

MOBILE CHARGING USING COIN INSERTION

## A MINOR PROJECT – III REPORT

***Submitted by***

|  |  |
| --- | --- |
| **VIDHYA SHREE S** | **927621BEC239** |
| **SIVAPRIYA S** | **927621BEC200** |
| **VINOTHA P** | **927621BEC241** |
| **SHOBIKA S** | **927621BEC198** |

# BACHELOR OF ENGINEERING

in

# DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**M.KUMARASAMY COLLEGE OF ENGINEERING**

(Autonomous)

# KARUR – 639 113

# OCTOBER 2023

**M.KUMARASAMY COLLEGE OF ENGINEERING,**

# BONAFIDE CERTIFICATE

Certified that this **18ECP105L - Minor Project III** report “**MOBILE CHARGING USING COIN INSERTION”**is the bonafide work of “**VIDHYA SHREE S (927621BEC239), SIVAPRIYA S (927621BEC200),VINOTHA P (927621BEC241), SHOBIKA S(927621BEC198)** who carried out the project work under my supervision in the academic year **2021-2025 - ODD**.

|  |  |
| --- | --- |
| **SIGNATURE**  **Dr.A.KAVITHA B.E., M.E., Ph.D.,**  **HEAD OF THE DEPARTMENT,**  Professor,  Department of Electronics and Communication Engineering,  M.Kumarasamy College of Engineering, Thalavapalayam,  Karur-639113. | **SIGNATURE**  **Mr.K.SUDHAKAR.,ME.,**  **SUPERVISOR,**  Assistant Professor,,  Department of Electronics and Communication Engineering,  M.Kumarasamy College of Engineering, Thalavapalayam,  Karur-639113. |

.

This Minor project-III report has been submitted for the **18ECP104L – Minor Project-III** Review held at M. Kumarasamy College of Engineering, Karur on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROJECT COORDINATOR**

# INSTITUTION VISION AND MISSION

**Vision**

To emerge as a leader among the top institutions in the field of technical education.

**Mission**

**M1:** Produce smart technocrats with empirical knowledge who can surmount the global challenges.

**M2:** Create a diverse, fully -engaged, learner -centric campus environment to provide quality education to the students.

**M3:** Maintain mutually beneficial partnerships with our alumni, industry and professional associations

# DEPARTMENT VISION, MISSION, PEO, PO AND PSO

**Vision**

To empower the Electronics and Communication Engineering students with emerging technologies, professionalism, innovative research and social responsibility.

**Mission**

**M1:** Attain the academic excellence through innovative teaching learning process, research areas & laboratories and Consultancy projects.

**M2:** Inculcate the students in problem solving and lifelong learning ability.

**M3:** Provide entrepreneurial skills and leadership qualities.

**M4:** Render the technical knowledge and skills of faculty members.

**Program Educational Objectives**

**PEO1:Core Competence:** Graduates will have a successful career in academia or industry associated with Electronics and Communication Engineering

**PEO2:Professionalism:** Graduates will provide feasible solutions for the challenging problems through comprehensive research and innovation in the allied areas of Electronics and Communication Engineering.

**PEO3:Lifelong Learning:** Graduates will contribute to the social needs through lifelong learning, practicing professional ethics and leadership quality

**Program Outcomes**

**PO 1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO 2: Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO 3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO 4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO 5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO 6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO 7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO 8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO 9: Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO 10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO 11: Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO 12: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**Program Specific Outcomes**

**PSO1:** Applying knowledge in various areas, like Electronics, Communications, Signal processing, VLSI, Embedded systems etc., in the design and implementation of Engineering application.

**PSO2:** Able to solve complex problems in Electronics and Communication Engineering with analytical and managerial skills either independently or in team using latest hardware and software tools to fulfill the industrial expectations.

|  |  |
| --- | --- |
| **Keywords** | **Matching with POs, PSOs** |
| Coin Sensor, Mobile, Coin acceptor, LCD display | PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12, PSO1,  PSO2 |

# ACKNOWLEDGEMENT

Our sincere thanks to **Thiru.M.Kumarasamy, Chairman** and **Dr.K.Ramakrishnan, Secretary** of **M.Kumarasamy College of Engineering** for providing extraordinary infrastructure, which helped us to complete this project in time.

It is a great privilege for us to express our gratitude to **Dr.B.S.Murugan., B.Tech., M.Tech., Ph.D., Principal** for providing us right ambiance to carry out this project work.

We would like to thank **Dr.A.KAVITHA B.E., M.E., Ph.D., Professor and Head**, **Department of Electronics and Communication Engineering** for his unwavering moral support and constant encouragement towards the completion of this project work.

We offer our wholehearted thanks to our **Project Supervisor, Mr.K.Sudhakar,M.E**,**Assistant Professor,**Department of Electronics and Communication Engineering for his precious guidance, tremendous supervision, kind cooperation, valuable suggestions and support rendered in making our project to be successful.

We would like to thank our **Minor Project Co-ordinator**,**Dr.K.Karthikeyan, B.E., M.Tech., Ph.D., Associate Professor**, Department of Electronics and Communication Engineering for his kind cooperation and culminating in the successful completion of this project work. We are glad to thank all the Faculty Members of the Department of Electronics and Communication Engineering for extending a warm helping hand and valuable suggestions throughout the project. Words are boundless to thank our Parents and Friends for their motivation to complete this project successfully.

# ABSTRACT

Now a days Mobiles are play’s important role in the present communication world as well as day to day life. This paper describes the mobile battery charger on coin insertion. The mobile phone business is currently worth billions of dollars and supports of most number of features in every mobile phone with different operating systems. So to operate these mobile phones public charging is needed, and it should be useful to public. This is designed based on ATMEL 89c51 a 40-pin micro controller that does the countdown timings for a period of 5 minutes with LCD displays showing the actual time left. During the timing period a relay output is latched and finishing timing in progress. Recommended locations include: Hotels, Conference centers, Exhibition halls, Serviced offices, Exchange halls, Hotels, Health clubs, Training centers, Golf clubs, Retail outlets, Shopping malls, Internet cafes, Universities, Colleges, Hall of residence, Airports, Train terminals, etc., so that the mobile phone users can reactivate a low or dead battery by simply plugging in and charging for one rupee.

