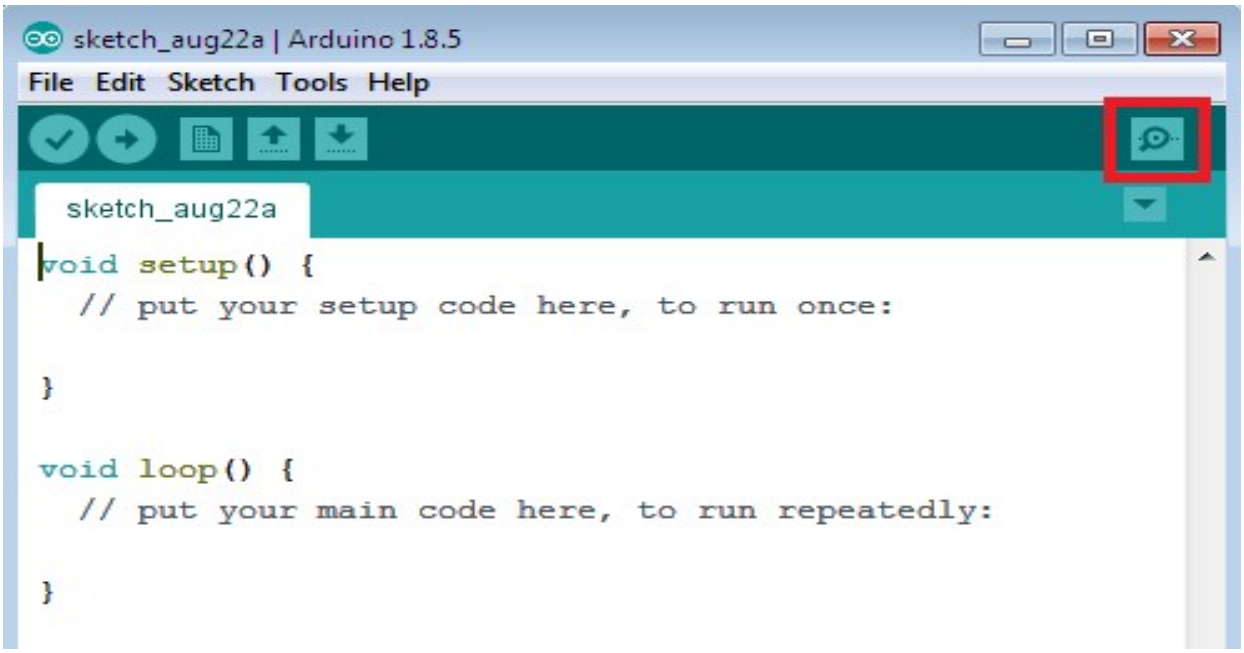


Fig 3.6.6 Arduino Software SERIAL MONITOR

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.

3.7 PYTHON

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and

automatic memory management and supports multiple programming paradigms, including objectoriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems.

CHAPTER: 4 - OUTPUT

4.1 OUTPUT:

WITH CARD TRANSACTION:

