Kalyani Government Engineering College

Kalyani, West Bengal 741235

Department of Electronics & Communication Engineering

Report on

Arduino based MP3 Player

Name :	Samrat Das
Stream :	Electronics & Communication Engineering
Semester :	5th ; **Year :** 3rd
Roll No. :	10200322053
Reg. No. :	221020120310
Session :	2023 – 2024

Remarks

Signature of Examiner

Acknowledgement

I would like to thank my teacher **Prof. Subhashis Maitra** who gave me this opportunity to work on this project. I got to learn a lot from this project about several unknown electronics components and have also gained knowledge on several simulators. I would also like to thank our principal **Sourav Kumar Das**.

At last, I would like to extend my heartfelt thanks to **my parents** because without their help this project would not have been successful. Finally, I would like to thank **my Classmates Nilavo Roy (10200322052), Anshuman Ghorai (10200322051), Moumita Biswas (10200322055)** who has been with me all the time.

Title : Arduino based mp3 player

Bachelor in Technology : Bachelor in Electronics & Communication Engineering

Author : Samrat Das

Co-ordinators : Moumita Biswas , Nilavo Roy , Anshuman Ghorai

Director : Prof. Subhashis Maitra

Date :

Abstract

The aim of the project is to design a low cost efficient Digital Mp3 Player using Arduino. Arduino has become a popular open-source single-board microcontroller among electronic hobbyists, and it is gaining acceptance as a quick prototyping tool for engineering and educational projects also.

A computer, using a control application done in LabVIEW or IDE Arduino, must control all the system by cable or wireless.

Now, one can ask a question that is why Mp3 players because a Smartphone can do better than this.

When it comes to sound quality, MP3 players have the upper hand over smartphones. The reason behind this lies in their dedicated audio hardware and software optimizations. Unlike smartphones that are designed to perform multiple tasks simultaneously, MP3 players focus solely on delivering high-quality audio output.

These whole project has been divided into five chapters containing several meanings are as follows –

The chapter **One** is all about the project definition is introduced detailing each one of the requirements and specifications must meet the overall system.

The **second** chapter tells all about the difficulties those have been faced. Depending upon which the planning has been changed contiguously. The solutions of those difficulties have been attached also with the problems.

The **third** chapter shows the platform selected to implement the overall system. For the selection of each component, it has considered different alternatives according to the requirements specified on the project definition.

In the **Fourth** chapter the system has been tested physically and the results those meet our expectations or not.

Finally in the **Last / fifth** Chapter, it deploys an application to check the correct performance of the overall system. With this objective, it has been considered various scenarios using some components of the platform. Basically, in the scenario where all the stages happen on the Arduino module is known as autonomous system. In contrast, when these stages are divided between the Arduino module, and other system, the scenario is known as dependent system.

Index

Introduction

MP3 Player vs. Smartphone: Which is better for Music Lovers?

In today's digital age, music has become an integral part of our lives. Whether we're commuting to work, working out at the gym, or simply relaxing at home, having access to our favourite tunes is a must. With the rise of smartphones, many might argue that MP3 players have become obsolete. However, there are still several reasons why music lovers might prefer an MP3 player over a Smartphone when it comes to enjoying their favourite tracks. In this article, we will compare the two devices and explore which one is better suited for music enthusiasts.

Sound Quality

When it comes to sound quality, MP3 players have the upper hand over smartphones. The reason behind this lies in their dedicated audio hardware and software optimizations. Unlike smartphones that are designed to perform multiple tasks simultaneously, MP3 players focus solely on delivering high-quality audio output. This results in superior sound reproduction with clearer highs and deeper lows.

Additionally, many MP3 players offer customizable equalizer settings that allow users to fine-tune the sound according to their preferences. This level of control over the audio output is often lacking in smartphones where sound quality can be compromised due to various factors like background processes or limited audio processing capabilities.

Battery Life

One significant advantage of MP3 players over smartphones is their impressive battery life. Smartphones are notorious for draining their batteries quickly when used for media consumption purposes like playing music or streaming videos. On the other hand, MP3 players are specifically designed to optimize battery usage and provide extended playback time.

With an MP3 player, you can enjoy hours upon hours of uninterrupted music without worrying about running out of battery power halfway through your playlist. This makes them perfect for long commutes or road trips where charging options may be limited.

Storage Capacity

Another area where MP3 players shine is storage capacity. While smartphones offer a range of storage options, they are often limited and can quickly fill up with various apps, photos, and videos. This leaves little room for storing an extensive music library.

MP3 players, on the other hand, are specifically designed to accommodate large music collections. They come with generous internal storage capacities and often have expandable memory options through microSD cards. This allows music lovers to carry their entire music library with them wherever they go without worrying about running out of space.

Distraction-Free Listening Experience

One of the main advantages of using an MP3 player for enjoying music is the distraction-free listening experience it provides. Smartphones are multi-purpose devices that can be a source of distractions with constant notifications, calls, and app temptations. On the other hand, MP3 players are dedicated devices solely focused on providing a seamless and uninterrupted music experience.

By using an MP3 player, you can disconnect from the digital noise around you and immerse yourself in your favorite tracks without any interruptions. This can greatly enhance your overall music listening experience and allow you to appreciate the artistry behind each song.

In conclusion, while smartphones offer convenience and versatility in many aspects of our lives, MP3 players still hold their ground when it comes to delivering an exceptional music listening experience. With superior sound quality, impressive battery life, ample storage capacity, and a distraction-free environment, MP3 players remain a viable choice for individuals who consider themselves true music lovers.

Here we develop a mini Arduino powered MP3 player than can play songs from an SD card. The system makes is not only for playing music but also can be used as a reverse parking indicator player, miniature speaker, instruction playing etc.

To develop this system we make use of an Arduino Pro Mini, SD card reader, OLED Display, Speaker Module / 3.5mm Female Audio Port, basic electronics components and PCB board to develop the system. The SD card reader connects to the Arduino using serial communication. We now use the Arduino controller to play the songs from the SD card on the speaker. Also we provide 3 buttons on the PCB for 3 Commands:

- ✓ Play/Pause
- ✓ Previous Song
- ✓ Next Song

These whole project has been divided into five chapters containing several meanings are as follows –

The chapter **One** is all about the project definition is introduced detailing each one of the requirements and specifications must meet the overall system.

The **second** chapter tells all about the difficulties those have been faced. Depending upon which the planning has been changed contiguously. The solutions of those difficulties have been attached also with the problems.

The **third** chapter shows the platform selected to implement the overall system. For the selection of each component, it has considered different alternatives according to the requirements specified on the project definition.

In the **Fourth** chapter the system has been tested physically and the results those meet our expectations or not.

Finally in the **Last / fifth** Chapter, it deploys an application to check the correct performance of the overall system. With this objective, it has been considered various scenarios using some components of the platform. Basically, in the scenario where all the stages happen on the Arduino module is known as autonomous system. In contrast, when these stages are divided between the Arduino module, and other system, the scenario is known as dependent system.