## TABLE OF CONTENTS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CHAPTER**  **No.** |  | **CONTENTS** | | **PAGE**  **No.** |
|  | **Institution Vision and Mission** | | | iii |
|  | **Department Vision and Mission** | | | iii |
|  | **Department PEOs, POs and PSOs** | | | iv |
|  | **Abstract** | |  | viii |
|  | **List of Tables** | | | xi |
|  | **List of Figures** | | | xii |
|  | **List of Abbreviations** | | | xiii |
| **1** | **INTRODUCTION** | | | **1** |
|  | 1.1 | Project Details | | 1 |
|  | 1.2 | Description | | 2 |
|  |  | 1.2.1 | Components of Mobile charging using coin insertion | 2 |
|  |  | 1.2.2 | Observation Networks | 2 |
| **2** | **LITERATURE SURVEY** | | | **3** |
|  | 2.1 | Prominent measure of the gathered data | | 3 |
|  | 2.2 | Cantilever type of sensors in the coin based insertion | | 4 |
| **3** | **EXISTING SYSTEM`** | | | **6** |
|  | 3.1 | Mobile phone charger | | 6 |
|  | 3.2 | The driver circuit | | 6 |
| **4** | **PROPOSED SYSTEM** | | | 7 |
| **5** | **WORKING PRINCIPLE** | | | 8 |
| **6** | **HARDWARE AND SOFTWARE REQUIREMENTS** | | | 15 |

|  |  |  |
| --- | --- | --- |
| **7** | **RESULT AND DISCUSSION** | 41 |
| **8** | **CONCLUSION AND FUTUTE WORK** | 42 |
|  | **APPENDICES** | 43 |
|  | **REFERENCES** | 44 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE No.** | **TITLE** | **PAGE No.** |
| 6.1 | Applications of Infrared Sensors | 23 |

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE No.** | **TITLE** | **PAGE No.** |
| 5.1 | Block Diagram | 8 |
| 5.2 | Mega 2560 | 9 |
| 5.3 | HD44780 Pin diagram | 11 |
| 5.4 | Power Supply | 13 |
| 5.5 | Block Diagram of RPS | 14 |
| 6.1 | Arduino | 15 |
| 6.2 | Arduino UNO pinout | 18 |
| 6.3 | Infrared Sensor | 21 |
| 6.4 | Infrared LED | 22 |
| 6.5 | Atmega 328 | 27 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **IR** | Infrared |
| **LDR** | Light Dependent Resistors |
| **LCD** | Liquid Crystal Display |
| **EEPROM** | Electrically Erasable Programmable Read-only Memory |
| **USB** | Universal Serial Bus |
| **PSU** | Power Supply Unit |

# CHAPTER 1

## INTRODUCTION

* 1. **PROJECT DETAILS**

This is the smart coin based Mobile charging system that charges the vehicle for the particular amount of time. This system is used by shop owners, public places like railway stations to provide charging facilities. So the system consists of a coin recognition module that recognizes the valid coin is found it signals the micro controller for further action. If a valid coin is found, it signals the microcontroller and it starts the charging mechanism providing a supply through the power supply section to the vehicle, now system also needs to monitor the amount of charging to be provided. So the system can be used for mobile charging at public places.

## DESCRIPTION

The objective of this project is inserting a coin using charge for your mobile in public places. This project is very useful to people who are all using mobiles without charging condition in public place. In this project, who are all using e-vehicles in outside of home or office without charging condition. The coin based e-vehicle is very useful to that person for using coin to charge for that e-vehicle. A sensor system is used to detect the presence of coin. It may be of different type (IR sensor,using LDR, etc…)The coin is inserted between the transmitted and received signal.

## COMPONENTS OF INDIAN TSUNAMI WARNING SYSTEM

The Mobile charging using coin insertion includes IR sensor, LDR, System module, Microcontroller-AT89S52, LCD, Transistor ,Diode, Resistor, Relay

## OBSERVATION NETWORKS

When a signal came from sensor unit, the microcontroller activates the charger unit for a predefined time. After that it will reset to normal case. Driver circuit is used for provide the sufficient input voltage of relay. The relay will onto activate the 203v charger, we will use charger to charge for your e-vehicles. The major action in this system is controlled by transmitter section: this section consists of IR transmitter and IR receiver. Here we need to generate IR frequency continuously. So that by using a small tiny microcontroller frequency is produced and is connected IR receiver continuously receives the signal from the transmitter. Whenever the light path in between IR transmitter and IR receiver cuts by an obstacle receiver signal gives low to high pulse. By to the IR lead to generate IR light rays of 38KHz frequency.

# CHAPTER 2

## LITERATURE SURVEY

* 1. **Prominent measure of the gathered data**

A more prominent measure of the gathered data have been genuinely orchestrated, so the proportion of data to survey our procedure has been in like manner extended. Another kind of charger proposed for open people utilities. This sort charger will be useful for the open people; regularly the battery ends up being level in conversation in particularly at gravely masterminded events when access to a standard is past the domain of creative mind. Consequently sun based base adaptable charger is continuously important. The power deftly for the charger is settled from sun situated power and current smoothly. As demonstrated by S. B. SHRIDEVI, delineate coin base flexible charger using daylight based after structure. In this investigation, the structure is plan for open in nation similarly as semi urban areas. This is organized base on microcontroller that does the initiation time for a period of 3 min with LCD show exhibiting the continuous left. During the timespan an exchange out is snare and finishing time in progress. According to S. BHANU. PRATAP, Coin based convenient charger is useful in many making countries where the present deftly isn't open for a couple of hours on steady timetable. In correspondence domain flexible is fundamental for correspondence. They organized sun controlled adaptable charger for charging various makers mobiles. It is used to help the people where the power nimbly isn't open for a long time step by step.The coin-based versatile battery charger is created for offering a novel help to the natural open where lattice power isn't available for mostly/full daytime and a wellspring of salary for site providers.

## Cantilever type sensors

In 2017, Dhara G. Rangani, Nikunj V. Tahilramani [1] have presented mobile charging using coin in which their main focus was coin detection for which they have used cantilever type sensor for coin detection. Cantilever type sensor detects weight of 5 rupee coin and gives digital signal to ADC. Using this controller check whether coin is original or duplicate. They have also used solar power for charging the mobile station battery and used greed power when solar power is not available.

In May 2017, Mr. C V Raja Reddy, Uzoigwe Daniel, Rupesh Rai, Balaji R [2] have proposed coin based mobile charging with solar tracking in which their main focus was solar tracking for which they have used LDRs so according to the sunlight intensity LDR resistance will be varied. When the sun intensity is high then LDR offers less resistance, the voltage across each LDR is given to the ADC, then controller checks in accordance with the algorithm designed and rotates the motor in specified direction.

In 2015, Nupur Khera [3] have presented a solution to improve the charging and discharging control of battery. The solar charge controller will prevent the overcharging of the battery thus it will help to increase the life of battery. Solar charge controller will also prevent reverse flow of current from the batteries to the solar panels at night.

In 2013, S. B. Sridevi [4] have provided a solution to solar tracking. As sun rises in the east and sets in the west, So if the system cannot change the direction towards sun then it will not be able to consume maximum sunlight this makes the system inefficient so to overcome the problem solar tracking system has been developed.

One of the most notable manageable force sources is sun based essentialness. Sun fueled after enables greater imperativeness to be made in light of the fact that the sun based barricade can hold a contrary profile to the sun's shafts. This structure develops a previous senior arrangement adventure where understudies amassed a sun situated controlled battery charger, therefore making this system ideally free. According to T.GUNAWAN perfect circumstance for ingestion inverse to the sun controlled radiation during daylight hours, can extend the accumulated essentialness by up to 50%, Commercially; single-rotate and two center point following segments are available. For the most part, the single center tracker follows the Sun's East-West turn of events, while the two-rotate tracker follows furthermore the Sun's changing height point. An after system must have the choice to follow the sun with a particular degree of precision, return the position to its interesting circumstance continuously end and moreover track during times of spread over. They probably inspected the working of Coin based telephone charger with sun fueled after system by top power arranging.