By selecting 1 from the keypad, we can choose the card option

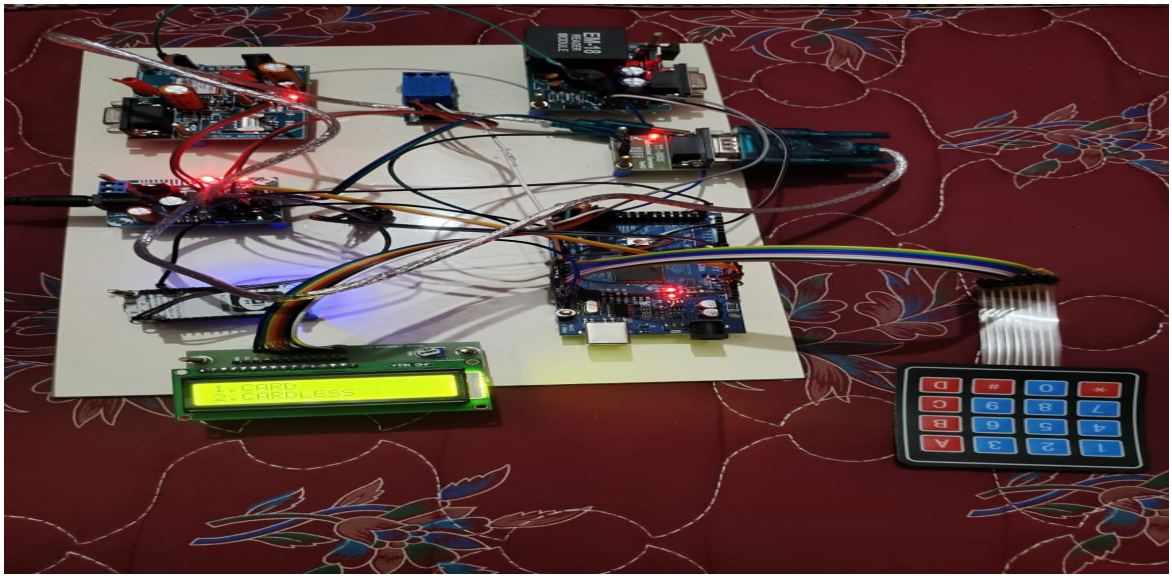
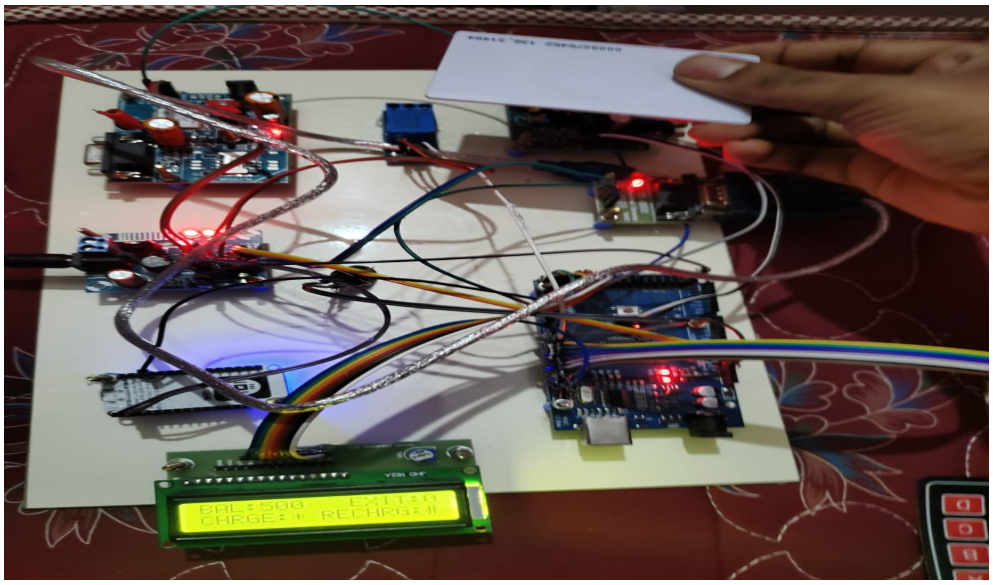


Fig 4.1.1 WITH CARDTRANSACTION

The rfid reader reads the card and shows us the balance of its card details



After reading the card the machine shows us the card balance as below.

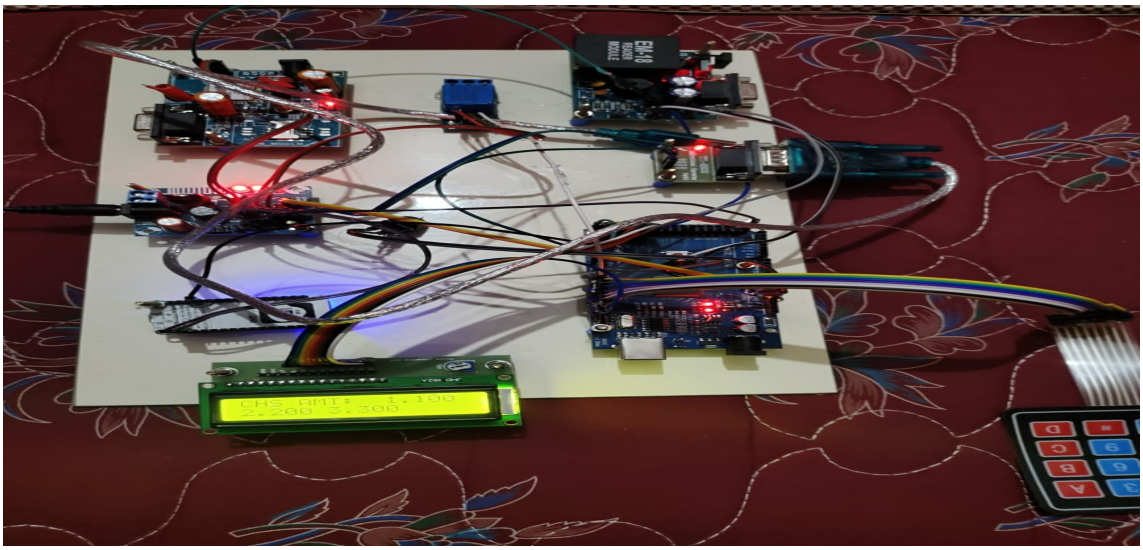


FIG 4.1.2 choosing the amount.