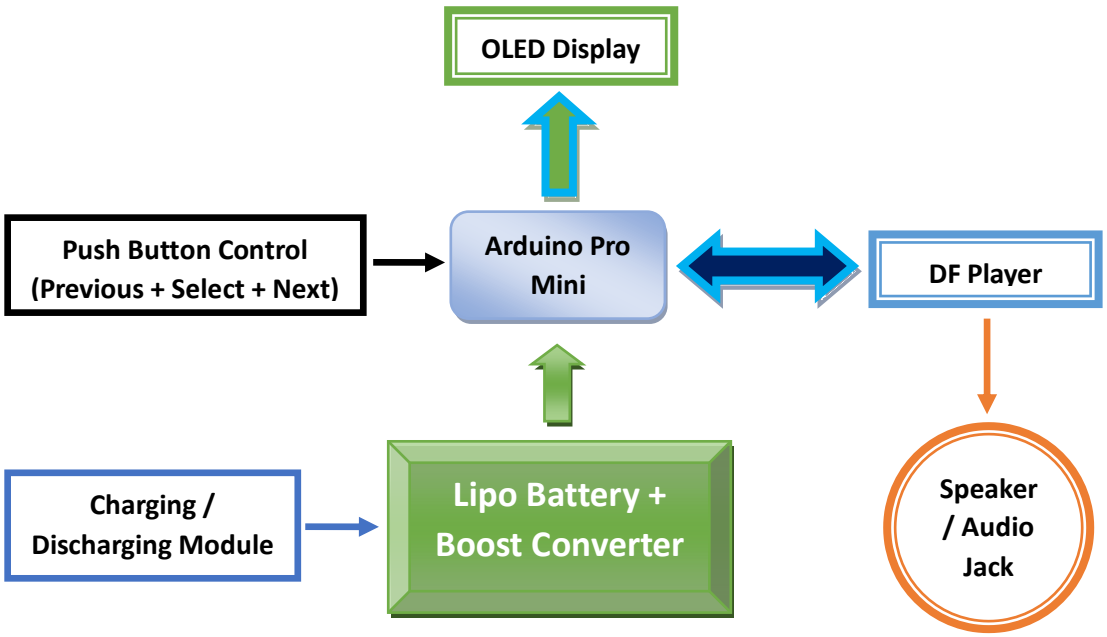
Chapter 1 : Project Definition

In this chapter, it will be defined the objectives and the main specifications and requirements to develop the system.

2.1 Objectives

In today's digital age, music has become an integral part of our lives. Whether we're commuting to work, working out at the gym, or simply relaxing at home, having access to our favourite tunes is a must. With the rise of smartphones, many might argue that MP3 players have become obsolete. However, there are still several reasons why music lovers might prefer an MP3 player over a Smartphone when it comes to enjoying their favourite tracks. In this article, we will compare the two devices and explore which one is better suited for music enthusiasts.

A detailed block diagram of the project has been given below –



The objectives of an Arduino-based MP3 player project typically revolve around creating a functional device capable of playing MP3 audio files. Here are some common objectives:

1. **Audio Playback:** Develop a system that can reliably decode and play MP3 audio files stored on a storage medium such as an SD card or USB drive.
2. **User Interface:** Design an intuitive user interface for controlling playback, including functions like play, pause, stop, volume control, and track selection.
3. **Hardware Integration:** Integrate necessary hardware components such as an Arduino board, an MP3 decoder module, an audio amplifier, and an audio output device (e.g., speaker or headphones).

4. **File Management:** Implement file management functionalities to navigate through MP3 files stored on the storage medium, including browsing folders and selecting specific tracks.
5. **Display:** Incorporate a display screen to provide feedback to the user, such as track information, playback status, and menu navigation.
6. **Customization:** Allow for customization options such as equalizer settings, repeat modes, shuffle functionality, and possibly even playlist creation.
7. **Power Management:** Optimize power consumption and implement features like standby mode or auto power-off to conserve energy when the device is not in use.
8. **Expandability:** Design the system with expansion capabilities, allowing users to add features or modify the functionality according to their needs, such as connecting additional sensors or peripherals.
9. **Documentation:** Provide comprehensive documentation, including schematics, code explanations, and assembly instructions, to assist others in replicating or modifying the project.
10. **Reliability and Performance:** Ensure the system operates reliably without glitches or interruptions during playback, and strive for good audio quality and performance.

By achieving these objectives, the Arduino-based MP3 player can serve as an educational tool, a hobbyist project, or even a commercial product catering to specific audio playback needs.

Creating an Arduino-based MP3 player can fulfill several needs:

1. **Entertainment:** It provides a means to listen to music or audio files, offering entertainment and relaxation.
2. **Customization:** By building it yourself, you can tailor the player to your specific preferences, such as adding features like equalizers, playlists, or custom interfaces.
3. **Learning and Skill Development:** Building an Arduino-based MP3 player involves learning about electronics, programming, and interfacing different components. It's an excellent way to develop technical skills.
4. **Cost-Effectiveness:** Arduino-based projects often use affordable components, making them a cost-effective solution compared to commercial MP3 players.
5. **Creative Expression:** You have the freedom to design and personalize the player according to your aesthetic preferences, adding LEDs, displays, or unique enclosures.
6. **Functionality Expansion:** Arduino allows for easy integration with other sensors or modules. You can expand the functionality of your MP3 player by adding features like Bluetooth connectivity, voice control, or even IoT capabilities.
7. **Accessibility:** You can customize the user interface to make it more accessible for specific needs, such as larger buttons or voice commands, catering to users with disabilities.

2.2 Specifications and technical requirements

In this paragraph, all the specifications and technical requirement to take into account on the development of this project will be described.

Total components those are required for our project have been divided into two parts –

- **Electronic Components**
- **Non-electronic components**

[*The (Star) marked components has been selected as per sudden requirements. Hence the actual quantity has not been shown here.]

☑ **Electronic Components :**

Sl. No.	Component	Specification	Quantity
1.	Arduino NANO + USB Cable	**Microcontroller:** ATmega328. **Operating Voltage:** 5V. **Input Voltage:** 7-12V. **Digital I/O Pins**: 22 (of which 6 provide PWM output) **Analog Input Pins:** 8. **DC Current per I/O Pin**: 20 mA (I/O Pins). **SRAM:** 2 KB. **Clock Speed**: 16 MHz. **EEPROM:** 1KB. **PWM Output**: 6. **Power consumption:** 19 mA. **Weight:** 7g. **Product Code:** A000005.	× **1**
2.	MT3608 Step Up Module	**Input Voltage**: 2-24V DC. **Output Voltage:** 5-28V DC. **Maximum Output Current:** 2A. **Switching Frequency:** 1.2Mhz. **Output Ripple:** <100mV. **Module Size:** 37.2mm x 17.2mm x 14.0mm. **Efficiency:** About 93%.	× **1**
3.	DF Player Mini MP3 player	**Supported sampling rates (kHz)** : 8/11.025/12/16/22.05/24/32/44.1/48. **DAC output:** 24 -bit (support for dynamic range 90dB, SNR support 85dB). **Audio data sorted by folder supports:** up to 100 folders. **Volume:** 30 level adjustable volume. **EQ**: 6 -level EQ adjustable.	× **1**

4.	1.3inch OLED Display	**Display Size:** 1.3inch. **Resolution:** 128x64. **Interface:** IIC. **Colour:** Blue. **Driver IC:** SSD1306. **Input Voltage:** 3.3V/5V. **Pins:** 4P. **Operating Temperature:** -30~70℃. **Storage temperature:** -40~80℃.	× **1**
5.	DC Voltmeter	**Accuracy:** 0.1 V. **Color**: Red. **Wires**: 3. **Display ON Voltage (V):** 3. **Measurable Voltage Range**: 0 to 100V. **Current Consumption (mA):** 10. **Refresh Speed (ms):** 200 (one time). **Minimum Input (V):** 3. **Highest Input (V):** 30. **Operating Temperature (C):**-10 to 60. **Holes for Mounting:** 0.3 cm. **Height (cm):** 0.72 cm. **Length (cm):** 1.3 cm. **Width (cm):** 3 cm. **Weight (gm):** 3 gm.	× **1**
6.	TP 4056 charging Module - Type C	**Input interface:** Type C USB. **Current:** 1A adjustable. **Charge precision**: 1.5%. **Input voltage:** 4.5V-5.2V. **Full charge voltage:** 4.2V. **Battery over-current protection:** 3 A. **The battery Undervoltage protection:** 2.5 V.	× **1**
7.	Mini Lithium Polymer Battery	**Product:** Small Li-Po Battery. **Voltage:** 3.7v. **Capacity**: 430mAh. **Discharge Rate:** 0.5C. **Weight:** 0.05g. **Dimensions:** 15 x 10.5 x 4.5mm.	× **1**
8.	Male to Male Jumper Wire Set	**Gender Type:** Male. **Pin Style:** Square Pins. **Number of Pins:** 40. **Pin Spacing:** 2.54 mm.	× **1**