# CHAPTER 3

## EXISTING SYSTEM

## 3.1 Mobile phone charger

## In previous models, the mobiles were charged using various types like solar panel, coin-based charging. This coin based mobile charger is used to charge the mobile phones in case of emergency provided that they don’t have power bank. So this project model can be implemented in streets or they could be fixed to the post. The charging time for the phone is displayed on the LCD. In addition, a sun panel may be used to fee the device. Solar energy is used to charge cellphones in the majority of cases since it converts light energy into DC current. Insertion of a fixed coin size for charging is not developed during the existing system of the process.

## 3.2 The driver circuit

The driver circuit ensures that the relay's input voltage is sufficient.  The relay will activate, allowing us to utilize the 230v charger to charge our phones. The ESP8622 WI-FI module was used to link Arduino to the internet. Because of its simple serial communication connectivity, the ESP8622 WI-FI module can be used as a server, WI-FI adapter, and wireless internet access interface for any microcontroller. Our ESP8266 will function as an Access Point (AP Mode), allowing other devices (stations) to connect to the Wi-Fi network over a wired network. The ESP8622 WI-main FI's purpose is to update the number of entries in the system's currency status to the cloud, which only requires one communication channel. The ESP8622 WI-FI module is used to establish a connection as well as to enable IoT services (cloud computation).

# CHAPTER 4

## PROPOSED SYSTEM

The existing system on the process would have been elaborated by the system with the help of the IR sensor it checks the coin is valid or not. Here we are checking the real coins. If it is valid sends particular signal to the Arduino where it can be processed as a fact that it can be provided to the LCD display for the time duration for which it is done and for which it is provided a display for the process.

And also it contains the WI-FI module which gives number of entries of a coin status to the system of cloud and it is based on the IOT service. Date time will be send on server side through mobile hotspot. It also have multiple pin to charge different mobiles. The system uses a coin acceptor machine which will detect for a valid coin. The coin acceptor is connected to Arduino board and it then connected to the LCD display. If a valid coin found it signals the Arduino and the Arduino will send signal to LDC to display the amount of time to be charged. The relay will control the power supply to charger, after completion of time duration it will stop supplying power. The SMPS (Switch Mode Power Supply) in the charger used for managing energy conversion. The LCD screen will show how much time left for the charging if the user wants to increase the duration of charging he needs to add another coin then the microcontroller adds the time in currently remaining time.

# CHAPTER 5

## WORKING PRINCIPLE

**BLOCK DIAGRAM**

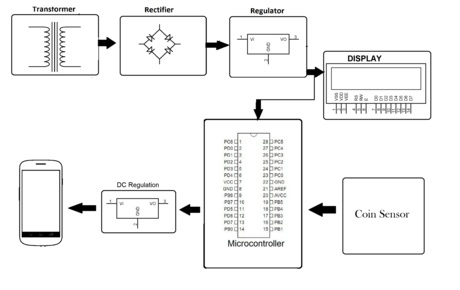


Fig 5.1 Block Diagram

`

In this project, we have used Arduino MEGA microcontroller which acts as brain of our system, hence the entire system program is stored in it. Here the use of the RFID and coin accepter module we can give the two option for the payment purpose. The RFID is used to make a payment using card and the coin 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 coin. GSM module is used to update payment information for the user. Buzzer give the alert for the start and end of the charging time. Whenever the person completed the payment relay will on and charging the vehicle. The LCD is used to update all the information of the device.

## Physical characteristics and field 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.

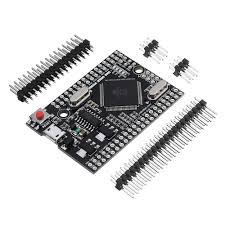


Fig 5.2 Mega 2560

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 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent 47 locations. Furthermore, the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega 2560 and Duemilanove / Diecimila boards. Please note that I2C is not located on the same pins on the Mega 2560 board (20 and 21) as the Duemilanove / Diecimila boards (analog inputs 4 and 5).

Rather then requiring a physical press of the reset button before an upload, the Mega 2560 is designed in a way tha t allows it to be reset by software running on a connected computer. One of the hardwa re flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Mega 2560 board is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the ATMega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

If a sketch running on the board receives one-time configuration or other data when 48 it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega 2560 board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

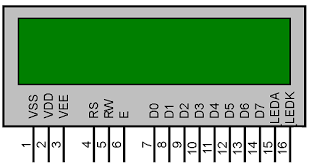


Fig 5.3 HD44780 Pin diagram

## BF - Busy Flag:

## Usually these days you will find single controller LCD modules are used more in the market. So in the tutorial we will discuss more about the single controller LCD, the operation and everything else is same for the double controller too. Lets take a look at the basic information which is there in every LCD. Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing. To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

## Instruction Register (IR) and Data Register (DR):

## There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc…

## When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD. We will discuss more on LCD instruction set further in this tutorial.

## Commands and Instruction set:

## Only the instruction register (IR) and the data register (DR) of the LCD can be controlled by the MCU. Before starting the internal operation of the LCD, control information is temporarily stored into these registers to allow interfacing with various MCUs, which operate at different speeds, or various peripheral control devices. The internal operation of the LCD is determined by signals sent from the MCU. These signals, which include register selection signal (RS), read/write signal (R/W), and the data bus (DB0 to DB7), make up the LCD instructions.

## There are four categories of instructions that:

## • Designate LCD functions, such as display format, data length, etc.

## • Set internal RAM addresses

## • Perform data transfer with internal RAM

## • Perform miscellaneous functions.

## Although looking at the table you can make your own commands and test them.

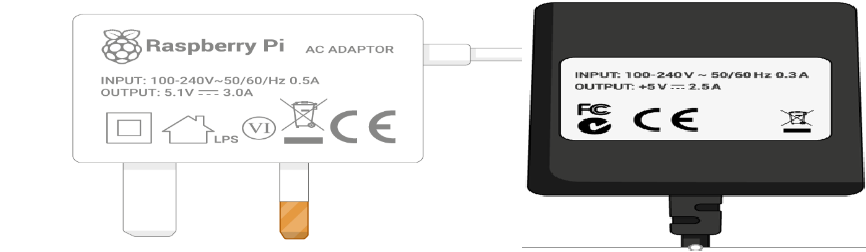
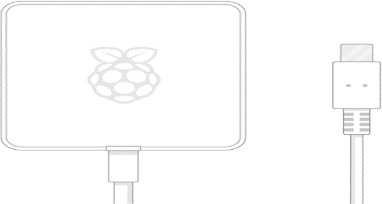


Fig 5.4 Power Supply You need a power supply that provides:

* At least 3.0 amps for raspberry pi 4
* At least 2.5 amps for raspberry pi 3

## Linear Power supply

An AC powered linear power supply usually uses a transformer to convert the voltage from the wall outlet (mains) to a different, usually a lower voltage. If it is used to produce DC, a rectifier is used. A capacitor is used to smooth the pulsating current from the rectifier. Some small periodic deviations from smooth direct current will remain, which is known as ripple. These pulsations occur at a frequency related to the AC power frequency (for example, a multiple of 50 or 60 Hz).