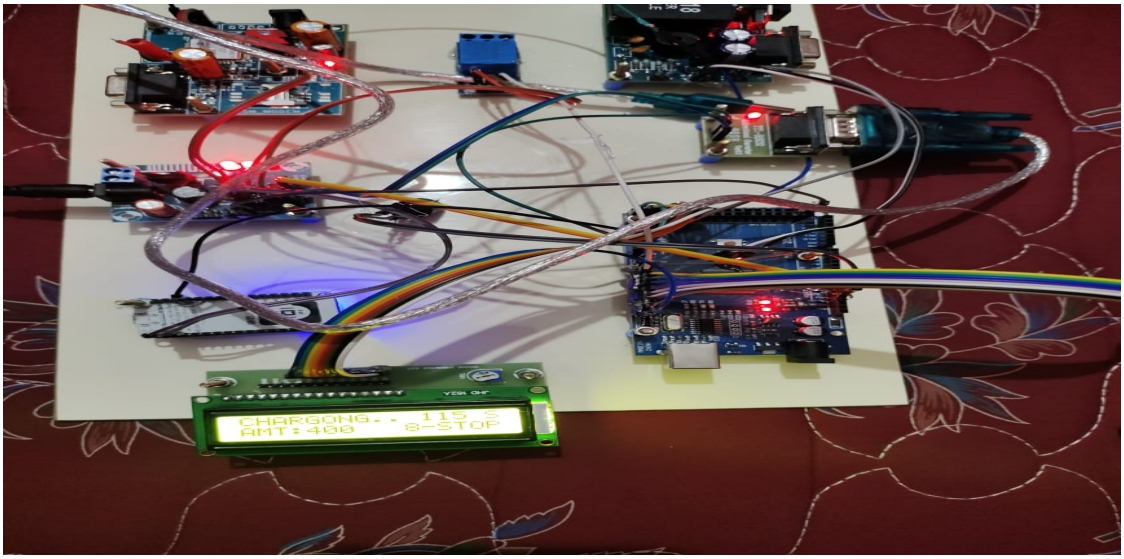
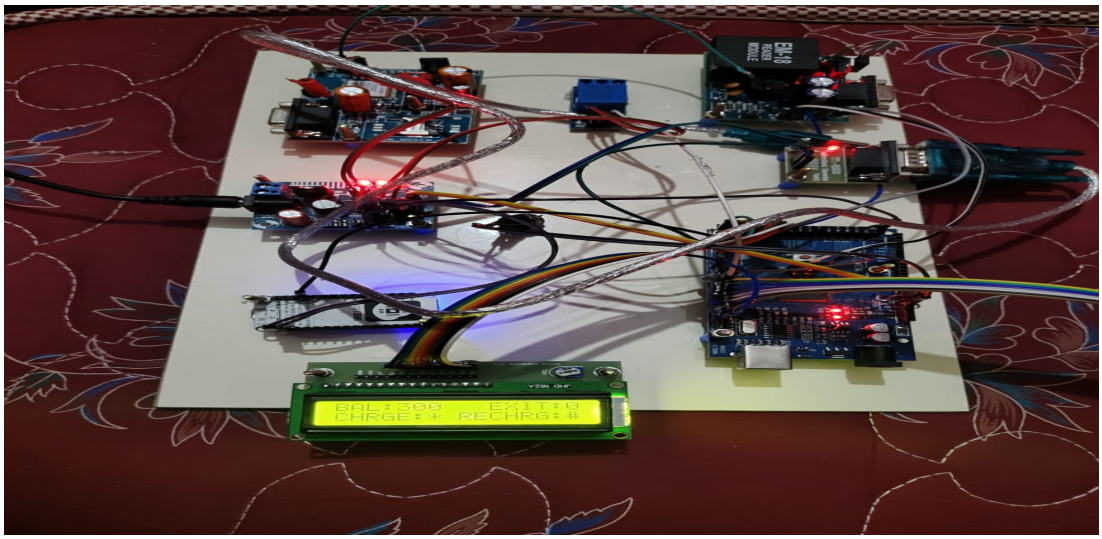


FIG 4.1.3 Charging through the relay.