9.	Customize Printed Circuit Board (PCB)	**Length:** 111.67mm. **Width:** 45.594mm. **Board Area:** 51.1. **PCB Thickness:** 1.6mm. **Copper Thickness:** 1 oz(35um). **Mask Colour:** Green. **Finish Type:** HASL Finish. **No. of Layers:** 2. **Shipping Method:** DTDC Standard.	× **1**
10.	Push Button - Micro Switch (Long Nose)	**Type:** normally open. **Rated load**: DC 12V 50mA. **Contact resistance:** m0.03 Ohm. **Insulation resistance:** 100M Ohm. **Temperature:** -30 to +70 degree Celsius. **Withstand voltage:** AC 250V.	× **3**
11.	1k resistor	**Product Type:** Metal Film Resistor. **Resistance Value:** 1K Ohm. **Power Rating:** 250 mW (1/4 W). **Tolerance:** ±1%.	× **1**
12.	SPDT Slide Switch	**Material:** Plastic and Metal. **Contact Type:** SPDT. **Position:** 2. **Terminal Pins:** 3. **Contact Resistance:** 20M. **Colour:** Black. **Voltage Max:** 50V. **Current Max:** 0.5A. **Approx Hole Diameter**: 2.5mm. **Approx Size in mm Length:** 22mm. **Width:** 7mm. **Height:** 12mm.	× **1**
13.	3.5 mm Female headphone jack	**Gender**: Female Jack. **Current Rating:** 0.5A at 30V. **Contact Resistance:** 30MΩ. **Withstand voltage:** 500 VAC. **Insulation Resistance:** 100MΩ. **Life Test:** 5000 Cycle. **No of Pin:** 5Pin. **Mounting:** PCB Mount.	× **1**

14.	*Female Header Pins	**Housing:** 30% Glass filled PBT / PA6T UL94V-0. **Contacts:** Brass or Phosphor Bronze . **Plating:** Au or Sn over 50u" Ni. **Current Rating:** 3 AMP. **Insulator Resistance:** 1000M Ohm min. at DC 1000V. **Contact Resistance:** 20m Ohm max. **Operating Temperature:** -40ºC~+105º.	-
15.	*Male Header Pins	**Housing**: 30% Glass filled PBT / PA6T UL94V-0. **Contacts:** Brass or Phosphor Bronze. **Plating**: Au or Sn over 50u" Ni. **Current Rating:** 3 AMP. **Insulator Resistance:** 1000M Ohm min. at DC 1000V. **Contact Resistance:** 20m Ohm max. **Operating Temperature**: -40ºC~+105º.	-

☑ **Non - electronic Components :**

Sl. No.	Component	Specification	Quantity
1.	3.5 mm Earphone	**Ear speaker impedance**: 16 ohms or higher. **Mic DC resistance:** 1000 ohms or higher. **Control Function Equivalent impedance:** 0 ohm. 240 ohm +/- 1% resistance. 470 ohm +/- 1% resistance. 135 ohm +/- 1% resistance.	× **1**
2.	16 GB Memory Card	**Storage Capacity:** 16GB . **Storage type**: UHS-I / Class 10. **Max Read Speed:** 80 MB/s. **Min Write Speed:** 10 MB/s.	× **1**
3.	Acrylic Sheet Transparent	**Material:** Acrylic Brand BIGIMALL. **Item dimensions:** L x W x H = 12 x 146 x 146 mm. **Colour:** Clear. **Item Form:** Sheet.s	× **1**
4.	Carrying Box	**Size:** Length- 6inch, Width- 4inch, Height- 3 inch. **Material:** Wood. **Weight:** 280grams.	× **1**

Chapter 2 : Challenges faced & Solutions

During making the project part by part we have faced several challenges. And for that we planning have been changed repeatedly. Here we discuss all the challenges we faced along with their solutions means how we coped up with them.

3.1 Selection of Proper Arduino board

As this project is completely Arduino based, hence we have considered some factors those should be fulfilled by our main Arduino board.

- **Portability / Small form factor**

MP3 player is a handy device. Now keeping it in mind we have should select the smallest Arduino board available in the market. Now there are two options –

1. Arduino Nano
2. Arduino Pro mini

- **Lack of memory**

Now both Arduino Nano and Arduino Pro mini are quite similar but Arduino Nano has 32 KB flash memory where pro mini has 16 Kb only. But our code is size was so big that it was already occupying 92% of the flash memory which may cause stability issue for which we decided to choose the *Arduino Nano* for our project.

- **Insufficient Power supply**

In this project the after calculating we have come to know that the Arduino Pro min was insufficient to fulfil the power requirements of the total circuitry. Hence we have decided to go with Arduino Nano.

3.2 Problem with Arduino power supply & selection of proper boost converter

According to the datasheet it has been known that VCC / 5V pin will give the voltage output of proper 5V along with a good amount of current rating if the Input voltage lies between 7V – 12V.

But to provide the supply the whole we have to use a single battery which is of 3.7V. Hence we need a voltage booster circuit / module maintains the small and handy form factor. For that Voltage boost converter *'MT3608'* is selected because its i/p and o/p voltage range is : **2V-24V DC** to **5V-28V** which was more than enough for our requirement. By varying the inbuilt potentiometer we set the voltage at 9 V DC and that voltage was sufficient to give supply and to get proper output power from Arduino Nano.

3.3 Visual Problem with display & selection of OLED

As a lot of information will be printed on the display hence we researched for such a display which will be tiny and was capable to fulfil the requirement.

The one and only solution was 1.3 Inch OLED Display.

3.4 Selection of Proper display library file for programming

'1.3 inch OLED' works with **'SH 1106'** library. Now there is another display '0.96 inch OLED' which may look similar like 1.3 inch OLED but it comes with different library file named as 'Adafruit SSD 1306'. Now there may arise a misconception.

To avoid that we have used a universal library file named 'U8g2' that works with all kinds of display.

3.5 Limitations of DF Player Mini module & Selection of Memory Card

For reading the mp3 files from SD card we have used DF Player Mini module that comes with lots of features. But it has a big limitation i.e. Memory size supported by this mp3 module is from 2 GB – 32 GB. For that we have chosen a *16 GB SD card* module which is rarely available in today's market.

3.6 Compactness v/s Battery backup

Battery capacity refers to the amount of energy a battery can store, while compactness refers to the space it takes up. They are inversely proportional to each other means if you increase battery capacity then size will be increased means compactness will decrease and vice-versa.

But in our project we cannot compromise with Compactness. For that reason the high density *Lithium Polymer battery* has been chosen to use as the main power distributer / source. Along with this it is also lightweight so the overall weight of the device will be very much less. And hence the device can be easily carried out in pocket.

3.7 Look v/s protection i.e. selection of enclosure

In commercial market there is a truth that If you run after look then you must have to sacrifice about protection. But the converse may not always true.

Researching on the material that matches perfectly with the converse we discovered that Acrylic sheets are such materials that can give a good durability and protection as well. Hence to make a good and durable enclosure of this model we have used '*Transparent Acrylic sheets'* for our project.

3.8 Problem with Boost converter and installation of Voltmeter

The boost converter that we have chosen 'MT 3608' has a problem that whenever the input voltage i.e. the battery voltage becomes lower than 2.5V then the module boosts the voltage up to its maximum limit i.e. 28V which is greater than the supportive voltage rating of all the other components. This may damage the components. For that we have one and only option that is to monitor the battery voltage continuously. For that we have used a tiny *DC voltmeter* that will perform as a live battery heath monitor.

Chapter 3 : Prototype Implementation

In this chapter, it will explain all the components used to design the whole system, detailing all the specifications, technologies and any particularities about each of them.