The voltage produced by an unregulated power supply will vary depending on the load and on variations in the AC supply voltage. For critical electronics applications a linear regulator will be used to stabilize and adjust the voltage. This regulator will also greatly reduce the ripple and noise in the output direct current. Linear regulators often provide current limiting, protecting the power supply and attached circuit from over current.

Adjustable linear power supplies are common laboratory and service shop test equipment, allowing the output voltage to be set over a wide range.

For example, a bench power supply used by circuit designers may be adjustable up to 30 volts and up to 5 amperes output. Some can be driven by an external signal, for example, for applications requiring a pulsed output.

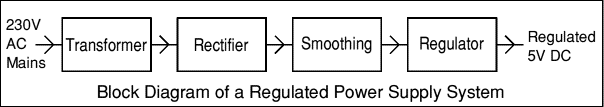


Fig 5.5 Block Diagram of a RPS

# CHAPTER 6

## HARDWARE AND SOFTWARE REQUIREMENTS

**HARDWARE REQUIREMENTS**

* + Arduino UNO
  + IR Sensor
  + Relay
  + Atmega 328
  + LCD Display
  + Power Supply

## ARDUINO

I was surprised to see a twelve-year-old boy giving life to his electronic gadgets. He was trying his hands on building his own creative toys which involved hard electronics and software skills. My zeal was on its peak to know the magical power inside the young chap. How did he understand the concepts of electronics so early? How did he develop the software? Anxiously I went down and asked him about the magic he was doing. The answer was “ARDUINO”.



Fig 6.1 Arduino

Arduino is an open source electronics platform accompanied with a hardware and software to design, develop and test complex electronics prototypes and products. The hardware consists of a microcontroller with other electronic components which can be programmed using the software to do almost any task. The simplicity of the Arduino language makes it very easy for almost everyone who has an interest in electronics to write programs without the understanding of complex algorithms or codes. Arduino is intended for an artist, tinker, designer or anyone, interested in playing with electronics without the knowhow of complex electronics and programming skills. Arduino is an excellent designed open source platform. It has specially designed boards which can be programmed using the Ardunio Programming Language (APL). The presence of Arduino is not only spreading between hobbyists, but it has also expanded its roots in industries and used by experts for making prototypes of commercial products. Arduino takes off the efforts required in complex coding and designing hardware. The open source nature of Arduino has been the main reason for its rapid horizontal growth. Since it is an Open Source project, all the files related to hardware and software is available for personal or commercial use. The development cost of the hardware is very small as against the costly similar proprietary products by the industrial giants. The open source nature doesn’t require any licenses to develop, use, redistribute or even sell the product. But the Arduino name is trade mark protected (Arduino™) i.e., you are free to sell the Arduino board under any other name however in order to sell it under the name “Arduino” you need to take permission from the founders and follow their quality terms. The Software files which includes all the source code library are also open sourced. A user can modify them to make the project more versatile and improve its capabilities. This provides a strong online community support.

## Concept of Arduino

The root of Arduino goes deep down to the development of Processing Language by MIT researchers. Processing language is an open source language designed to introduce the software development environment for the artistic people without the need of deep knowledge of programming of algorithms. Processing is based on java.

In year 2003 Hernando Barragan, a programmer developed an open source electronics development platform with software IDE, where anyone with a small knowledge in electronics and programming could use his project to give wings to their creativity. His focus was to reduce the burden of complexity in designing electronics hardware and software. The project was named as Wiring. The software IDE of the Wiring used processing language to write the codes.

As the program written in C\C++ is named as Project, in the same way the code written in Wiring (even in Processing and Arduino) is termed as Sketch. The name sketch gives a familiar look for an artist.

Wiring has predefined libraries to make the programming language easy. Arduino uses these libraries. The predefined libraries are written in C and C++. One can even write his software in C\C++ and use them on Wiring boards. The difference between writing a program in C/C++ and Wiring is that the WiringApplication Programmable Interface (API) has simplified programming style and the user doesn’t require detailed knowledge of the concepts like classes, objects, pointers, etc. While sketching hardware you need to call the predefined functions and rest will be handled by the Wiring software.

The basic difference between the Processing and the Wiring is that the Processing is use to write the program which can be used on other computers while Wiring program is used on microcontrollers.

## Hardware

This board is designed around the ATmega328 AVR microcontroller. It is an 8 bit microcontroller with 32KB of flash, 2KB of SRAM, 1KB of EEPROM, timers, ADC, I2C, SPI, and UART peripherals. Arduino Uno is based on ATmega328P Atmel AVR family microcontroller (MCU). This MCU has 32KB of flash, 2KB of SRAM and 1 KB of EEPROM. It has 14 digital IO pins (PORTD

– 8pins, PORTC – 6 pins, PORTB – 5pins), 6 Analogue input pins, which can be sampled using on-chip ADC. It also has 6 PWM outputs multiplexed on to the digital IO pins. A 16 MHz crystal is installed on the board.

## Arduino Uno Pinout

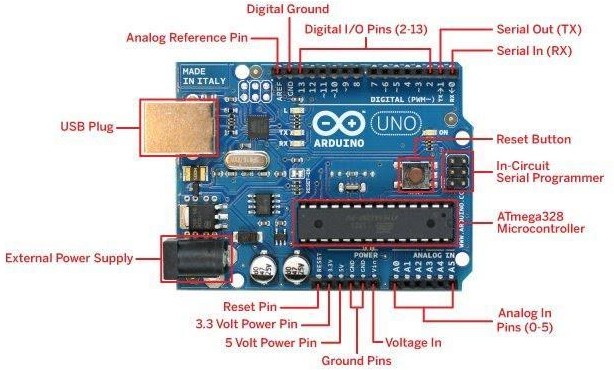


Fig 6.2 Arduino UNO pinout

**External Power Supply** - allows the Arduino to run when its not connected plugged into a USB port for power. It accepts between 7V- 12V of voltage.

**USB Plug** - This powers the Arduino without needing to use an external power supply and is what you use to upload sketches (program) to the microcontroller, and to communicate with your Arduino sketch (via Serial, println(), etc).

**AtMega328 Microcontroller** - The brains of the Arduino which you program through the USB plug. It contains three types of memory. It has 32KB of nonvolatile Flash memory. This is used to store applications and is stored on your board even after it is removed from it's power source. 2KB of volatile SRAM memory which is used to store variables used by applications while it's running. 1KB of EEPROM nonvolatile memory. This is used to store data that remains available even after the board is powered down and powered up again.