The remaining balance will be shown after coming of thank you message as shown above.

DASHBOARD

It is another platform where the user can access the below dashboard by entering the login credentials in our website

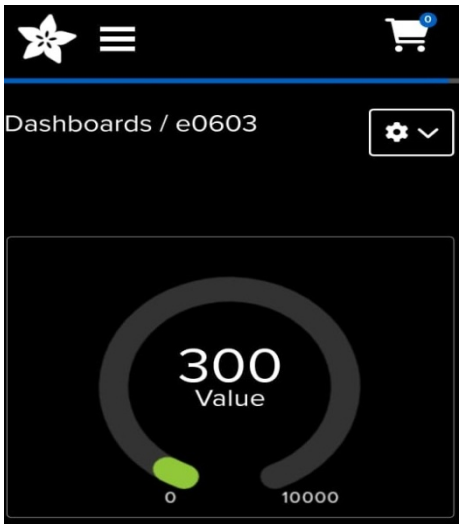


FIG 4.1.4 IOT Dashboard

WITHOUT CARD TRANSACTION:

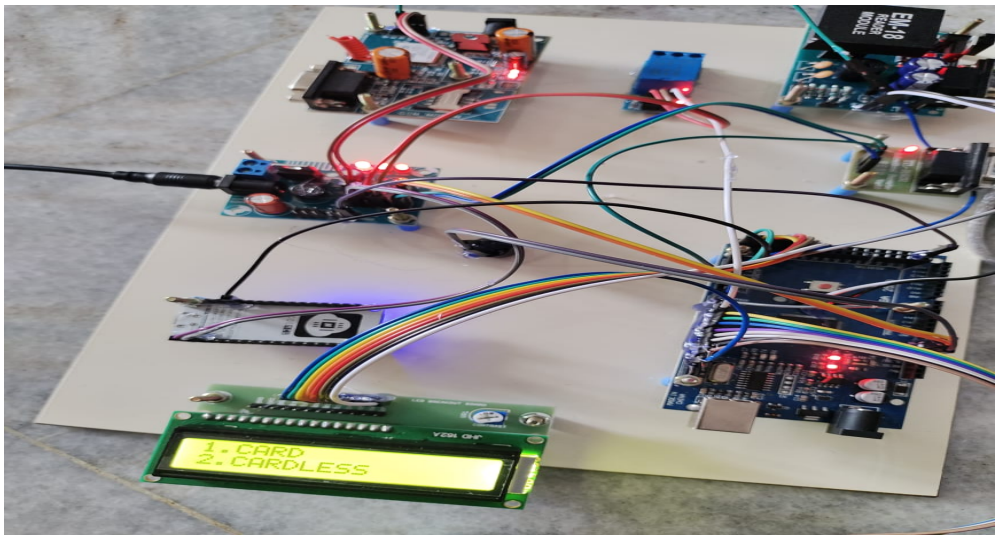
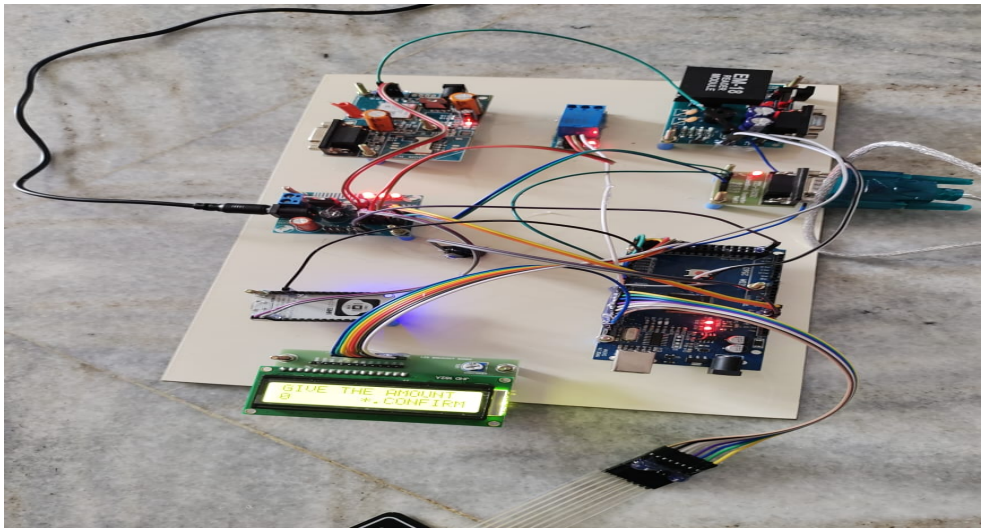


Fig 4.1.5 WITHOUT CARD TRANSACTION.