4.1 Platform

A platform is defined as a system that serves as a base to run a series of elements, either hardware or software.

For implementation of the Project (Automatic Sanitizing Door with Headcounter) we have chosen the Arduino platform according to the characteristics of out project elements.

What is Arduino?

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

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive -** Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50

- **Cross-platform -** The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- **Simple, clear programming environment -** The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- **Open source and extensible software -** The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- **Open source and extensible hardware -** The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

How do I use Arduino?

See the getting started guide. If you are looking for inspiration you can find a great variety of Tutorials on Arduino Project Hub.

4.2 Hardware

In this section, we will explain the details about the hardware elements of the prototype. According to our project the total hardware parts has been subdivided into two categories –

- Electronics Components
- Non – electronic Components

❑ **Electronic Components:**

An electronic component is any basic discrete device or physical entity in an electronic system used to affect electrons or their associated fields. Electronic components are mostly industrial products, available in a singular form and are not to be confused with electrical elements, which are conceptual abstractions representing idealized electronic components and elements.

Electronic components have a number of electrical terminals or leads. These leads connect to other electrical components, often over wire, to create an electronic circuit with a particular function (for example an amplifier, radio receiver, or oscillator). Basic electronic components may be packaged discretely, as arrays or networks of like components, or integrated inside of packages such as semiconductor integrated circuits, hybrid integrated circuits, or thick film devices. The following list of electronic components focuses on the discrete version of these components, treating such packages as components in their own right.

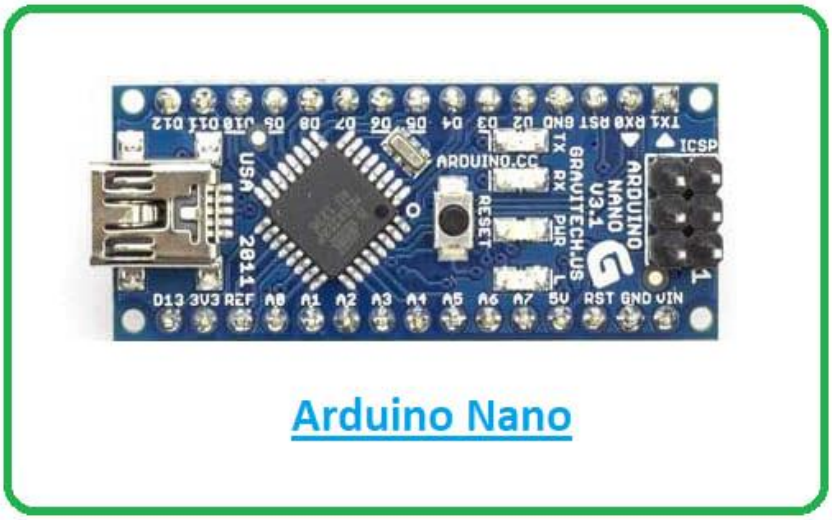
4.2.1 Arduino Nano Board

In this post today, I'll walk you through the Arduino Nano in detail as I'll cover each and everything related to Arduino Nano including what is Arduino Nano, its features, pinout, programming, and applications.

What is Arduino Nano?

Arduino Nano is a small, compatible open-source electronic development board based on an 8-bit AVR microcontroller. Two versions of this board are available, one is based on ATmega328p, and the other on Atmega168.

Arduino Nano can perform some functions similar to other boards available in the market, however, it is smaller in size and is a right match for projects requiring less memory space and fewer GPIO pins to connect with.



Arduino Nano

This unit features 14 digital pins which you can use to connect with external components, while 6 analog pins of 10-bit resolution each, 2 reset pins, and 6 power pins are integrated on the board.

Like other Arduino boards, the operating voltage of this device is 5V, while input voltage ranges between 6V to 20V while the recommended input voltage ranges from 7V to 12V.

The clock frequency of this unit is 16MHz which is used to generate a clock of a certain frequency using constant voltage.

The board supports a USB interface and it uses a mini USB port, unlike most Arduino boards that use the standard USB port. And there is no DC power jack included in this unit i.e. you cannot power the board from an external power supply.

Plus, this device is bread-board friendly in nature means you can connect this unit with breadboards and make a range of electronic projects.

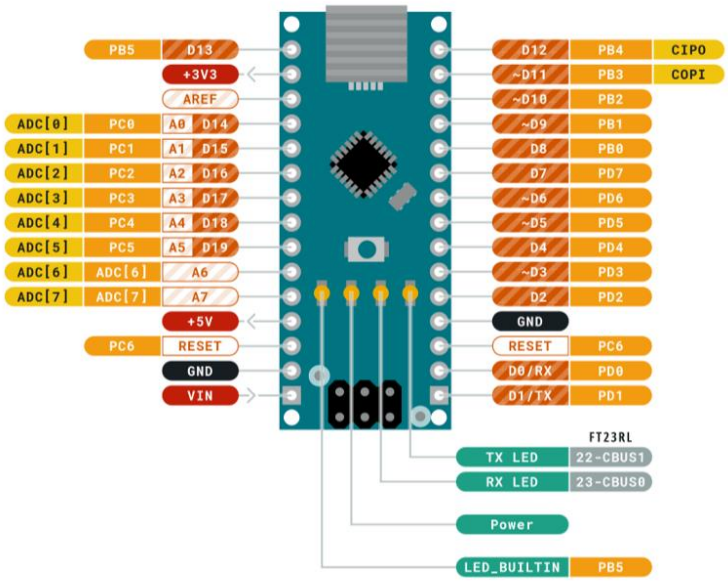
The flash memory is used to store the program and the flash memory of Atmega168 is 16KB (of which 2KB is used for the Bootloader) and the flash memory of Atmega328 is 32KB.

Similarly, the EEPROM is 512KB and 1KB, and SRAM is 1KB and 2KB for Atmega168 and Atmega328 respectively.

The Nano board is almost similar to the UNO board with the former smaller in size with no DC power jack.

Arduino Nano Pinout:

The following figure shows the pinout diagram of the Arduino Nano board.



Arduino Nano Pin Description:

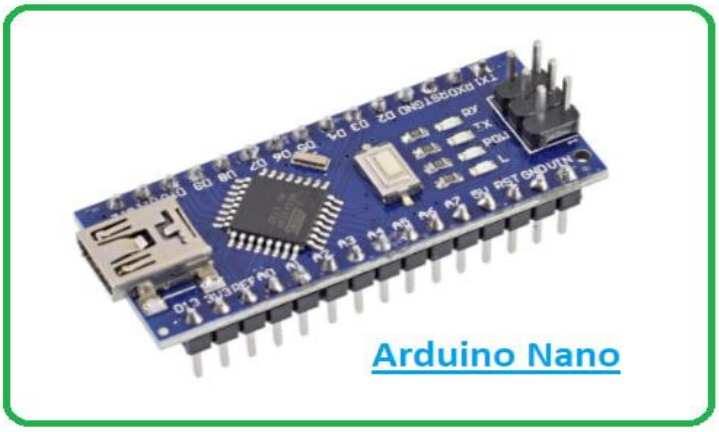
In this section, we'll cover the Arduino Nano Pinout, we will discuss pin description of each pin integrated on the board.

- **Digital Pins:** There are 14 digital pins on board which is used to connect external component.
- **Analog Pins:** 6 analog pins on board that is used to measure voltage in a range from 0 to 5V.
- **LED:** The unit comes with a built-in LED connected to pin 13 on the board.