**Pin Functions:**

**Power Pins (3.3 V, 5 V, GND)** - Use these pins to connect to circuitry at 3.3 V, 5V, or GND. Make sure that whatever you power doesn't draw more than a few milliamps.

**Serial Out (TX) and Serial In (RX)** - Pins (0-1) are RX and TX respectively and used for sending and receiving serial data. This port can be used to send and receive data from a GPS module, bluetooth modules, WIFI modules, etc.

**Digital I/O Pins (2-13)** - Accept 0 to 5 V input or output. Utilizing tristate logic Arduino makes it easy to change between inputs and outputs in software. You can use this pin as an output where it spits out 5V for a digital 1, or 0 V for a digital 0. You can also configure it to expect a voltage on the pin and that voltage could be interpreted as a 1 or a 0. These pins are used with digitalRead(), digitalWrite (). analogWrite() works only on pins with PWM symbol.

**External Interrupts** - Pins 2 and 3 can be configured to trigger an interrupt on low value, a rising or falling edge, or a change in value.

**PWM Pins** - any pins with ~ in front of them can be used to generate pulse modulated square waves. Pins 3, 5, 6, 9, 10, and 11 provide 8-bit PWM output with the analogWrite() function.

**Pin 13** - drives the built in LED, that is used by Arduino to receive power and useful for debugging. When pin is HIGH value, the LED is on, when pin is LOW value, it's off.

**Analog In Pins** - Pins A0 through A5 provide 10 bits of resolution. Accepts 0 to 5 V inputs and is used to measure continuous voltages anywhere from 0 V to 5 V. It is possible to change the upper end of their range using the AREF pin and the analog Reference () function.

**Analog Reference Pin (AREF)** - input pin used optionally if you want external voltage reference for ADC rather than internal Vref. You can configure using an internal register.

**Reset Pin** - bring this line low to reset the microcontroller. Typically used to add a reset button to shields that block the one on the board.

**Memory -** The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed.

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

## IR Sensor

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it is still on the same electromagnetic spectrum). Anything that emits heat (everything that has a temperature [above around five degrees Kelvin](https://www.livescience.com/50260-infrared-radiation.html)) gives off infrared radiation.

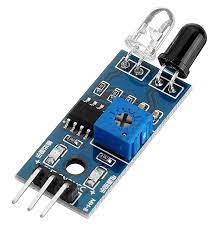


Fig 6.3 IR Sensor

## How IR Sensors work?

## There are different types of infrared transmitters depending on their wavelengths, output power and response time. An IR sensor consists of an IR LED and an IR Photodiode, together they are called as PhotoCoupler or OptoCoupler.

## https://robu.in/wp-content/uploads/2020/05/51fibl-5xL._SX342_.jpg

Fig 6.4 IR LED

Infrared Transmitter is a light emitting diode (LED) which emits infrared radiations called as IR LED’s. Even though an IR LED looks like a normal LED, the radiation emitted by it is invisible to the human eye. Infrared receivers or infrared sensors detect the radiation from an IR transmitter. IR receivers come in the form of photodiodes and phototransistors. Infrared Photodiodes are different from normal photo diodes as they detect only infrared radiation. Below image shows the picture of an IR receiver or a photodiode, Different types of IR receivers exist based on the wavelength, voltage, package, etc. When used in an infrared transmitter – receiver combination, the wavelength of the receiver should match with that of the transmitter.

The emitter is an IR LED and the detector is an IR photodiode. The IR photodiode is sensitive to the IR light emitted by an IR LED. The photo-diode’s resistance and output voltage change in proportion to the IR light received. This is the underlying working principle of the IR sensor.

**Applications**

An Infrared technology implemented in[**night vision equipment**](https://robu.in/product/5mp-ov5647-sensor-adjustable-wide-angle-fish-eye-lens-night-vision-camera-for-raspberry-pi-3-b/) if there is not enough visible light available to see unaided. Night vision devices convert ambient photons of light into electrons and then amplify them using a chemical and electrical process before finally converting them back into visible light.

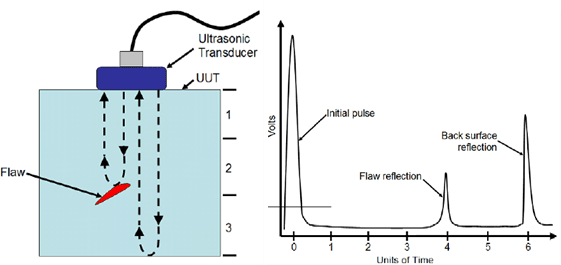
|  |  |  |
| --- | --- | --- |
| **Domain** | **Parameter** | **Applications** |
| Time | Tile-of-Flight, Velocity | Density, Thickness, Flaw Detection, Anisotropy, Robotics,  Remote Sensing etc. |
| Attenuation | Fluctuations in reflected  Transmitted Signals | Defect characterization,  microstructures, interface analysis |
| Frequency | IR Spectroscopy | Microstructure, grain size, porosity,  phase analysis. |
| Image | Time-of-Flight,velocity,  attenuation mapping in Raster C-Scan or SARs | Surface and internal Defect imaging, density, velocity, 2D and 3D imaging. |

Table 6.1 Applications of IR Sensors

IR sensors uses in radiation [thermometers](https://robu.in/product/gm320-non-contact-laser-lcd-display-digital-infraredir-thermometer-temperature-meter-gun-50%e2%84%83-330%e2%84%83/)to measure the temperature depend upon the temperature and the material of the object and these thermometers have some of the following features

* Measurement without direct contact with the object
* Faster response
* Easy pattern measurements

IR image device is one of the major applications of IR waves, primarily by virtue of its property that is not visible. It uses for thermal imagers,[night vision devices](https://robu.in/product/raspberry-pi-infrared-ir-night-vision-surveillance-camera-module-500w-webcam/) etc..



## Relay

It works on the principle of electromagnetism. The electromagnetic field that creates the temporary magnetic field is energised when the relay's circuit detects the fault current. This magnetic field moves the relay armature to open or close connections. A device that controls the flow of electrical power to a motor. It is designed to safely start and stop a motor, and provide overload protection. A set of contacts that has been designed to make or break the flow of current to a motor and can handle loads up to a specified amount of horsepower at a specific voltage.

Relays are normally used in the control panels, manufacturing, and building automation to control the power along with switching the smaller current values in a control circuit.