In this when we select the 2 option we can opt for without card transaction.



Then here we need to enter the amount of money using our deep learning model

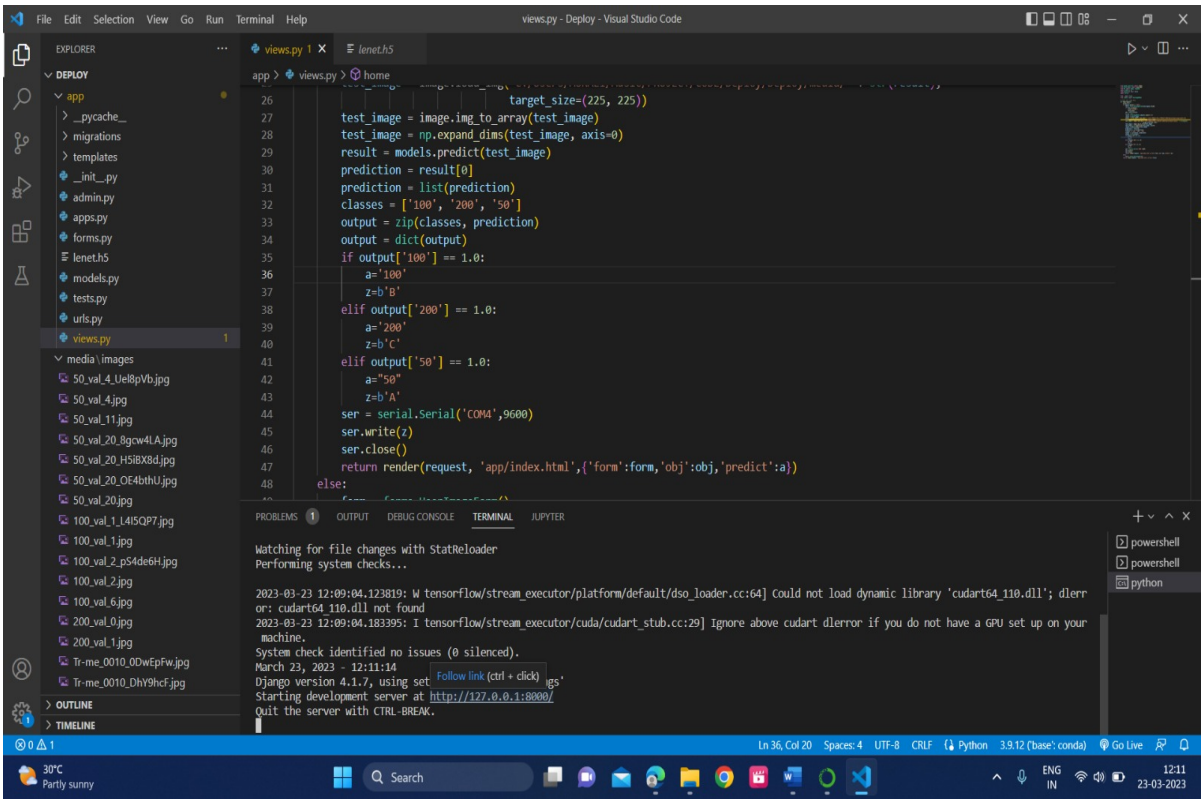


FIG 4.1.6 DEEP LEARNING MODEL.

By clicking the link mentioned as above we are redirected to the below mentioned website.