- **VIN:** This is an input voltage to the Arduino board when using an external power source (6-12V).
- **3.3V:** It is a minimum voltage produced by the voltage regulator on the board.
- **5V:** Regulated power supply used to power up the controller and other components on board.
- **AREF:** It is an Analog Reference that is applied to the unit as a reference voltage from an external power supply.
- **GND:** Two ground pins are available on the board.
- **Reset:** Two reset pins are integrated on the board. These pins are used to reset the controller internally through software.
- **External Interrupts:** Pin 2 and 3 are used to trigger external interrupts. These pins are used in case of emergency.
- **USART:** The board supports USART serial communication that carries two pins i.e. Rx which is used for receiving the serial data and Tx which is a transmission pin used to transmit serial data.
- **I2C:** The unit comes with an I2C communication protocol where two pins SDA and SCL are used to support this communication. SDA is a serial data line that carries the data while SCL is a serial clock line used for data synchronization between the devices on the I2C bus. The Wire Library of Arduino Software can be accessed to use the I2C bus.
- **SPI:** The device also supports SPI (serial peripheral interface) communication protocol where four pins (SS, MISO, MOSI, SCK) are used for this communication. This protocol is used to transfer data between the microcontroller and other peripheral devices.

How to Program Arduino Nano?

All Arduino boards can be programmed using Arduino IDE (Integrated Development Environment) Software – An official software introduced by Arduino. cc. All you need is a code to burn into the board to make it work as per the instructions fed into the board.



Plus, the board features a built-in Bootloader which sets you free from getting an external burner to burn the Arduino program. The unit supports a USB interface with a mini USB port. The USB cable is used to connect the board with the computer.

Arduino Nano Applications:

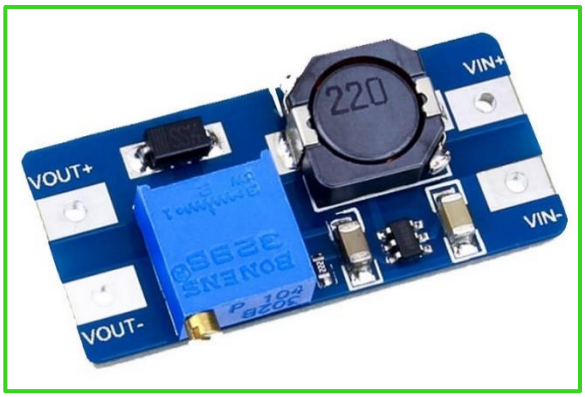
The best thing about Arduino boards is they can work as a stand-alone project or as a part of other electronic projects. You can interface Arduino Nano with other Arduino boards and Raspberry Pi boards. No technical expertise is required to use Arduino boards and anyone with little to no technical knowledge can make amazing projects with these units.

The following are the main applications of Arduino Nano Board.

- Medical Instruments
- GSM Based Projects
- Embedded Systems
- Arduino Metal Detector
- Industrial Automation
- Android Applications
- Virtual Reality Applications
- Real-Time Face Detection
- Automation and Robotics

4.2.2 MT3608 Step Up Module

The MT3608 power module is a step-up(Boost) converter module intended for small and low-power applications. The module has the capability to regulate the output voltage up to 28V and deliver an output current of a maximum of 2A. The module consists of MT3608 IC which comes in a 6-Pin SOT23-6 Package switches at 1.2Mhz which allows the use of tiny capacitors and inductors leading to a compact size power boost module.



MT3608 Step-Up Power Module

Features and Specifications of MT3608 Step-Up Power Module:

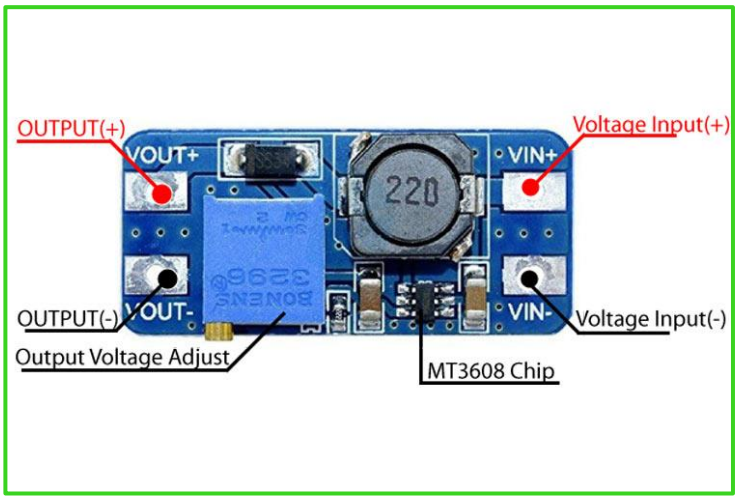
Below are some features and specifications of the MT3608 step-up power module:

1. Input Voltage: 2-24V DC
2. Output Voltage: 5-28V DC
3. Maximum Output Current: 2A
4. Switching Frequency: 1.2Mhz
5. Output Ripple: <100mV
6. Module Size: 37.2mm x 17.2mm x 14.0mm
7. About 93% Efficiency
8. Features like an under-voltage lockout, thermal overload protection

Pin Configuration of MT3608 Step-Up Power Module:

Pin configuration of the Step up module is given in the table below. The module has 4 pins, 2 of which are for voltage input and 2 are for regulated output voltage. The module also consists of a potentiometer which can be used to adjust the output voltage levels.

Pin Type	Description
VIN+	Voltage Input
VIN-	Ground
VOUT+	Voltage Output
VOUT-	Ground

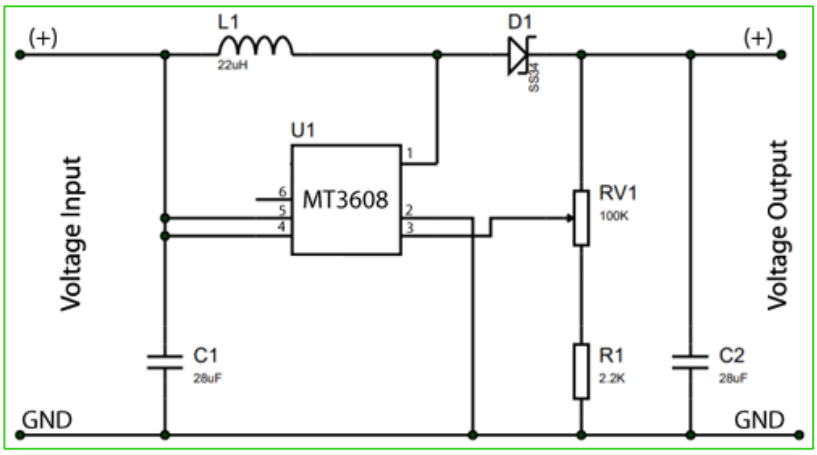


MT3608 Step-Up Power Module

Working Principle and Boosting Circuit of the MT3608 Step-Up Power Module

A boost converter(DC-DC step-up converter) is used to step up a lower voltage to a higher voltage level with quite a simple circuitry. It is a type of switch-mode power supply as it uses a switching device to regulate

the voltage. In our case, the switching module is the MT3608 IC which has a high switching frequency of 1.2MHz. The higher switching frequency also opens up the possibilities to use smaller indicators making the module compact and also providing high power output simultaneously.



The circuit above depicts how the connections are made for the operation of the MT3608 step-up module.

Working:

Whenever current is passed through the inductor(L1) it induces some magnetic field and when you change the current level passing through it, the magnetic field collapses and it generates a high voltage spike. Having a high switching frequency IC(MT3608) allows this principle to happen by generating and collapsing the magnetic field induced by the inductor. When the switching module is off, the voltage spike(high voltage level) passes through the Schottky diode(D1) and gets stored in the capacitor(C2) hence increasing the voltage of the capacitor and we obtain a higher voltage level output across the capacitor. The Schottky diode (D1) plays an important role in blocking a reverse current to the circuit.