## ESP-12E

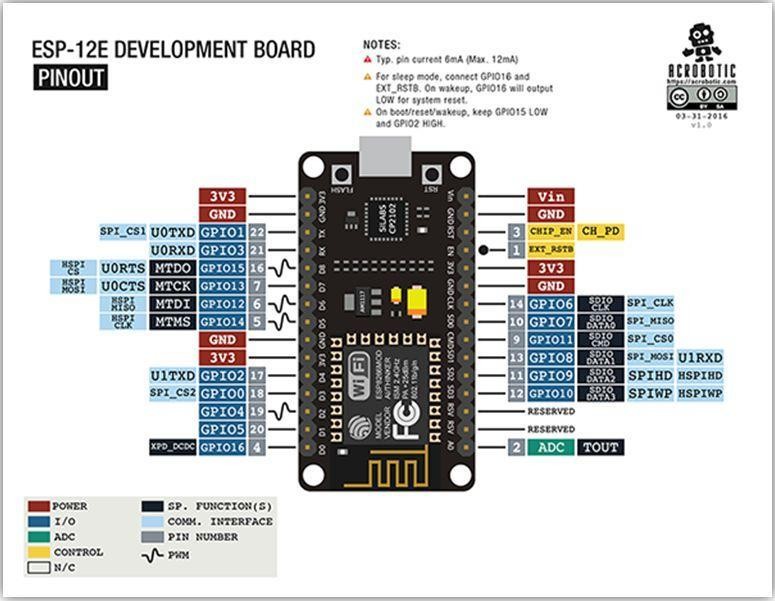
The ESP-12E is a board created by AI-THINKER, which consists of an ESP8266EX inside the metal cover.

## ESP8266EX



Made by Espressif, this microchip has integrated WiFi and low-power consumption. Processor RISC Tensilica L 106 32bit with a maximum clock of 160 MHz

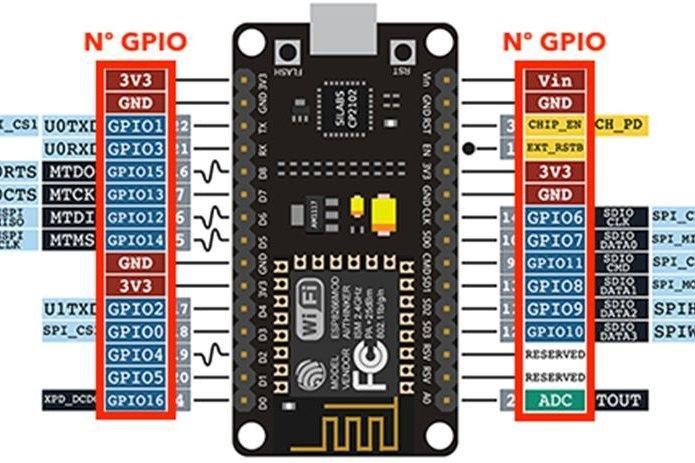
## Node MCU 1.0 ESP-12E Pinout



**ESP-12E Pinout**

I want to emphasize that NodeMCU and ESP-12E are not the same things. In the case of the ESP-12E, the recording uses the serial, the UART. In NodeMCU, this is performed by the USB.

## Step 6: And After All This, What's the Number to Put When Programming?



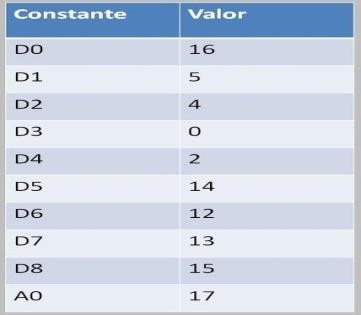
Use the number that is in front of the GPIO or the constants A0, D0, D1, D2, D3, D4, D5, D6, D7, and D8.

## Boot

We put the oscilloscope at the tip of each pin. This allows us to find, for

example, that when we turn on the NodeMCU, its pins are not all the same. Some are up and others down, by default. See the comments on the behavior of each post after the boot in the image below.

## Constants That Are Already Predefined



* **digitalWrite** did NOT work with GPIOs 6, 7, 8, 11, and ADC (A0)
* **digitalRead** did NOT work with GPIOs 1, 3, 6, 7, 8, 11, and the ADC (A0)
* **analogWrite** did NOT work with GPIOs 6, 7, 8, 11, and ADC (A0) (GPIOs 4, 12, 14, 15 have hardware PWM, and the others are by software)
* **analogRead** worked only with the ADC (A0)
* **6**, **7**, **8**, **11** do NOT work for the above four commands

## Atmega 328

The ATmega328 is a single-[chip](https://en.wikipedia.org/wiki/Integrated_circuit) [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) created by [Atmel](https://en.wikipedia.org/wiki/Atmel) in the [megaAVR](https://en.wikipedia.org/wiki/MegaAVR" \o "MegaAVR) family (later [Microchip Technology](https://en.wikipedia.org/wiki/Microchip_Technology) acquired Atmel in 2016). It has a [modified Harvard architecture](https://en.wikipedia.org/wiki/Modified_Harvard_architecture) [8-bit](https://en.wikipedia.org/wiki/8-bit) [RISC](https://en.wikipedia.org/wiki/Reduced_instruction_set_computer) processor core. ATmega328 is commonly used in many projects and [autonomous systems](https://en.wikipedia.org/wiki/Autonomous_system_(Internet)) where a simple, low-powered, low-cost micro-controller is needed. Perhaps the most common implementation of this chip is on the popular [Arduino](https://en.wikipedia.org/wiki/Arduino) development platform, namely the [Arduino Uno](https://en.wikipedia.org/wiki/Arduino_Uno), Arduino Pro Mini[[4]](https://en.wikipedia.org/wiki/ATmega328#cite_note-4) and [Arduino Nano](https://en.wikipedia.org/wiki/Arduino_Nano) models.

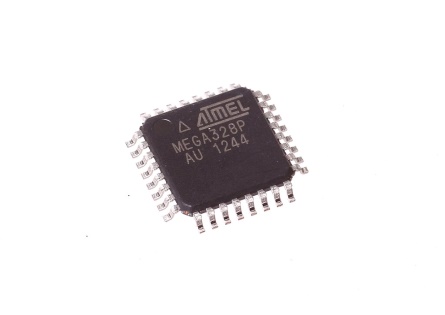


Fig 6.5 Atmega 328

The Atmel [8-bit](https://en.wikipedia.org/wiki/8-bit) [AVR](https://en.wikipedia.org/wiki/Atmel_AVR) [RISC](https://en.wikipedia.org/wiki/Reduced_instruction_set_computing)-based microcontroller combines 32 KB [ISP](https://en.wikipedia.org/wiki/In-system_programming) [flash](https://en.wikipedia.org/wiki/Flash_memory) memory with read-while-write capabilities, 1 KB [EEPROM](https://en.wikipedia.org/wiki/EEPROM), 2 KB [SRAM](https://en.wikipedia.org/wiki/Static_random-access_memory), 23 general-purpose I/O lines, 32 general-purpose working [registers](https://en.wikipedia.org/wiki/Processor_register), 3 flexible timer/[counters](https://en.wikipedia.org/wiki/Counter_(digital)) with compare modes, internal and external [interrupts](https://en.wikipedia.org/wiki/Interrupt), serial programmable [USART](https://en.wikipedia.org/wiki/USART), a byte-oriented 2-wire serial interface, [SPI](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus) serial port, 6-channel 10-bit [A/D converter](https://en.wikipedia.org/wiki/A/D_converter) (8 channels in [TQFP](https://en.wikipedia.org/wiki/Quad_Flat_Package) and [QFN](https://en.wikipedia.org/wiki/Quad_Flat_No-leads_package)/[MLF](https://en.wikipedia.org/wiki/Quad-flat_no-leads_package#Variants) packages), programmable [watchdog timer](https://en.wikipedia.org/wiki/Watchdog_timer) with internal [oscillator](https://en.wikipedia.org/wiki/Electronic_oscillator), and 5 software-selectable power-saving modes. The device operates between 1.8 and 5.5 volts. The device achieves throughput approaching 1 [MIPS](https://en.wikipedia.org/wiki/Million_instructions_per_second#Million_instructions_per_second)/MHz. Programming mode is entered when PAGEL (PD7), XA1 (PD6), XA0 (PD5), BS1 (PD4) is set to zero. RESET pin to 0 V and VCC to 0 V. VCC is set to 4.5–5.5 V. Wait 60 μs, and RESET is set to 11.5–12.5 V. Wait more than 310 μs. Set XA1:XA0:BS1:DATA = 100 1000 0000, pulse XTAL1 for at least 150 ns, pulse WR to zero. This starts the chip erase. Wait until RDY/BSY (PD1) goes high. XA1:XA0:BS1:DATA = 100 0001 0000, XTAL1 pulse, pulse WR to zero.