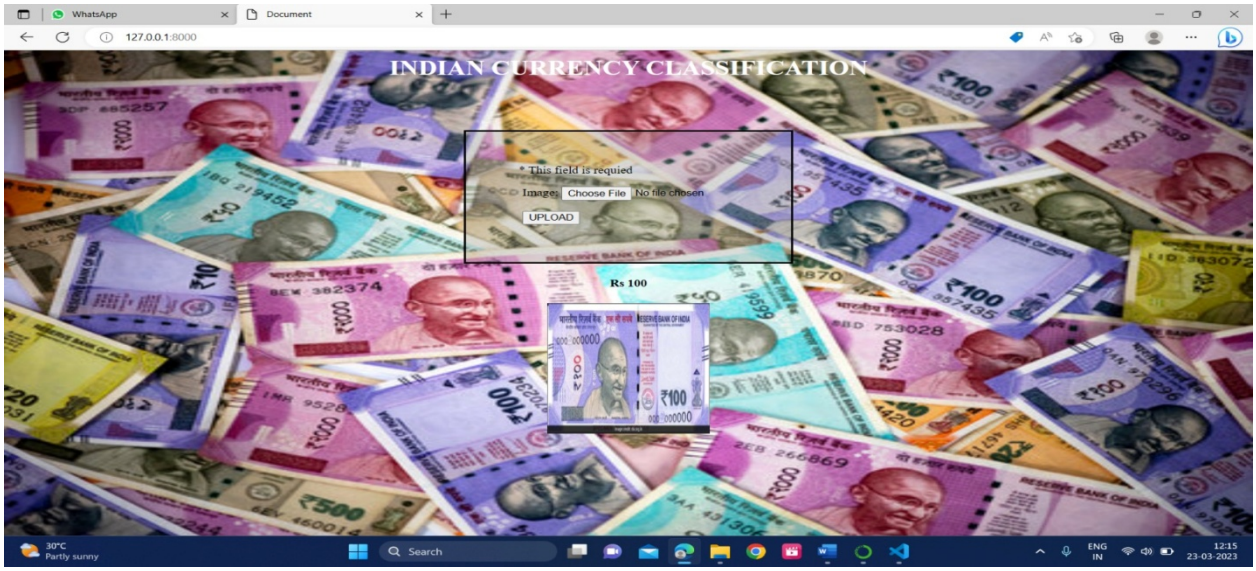


FIG 4.1.7 100 RUPEE NOTE UPLOADING IMAGE

As in the above portal we have uploaded 100 rupees note so the charging starts by using 100 rupees.