Applications of MT3608 Step-Up Power Module:

Here are some of the applications of the MT3608 Step-Up Power Module:

1. Power amplifiers
2. Battery power systems
3. Consumer electronics
4. DC motor drivers

4.2.3 DFPlayer Mini MP3 Player

The DF Player Mini MP3 Player For Arduino is a small and low cost MP3 module with an simplified output directly to the speaker. The module can be used as a stand alone module with attached battery, speaker

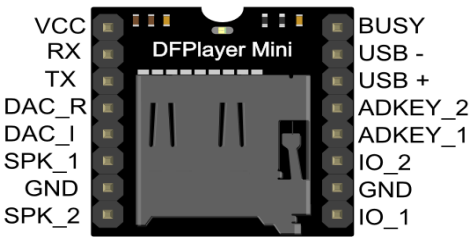
and push buttons or used in combination with microcontrollers such as Arduino, ESP32, Raspberry Pi and any microcontrollers with Uart.

Specification of DF Player Mini MP3 Player:

- Sampling rates (kHz): 8/11.025/12/16/22.05/24/32/44.1/48
- 24 -bit DAC output, support for dynamic range 90dB , SNR support 85dB
- Fully supports FAT16 , FAT32 file system, maximum support 32G of the TF card, support 32G of U disk, 64M bytes NORFLASH
- A variety of control modes, I/O control mode, serial mode, AD button control mode
- advertising sound waiting function, the music can be suspended. when advertising is over in the music continue to play
- audio data sorted by folder, supports up to 100 folders, every folder can hold up to 255 songs
- 30 level adjustable volume, 6 -level EQ adjustable.

DF Player Mini MP3 Player pinout:

Pin	Description	Note
VCC	Input Voltage	DC3.2~5.0V;Type: DC4.2V
RX	UART serial input	
TX	UART serial output	
DAC_R	Audio output right channel	Drive earphone and amplifier
DAC_L	Audio output left channel	Drive earphone and amplifier
SPK2	Speaker-	Drive speaker less than 3W
GND	Ground	Power GND
SPK1	Speaker+	Drive speaker less than 3W
IO1	Trigger port 1	Short press to play previous（long press to decrease volume）
GND	Ground	Power GND
IO2	Trigger port 2	Short press to play next（long press to increase volume）
ADKEY1	AD Port 1	Trigger play first segment
ADKEY2	AD Port 2	Trigger play fifth segment
USB+	USB+ DP	USB Port
USB-	USB- DM	USB Port
BUSY	Playing Status	Low means playing \High means no



DFPlayer Mini MP3 Player

Work Mode of DF Player Mini MP3 Player:

1. Serial Mode

Support for asynchronous serial communication mode via PC serial sending commands Communication Standard:9600 bps Data bits :1 Checkout :none Flow Control :none

- Instruction Description

Format: $S VER Len CMD Feedback para1 para2 checksum $O		
$S	Start bit 0x7E	Each command feedback begin with $, that is 0x7E
VER	Version	Version Information
Len	the number of bytes after "Len"	Checksums are not counted
CMD	Commands	Indicate the specific operations, such as play / pause, etc.
Feedback	Command feedback	If need for feedback, 1: feedback, 0: no feedback
para1	Parameter 1	Query high data byte
para2	Parameter 2	Query low data byte
checksum	Checksum	Accumulation and verification [not include start bit $]
$O	End bit	End bit 0xEF

For example, if we specify play NORFLASH, you need to send: 7E FF 06 09 00 00 04 FE EE EF
Data length is 6, which are 6 bytes [FF 06 09 00 00 04]. Not counting the start, end, and verification.

- Serial Control Cmd

CMD	Function Description	Parameters(16 bit)
0x01	Next	
0x02	Previous	
0x03	Specify tracking(NUM)	0-2999
0x04	Increase volume	
0x05	Decrease volume	
0x06	Specify volume	0-30
0x07	Specify EQ(0/1/2/3/4/5)	Normal/Pop/Rock/Jazz/Classic/Base
0x08	Specify playback mode (0/1/2/3)	Repeat/folder repeat/single repeat/ random
0x09	Specify playback source(0/1/2/3/4)	U/TF/AUX/SLEEP/FLASH
0x0A	Enter into standby – low power loss	
0x0B	Normal working	
0x0C	Reset module	
0x0D	Playback	
0x0E	Pause	
0x0F	Specify folder to playback	1~10(need to set by user)
0x10	Volume adjust set	{DH= 1:Open volume adjust }{DL: set volume gain 0~31}
0x11	Repeat play	{1:start repeat play}{0:stop play}

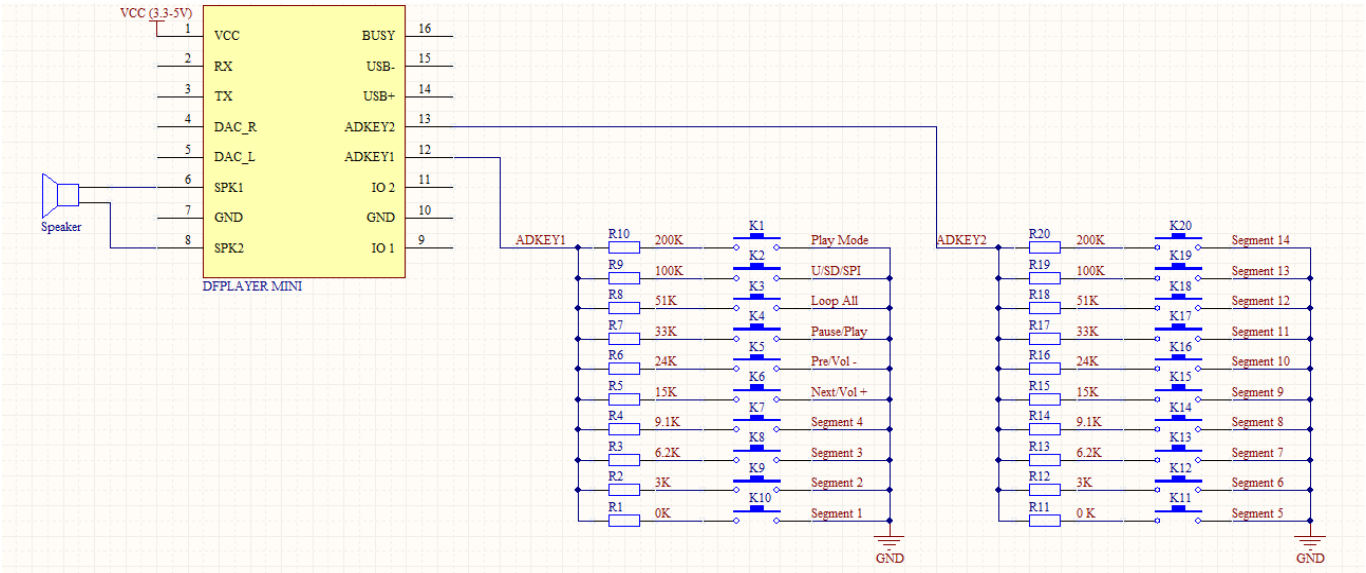
- Serial Query CMD

Commands	Function Description	Parameters(16 bit)
0x3C	STAY	
0x3D	STAY	
0x3E	STAY	
0x3F	Send initialization parameters	0 - 0x0F(each bit represent one device of the low-four bits)
0x40	Returns an error, request retransmission	
0x41	Reply	
0x42	Query the current status	
0x43	Query the current volume	
0x44	Query the current EQ	
0x45	Query the current playback mode	
0x46	Query the current software version	
0x47	Query the total number of TF card files	
0x48	Query the total number of U-disk files	
0x49	Query the total number of flash files	
0x4A	Keep on	
0x4B	Queries the current track of TF card	
0x4C	Queries the current track of U-Disk	
0x4D	Queries the current track of Flash	

2. AD KEY Mode

We use the AD module keys, instead of the traditional method of matrix keyboard connection, it is to take advantage of increasingly powerful MCU AD functionality, Our module default configuration 2 AD port, 20 key resistance distribution.