# SOFTWARE REQUIREMENTS

## 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.

## 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 righthand 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.

|  |  |
| --- | --- |
| Description: https://www.arduino.cc/en/uploads/Guide/play.png | *Verify*  Checks your code for errors compiling it. |
| Description: https://www.arduino.cc/en/uploads/Guide/export.png | *Upload*  Compiles your code and uploads it to the configured board. See [uploading](https://www.arduino.cc/en/Guide/Environment#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" |

|  |  |
| --- | --- |
| Description: https://www.arduino.cc/en/uploads/Guide/new.png | *New*  Creates a new sketch. |
| Description: https://www.arduino.cc/en/uploads/Guide/open.png | *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 | Sketchbookmenu instead. |
| Description: https://www.arduino.cc/en/uploads/Guide/save.png | *Save*  Saves your sketch. |
| Description: https://www.arduino.cc/en/uploads/Guide/serial_monitor.png | *Serial Monitor*  Opens the [serial monitor](https://www.arduino.cc/en/Guide/Environment#serialmonitor). |

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

## File

* + **New**

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

## Open

Allows to load a sketch file browsing through the computer drives and folders.

## Open Recent

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.

## Save as...

Allows to save the current sketch with a different name.

## Page Setup

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

* + **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.

## Copy for Forum

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

## Copy as HTML

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.

## Select All

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/Decrease Indent

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.

## Find Next

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

## Find Previous

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

## 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 bootloader on the board; you will need to use Tools > Burn Bootloader 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](https://www.arduino.cc/en/Guide/Environment#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 o the toolbar.

## Tools

* + **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.

## Archive Sketch

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

## 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](https://www.arduino.cc/en/Guide/Environment#boards) [boards](https://www.arduino.cc/en/Guide/Environment#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.

## Programmer

For selecting a harware 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 bootloader](https://www.arduino.cc/en/Tutorial/Bootloader) to a new microcontroller, you will use this.

## Burn Bootloader

The items in this menu allow you to burn a [bootloader](https://www.arduino.cc/en/Hacking/Bootloader) 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 come without a bootloader).

* + Ensure that you've selected the correct board from the Boards menu before burning the bootloader 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.

## Find in Reference

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.

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

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The [boards](https://www.arduino.cc/en/Guide/Environment#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 File 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 bootloader 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 #includestatements from the top of your code.

There is a [list of libraries](https://www.arduino.cc/en/Reference/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](https://www.arduino.cc/en/Guide/Libraries) [for installing a third-party library](https://www.arduino.cc/en/Guide/Libraries).

## 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.

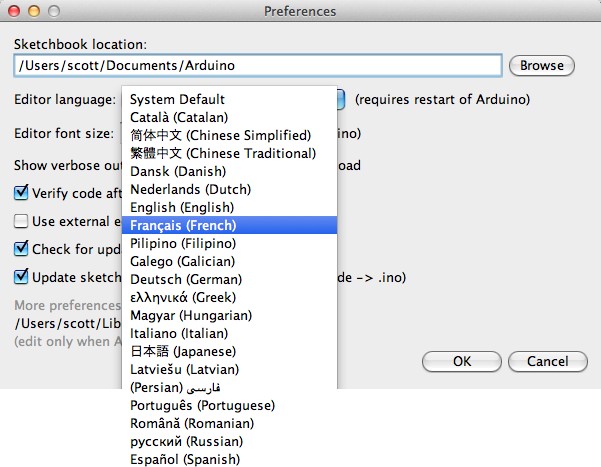
## Serial Monitor

Displays serial data being sent from the Arduino or Genuino board (USB or serial board). 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 that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux, the Arduino or Genuino board will reset (rerun your sketch execution to the beginning) when you connect with the 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.

## 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 selectingSystem 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.

Boards

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader. You can find a comparison table between the various boards [here](https://www.arduino.cc/en/Products/Compare).

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The[Boards Manager](https://www.arduino.cc/en/Guide/Cores) included in the standard installation allows to add support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, Galileo and so on.

## EMBEDDED C

An embedded system is a [computer system](http://en.wikipedia.org/wiki/Computer_system) designed to perform one or a few dedicated functions often with [real-time computing](http://en.wikipedia.org/wiki/Real-time_computing) constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a [personal computer](http://en.wikipedia.org/wiki/Personal_computer) (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Embedded systems are controlled by one or more main processing cores that are typically either [microcontrollers](http://en.wikipedia.org/wiki/Microcontroller) or [digital signal processors](http://en.wikipedia.org/wiki/Digital_signal_processor) (DSP).The key characteristic, however, is being dedicated to handle a particular task, which may require very powerful processors. Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from [economies of scale](http://en.wikipedia.org/wiki/Economies_of_scale).

Peripherals

Embedded Systems talk with the outside world via [peripherals](http://en.wikipedia.org/wiki/Peripheral), such as:

* + - Serial Communication Interfaces (SCI): [RS-232](http://en.wikipedia.org/wiki/RS-232), [RS-422](http://en.wikipedia.org/wiki/RS-422), [RS-485](http://en.wikipedia.org/wiki/RS-485) etc.
    - Synchronous Serial Communication Interface: [I2C](http://en.wikipedia.org/wiki/I2C), [SPI](http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus), SSC and ESSI (Enhanced Synchronous Serial Interface)
    - [Universal Serial Bus](http://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB)
    - Analog to Digital/Digital to Analog ([ADC](http://en.wikipedia.org/wiki/Analog-to-digital_converter)/[DAC](http://en.wikipedia.org/wiki/Digital-to-analog_converter))

## Advantages

* + - * Reliability
      * Simple control loop
      * Interrupt controlled system

# CHAPTER 7

## RESULT AND DISCUSSION

We observed that the mobile charging system using coin insertion achieved an average charging efficiency. This indicates how effectively the system converted the coins into electrical energy for the mobile.