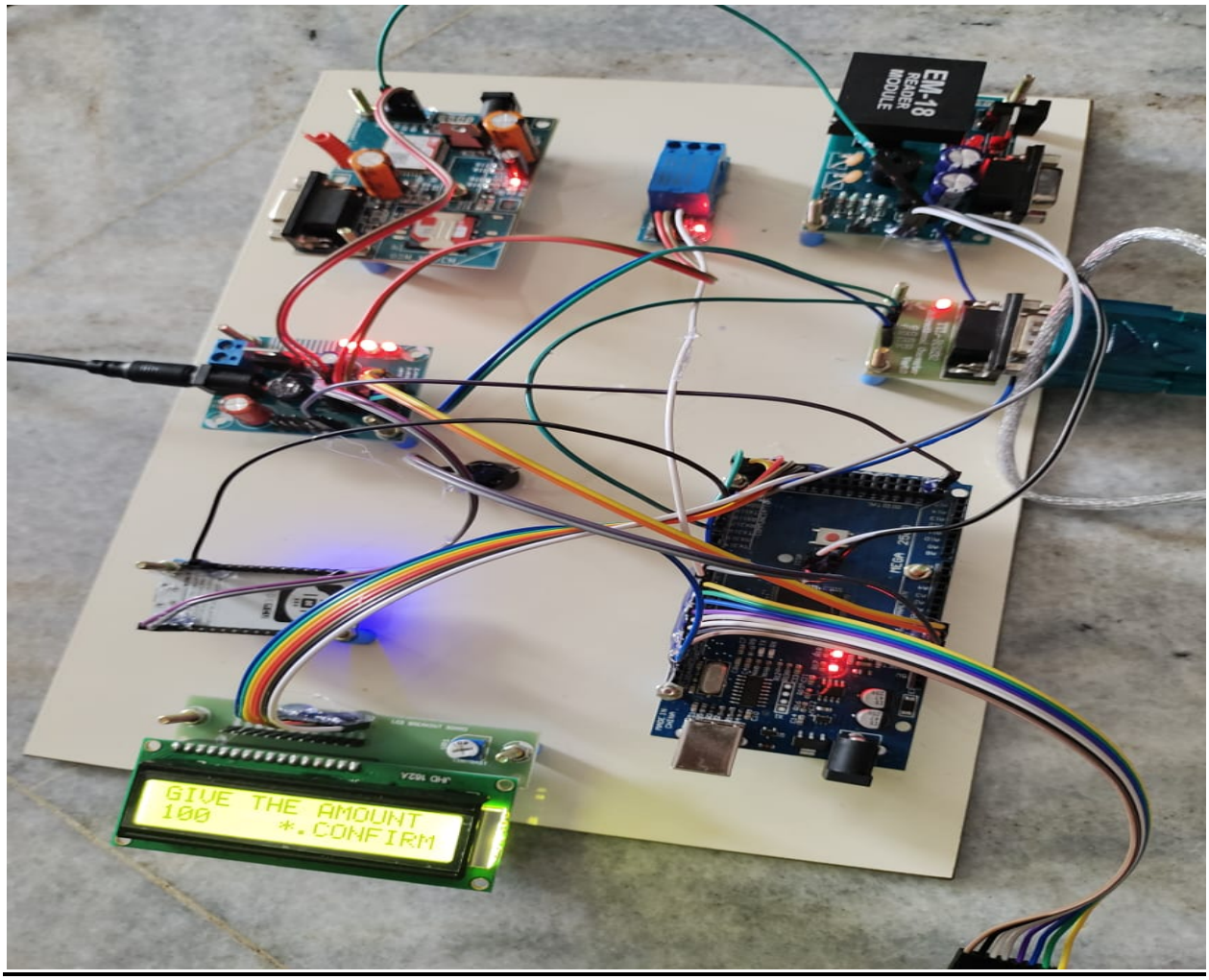


FIG 4.1.8 MONEY UPLOADED IN THE HARDWARE

As an extra feature we have used gsm module to send the message of charging status.

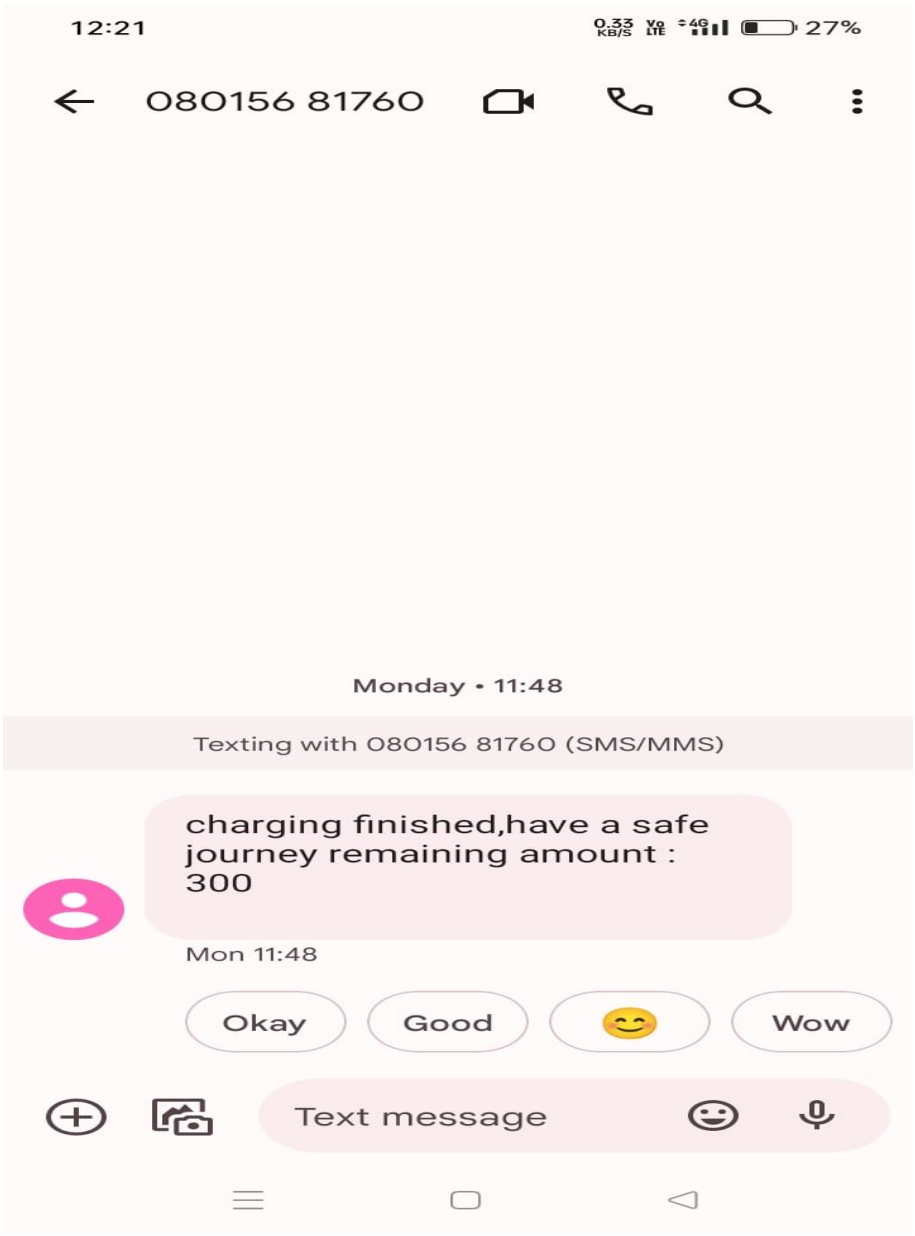


FIG 4.1.9 SMS SENDED FROM THE GSM MODULE.