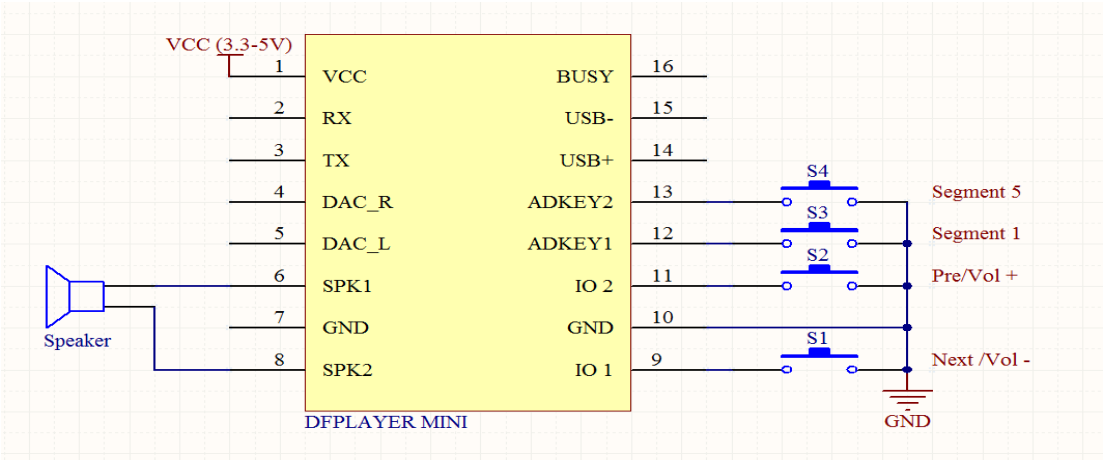
Get AD Button from DFRobot Store or DFRobot Distributor.



3. I/O Mode

Here comes the most simple way to use this module.

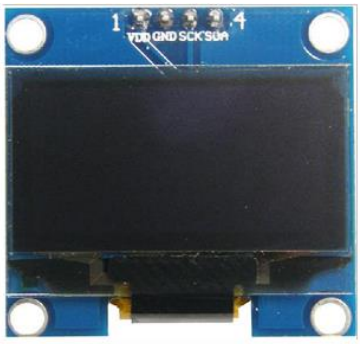
- Refer diagram

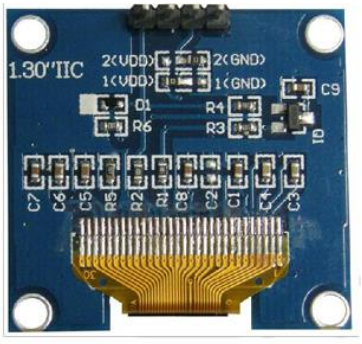


Note: short time press means pre/next and long time press means vol- ,vol +

4.2.4 1.3 inch OLED Display

In contrast to LCD technology, Organic Light-Emitting Diode (OLED) displays do not require a backlight and are regarded as the ultimate technology for the next generation of flat-panel displays. OLED displays are composed of a thin, multi-layered organic film placed between an anode and cathode, which are made up of electric conductive transparent Indium Tin Oxide. The multi-layered organic film includes a Hole Transporting Layer, Emission Layer and Electron Transporting Layer. By applying an appropriate electrical voltage, the holes and electrons are injected into the Emission Layer from the anode and cathode respectively and combine to form excitons, after which electroluminescence occurs. This 1.3” 128*64 White OLED Module offers 128*64-pixel resolution. They are featuring much less thickness than LCD Displays with good brightness and produce better and true colors. This OLED Display Module is very compact and will add a great ever user interface experience to your Arduino project. The connection of this display with Arduino is made through the I2C (also called as IIC) serial interface.





Features of OLED Display:

- Supply voltage: 3.3V-5V
- Pixel: 128*64
- Display size- 1.3 inch
- Operating temperature range: -40°C - +80°C
- Use I2C Interface
- Chip No: SSD1306
- Color: White
- Super high contrast and brightness(adjustable)
- PCB Size: 33.7 mm x 35.5 mm
- Low power consumption
- High contrast, thus supporting clear display with no need of backlight
- For OLED SSD1306, a more elaborate and beautiful screen than LCD with more functions.

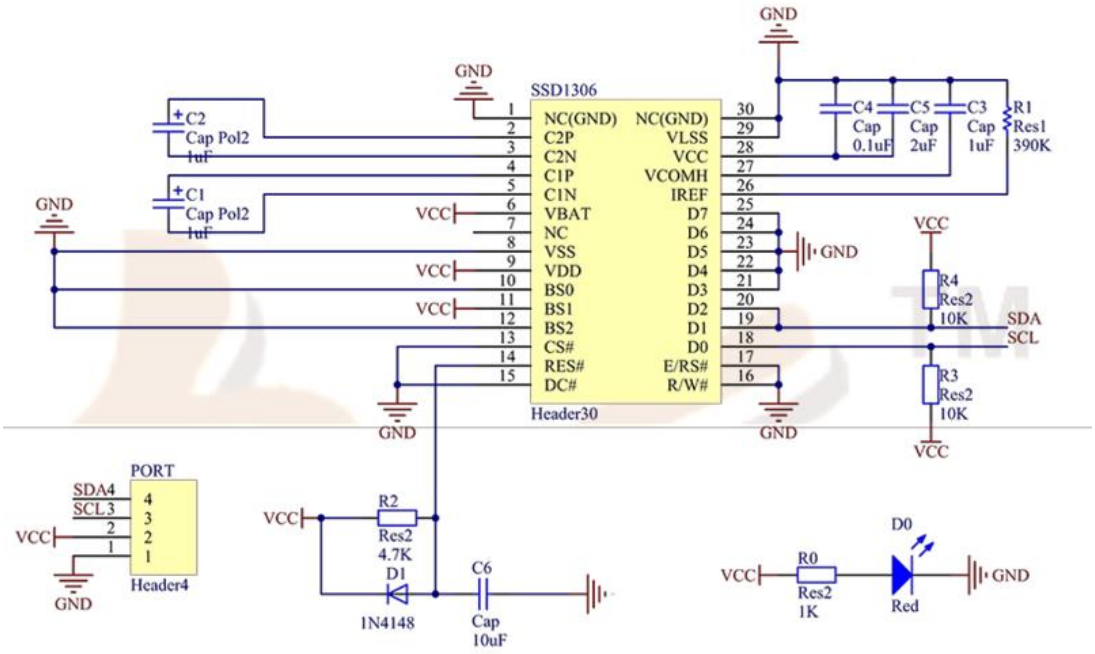
Pin description:

Pin No.	Pin Name	Description
1.	Supply Voltage (Vcc, 5V)	Can be powered by either 3.3V or 5V
2.	Ground (GND)	Pin Ground
3.	Serial Clock(SCL)	Pin SCL of I2C interface
4.	Serial Data(SDA)	Pin SDA of I2C interface

Electrical characteristics:

ITEM	SYMBOL	TEST CONDITION	TYP	UNIT
Operating voltage	VDD	Ta=25°C	5.0	V
Operating voltage for LCD	VLCD	Ta=25°C	5.0	V
Supply current	IDD	Ta=25°C, VDD=5.0V	2.0	MA
Supply current for Back light	IF	Ta=25°C, VF=5V	20	MA

Schematic diagram:



4.2.5 DC Voltmeter

What Is a Voltmeter?

A voltmeter, also known as a voltage meter, is an instrument that measures the voltage or potential difference between two points of an electronic or electrical circuit. Usually, the voltmeter is used for Alternating Current (AC) circuits or Direct Current (DC) circuits. Alternatively, Radio Frequency (RF) voltage can also be measured by specialised voltmeters.

A voltmeter measures voltages usually calibrated in volts, millivolts (0.001 volt), or kilovolts (1,000 volts). In order to measure a device's voltage, a voltmeter is connected in parallel to a device. This setup is important as objects in parallel usually tend to experience the same potential difference. It is connected in parallel with the circuit, mainly because the same voltage drop occurs across it.