Feedback from users revealed that the coin insertion method was user-friendly, with a satisfaction. Most users found it convenient and easy to use, making it a viable option for public mobile charging stations.

Over the course of the study, we collected the amount of revenue from the coin-based charging system. This revenue can be reinvested in maintaining and expanding the charging infrastructure.

The system demonstrated high reliability, with only a minimal number of reported issues. Certain users reported no problems with the charging process,while encountered minor issues that were resolved promptly.

The coin-based charging system appears to be a cost-effective solution for public mobile charging. It eliminates the need for complex payment infrastructure and reduces administrative costs.

While coin insertion is user-friendly, it's important to consider integrating modern payment methods such as mobile payments or credit cards to cater to a wider range of users.

# CHAPTER 8

## CONCLUSION AND FUTURE WORK

Our paper describes the new way of providing charging services to the public. It would be of less cost because conventional grid power is used and beneficial to the long distance travelers. This coin based mobile charging system can be installed at various public places for the convenience of mobile users. The objective of this paper is to help mobile users by providing coin based charging that would be easily available whenever they need it.

The Share & Charge Foundation develops tools based on distributed technologies, like block chain technology, to support solutions for mobile phones. 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.

## APENDICES

Creating appendices for a project or report on mobile charging using coin insertion typically involves including additional information, data, or documentation that supports and complements the main content. Here are some items you might consider including in your appendices:

1.Data Tables: Include detailed data on charging station usage, coin collection, revenue generated, etc.

2.Technical Diagrams: Schematics or technical drawings of the charging system.

3.User Surveys: Any surveys or questionnaires used to gather user feedback.

4.Financial Projections: Detailed financial projections for the project, including revenue, expenses, and ROI calculations.

5.Coin Handling Mechanism: Detailed explanations or diagrams of how coins are handled within the system.

6.Regulatory Compliance: Any documentation related to legal and safety compliance for the charging station.

7.Case Studies: If applicable, include any case studies or examples of successful coin-based EV charging stations.

8.Supporting Documentation: Any additional documents that provide context or further explanation for the project.

## REFERENCES

## [1] Y. Yan, Q. Li, W. R. Chen, W. Q. Huang, and J. W. Liu, “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.

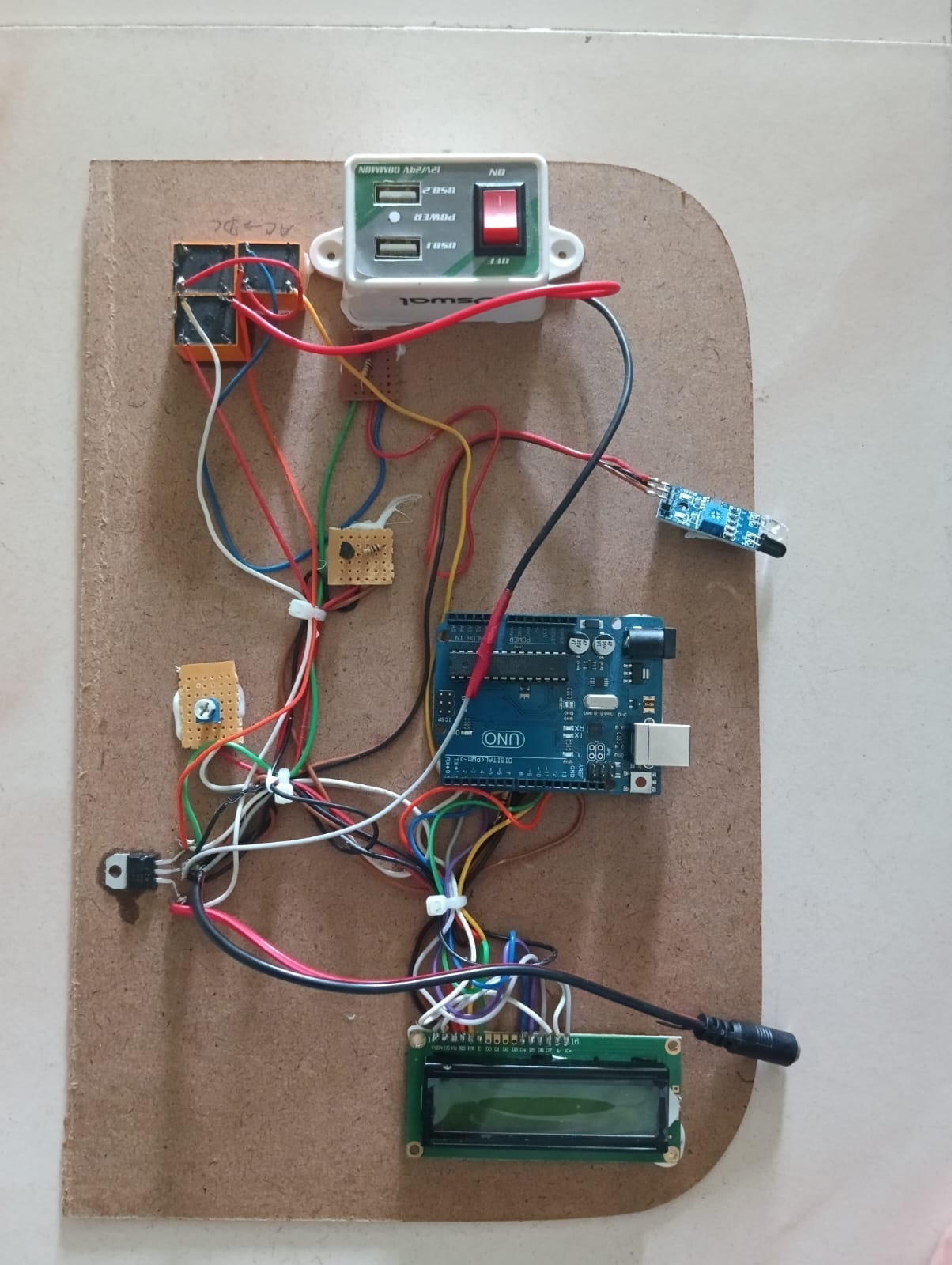
## [2] L. Zhang, X. S. Hu, Z. P. Wang, F. C. Sun, J. J. Deng, and D. G. Dorrell, “Multi objective optimal sizing of hybrid energy storage system for electric vehicles,” IEEE Transactions on Vehicular Technology, vol. 67, no. 2, pp. 1027–1035, Feb. 2018.

## [3] Q. Li, B. Su, Y. C. Pu, Y. Han, T. H. Wang, L. Z. Yin, and W. R. Chen, “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] Q. Yan, B. Zhang, and M. Kezunovic, “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.

## [5] Q. Y. Sun, S. Chen, L. Chen, and D. Z. Ma, “Quasi-Z-source networkbased hybrid power supply system for aluminum electrolysis industry,” IEEE Transactions on Industrial Informatics, vol. 13, no. 3, pp. 1141– 1151, Jun. 2017.

**PROJECT KIT**

****