CHAPTER: 5 -FUTURE ENHANCEMENT

5.1 APPLICATIONS

It is used to responsible for charging infrastructure implementation in smart city.

5.2 FUTURE ENHANCEMENT:

- ✔ No Fuel, Cheaper To Maintain.
- ✔ More Eco-Friendly, Lower Carbon Footprint.
- ✔ Less Noise Pollution, Smoother Ride.
- ✔ Range Anxiety, Lack of Charging Infrastructure.
- ✔ Long Charging Times.
- ✔ Lower Battery Life, High Battery Costs.
- ✔ Fully implemented in the AI concept.

5.3 ADVANTAGES:

- ❖ EV Smart charging provides significant benefits to the EV charging ecosystem as a whole – from the utility to individual EV drivers.
- ❖ Investing in EV charging, especially smart EV charging, brings long-term capital and operating benefits to businesses With EV smart charging,
- ❖ For individual EV drivers, smart EV charging greatest incentive is cheaper, eco-friendly, and safer charging.

5.4 CONCLUSION:

The Share & Charge Foundation develops tools based on distributed technologies, like block chain technology, to support solutions around Electric Vehicle charging. Different to fueling an ICE vehicle, charging an Electric Vehicle increases the amount of transactions between users of vehicles and their provider of energy. Charging will take place at many new locations, be it at home, along the journey or at any destination (e.g., work place). These use-cases create the necessity of a simple and efficient payment method for charging sessions – including the possibility for micro-transactions and instant payments.

5.5 REFERENCES:

[1] Chen W. R., Dai C. H., Li q., Ma L., Wang T. H., and "power management strategy based on adaptive droop control for a fuel cell battery – super capacitor hybrid tramway," IEEE transactions on vehicular technology, vol. 67, no. 7, pp. 5658–5670, jul. 2018.

[2] Chen W. R., Huang W. Q., J. W. Liu, Li Q., Yan y., and " hierarchical management control based on equivalent fitting circle and equivalent energy consumption method for multiple fuel cells hybrid power system" IEEE transactions on industrial electronics, vol. 67, no. 4, pp. 2786–2797, apr. 2020.

[3] Chen W. R., Han Pu, Y., Li q., Su B., Wang Y. C. T. H., Yin L. Z., and "A state machine control based on equivalent consumption minimization for fuel cell/super capacitor hybrid tramway," IEEE transactions on transportation electrification, vol. 5, no. 2, pp. 552–564, jun. 2019.

[4] Chen W. R., Li J. C., Li Q., Liu M., Meng X., Qiu Y. B., Wang t. H., Wang X. T., and "an optimized energy management strategy for fuel cell hybrid power system based on maximum efficiency range identification," journal of power sources, vol. 445, pp. 227333, jan. 2020.

[5] Chen S., Chen L., and Ma D. Z., Sun q. Y., "quasi-z-source network based hybrid power supply system for aluminum electrolysis industry," IEEE transactions on industrial informatics, vol. 13, no. 3, pp. 1141– 1151, jun. 2017.

[6] Deng, Dorrell G D., Hu S., Sun F. C.,Wang Z. P., Zhang X. and "multi objective optimal sizing of hybrid energy storage system for electric vehicles," IEEE transactions on vehicular technology, vol. 67, no. 2, pp. 1027–1035, feb. 2018.

[7] Feng.H, Lukic.S, Srdic .S,Tu .H, and "Extreme fast charging of electric vehicles: a technology overview," IEEE Transactions on Transportation Electrification, vol. 5, no. 4, pp. 861–878, Dec. 2019.

[8] Iravani M. R. and Mahfouz M. M , "Grid-integration of battery-enabled dc fast charging station for electric vehicles," IEEE Transactions on Energy Conversion, vol. 35, no. 1, pp. 375–385, Mar. 2020.

[9] Kezunovic .M, Yan.Q ,Zhang B., and "Optimized operational cost reduction for an EV charging station integrated with battery energy storage and PV generation," IEEE Transactions on Smart Grid, vol. 10, no. 2, pp. 2096–2106, Mar. 2019.

[10] Li Y. W and Nejabatkhah.F, "Overview of power management strategies of hybrid AC/DC micro grid," IEEE Transactions on Power Electronics, vol. 30, no. 12, pp. 7072–7089, Dec. 2015.