A voltmeter also has high internal resistance. This is done mainly because it is used in measuring the potential difference between the two points of the circuit. As such, the current of the measuring device remains the same. In other words, the high resistance of the voltmeter will impede the flow of current through it. This allows the device to take correct readings of the voltage.

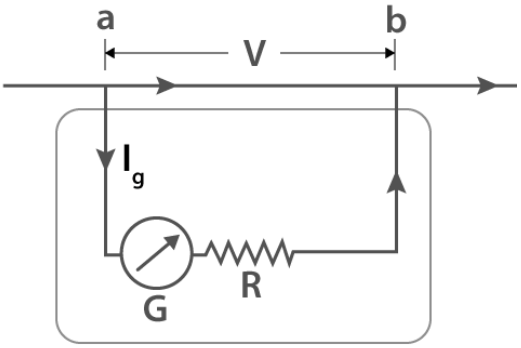
Voltmeter Symbol

The voltmeter is usually represented by the letter V, which is placed inside a circle adjoining two terminals.

Types of Voltmeter:

- **Analogue or Analog Voltmeter**

An analogue voltmeter is used mainly for measuring the AC voltage. The reading is displayed with the help of a pointer that is fixed on the calibrated scale. The movement of the pointer is affected by the torque that is acting on it. The magnitude of the torque that is developed is directly proportional to the voltage being measured.



Voltmeter

A galvanometer (current meter) that is sensitive and is part of a high resistance series is what makes a basic analogue voltmeter. The meter should have high internal resistance, or else the circuit operation during the test would be interrupted by drawing current significantly. The voltage range displayed by the meter is determined by the series resistance value and the galvanometer sensitivity.

On the other hand, to measure low voltages, an oscilloscope is often used where instantaneous voltage is depicted by the vertical displacement. RF and AC applications have their peak-to-peak and peak voltage measured by the oscilloscopes. Wiring, insulators and heavy-duty probes are crucial for making the meters for measuring high potential differences.

- **Digital Voltmeter**

Another voltmeter that is quite often used among voltage measurement instruments is the digital voltmeter. A digital voltmeter (DVM) measures an unknown input voltage by converting the voltage to a digital value, and then displays the voltage in numeric form. DVMs are usually designed around a special type of analogue-to-digital converter called an integrating converter.

There are different factors that have an impact on the accuracy of the DVM, like input impedance, temperature, and power supply voltage variations of the DVM. Around 10 MΩ is the input resistance of DVMs that are the least expensive. The input resistances for precision DVMs of 1 GΩ or higher for ranges of low voltages (below 20 V). The DVM must be periodically calibrated with a voltage standard, like the Weston Cell, as a way to ensure the manufacturer's specified tolerances.

- **Other Types of Voltmeter**

These are voltmeters based on their construction.

- **MI Voltmeter:** Moving Iron (MI) voltmeter is a device that is used for measuring both AC and DC voltages. In this device, the deflection is directly proportional to the voltage of the coil. It is further

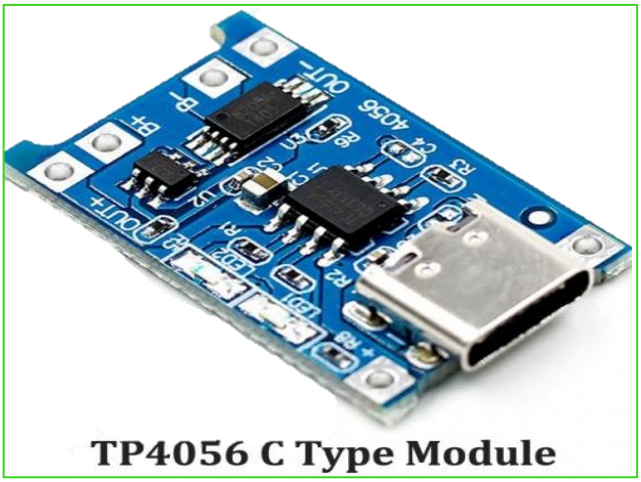
divided into two types – Attraction Type Moving Iron Instruments and Repulsion Type Moving Iron Instruments.

- **Rectifier Voltmeter:** These are widely used in AC circuits for measuring voltage. This voltmeter converts the AC into DC with the help of a rectifier. The converted DC signal is then measured using the PMMC instrument.
- **PMMC Voltmeter:** A Permanent Magnet Moving Coil (PMMC) voltmeter, also known as a D'Arsonval meter or simply galvanometer, measures the current in a coil by observing the coil's angular deflection in a uniform magnetic field. The current is induced in the PMMC instrument due to the measure and voltage, and deflection of the pointer occurs. The PMMC voltmeter is used for DC measurement.
- **Electro-dynamometer Voltmeter:** This voltmeter is used to measure the voltage of both AC and DC circuits. The calibration is usually kept the same for both the AC and DC measurement.
- **Amplified Voltmeter:** These are voltmeters whose sensitivity and input resistance can be increased or decreased. This can be done if the current required to deflect the meter pointer is supplied by an amplifier and power supply.

4.2.6 TP4056 Type – C charging module

This is a TP4056 1A Li-Ion Battery Charging Board Type C with Current Protection is a tiny module, perfect for charging single cell 3.7 Volts 1 Ah or higher lithium-ion (Li-Ion) cells such as 16550s that don't have their own protection circuit. Based on the TP4056 charger IC and DW01 battery protection IC this TP4056 li-ion battery charging module will offer 1 Ampere charge current then cut off when finished.

Furthermore, when the battery voltage drops below 2.4 Volts the protection IC will switch the load off to protect the cell from running at too low of a voltage – and also protects against over-voltage and reverse polarity connection (it will usually destroy itself instead of the battery) however please check you have it connected correctly the first time.



TP4056 C Type Module

Using TP4056 Type C Battery Charging USB Module:

- Connect Type C cable for power, or 5V DC to pads marked IN+ and IN- on the left-hand side of the module
- Connect cell to charge to B+/B- pads on the right-hand side of the module.

- A load (something for the battery to power) can be connected to the OUT+/OUT- pads on the right-hand side
- Important! Disconnect load when charging
- The red LED indicates charging in progress, the green LED indicates charging has finished.
- Never charge your battery at a rate greater than 1C.

Features of TP4056 Type C Battery Charging USB Module

- Led indicator : Red is charging Green is fully charged.
- Current Protection : Yes
- Inversed polarity : NO.
- Use mature charging chip TP4056 for simple peripheral circuits, good protection performance, and high charging accuracy.
- Fully machinery automated processing, all patch parts manufacturing.

4.2.7 Mini Lithium Polymer Battery

A **lithium polymer battery**, or more correctly **lithium-ion polymer battery** (abbreviated as **LiPo**, **LIP**, **Li-poly**, **lithium-poly** and others), is a rechageable battery of lithium-ion technology using a polymer electrolyte instead of a liquid electrolyte. High conductivity semisolid (gel) polymers form this electrolyte. These batteries provide higher specific energy than other lithium battery types and are used in applications where weight is a critical feature, such as mobile devices, radio-controlled aircraft and some electric vehicles.

History:

Batteries are the first practical way of generating electricity and were invented by Alessandro Volta. Before generators came into the scene, batteries were the main source of electricity till the end of the The ancient electric cars also used the semi-sealed wet cells. This background of the batteries was the key for the development of the LiPo cells. The discovery of the Lithium Polymer Battery cells came because of the Lithium-ion and lithium-metal cells as they went to depth in the 1980s. A significant, yet remarkable milestone was the first commercial Li-ion cell of Sony in 1991. There was a revolution thereafter which introduced a pouch form of battery called “LiPo”.

Design origin and terminology:

Lithium polymer cells have evolved from lithium-ion and lithium-metal batteries. The primary difference is that instead of using a liquid lithium-salt electrolyte (such as LiPF6) held in an organic solvent such as EC/DMC/DEC, the battery uses a solid polymer electrolyte (SPE) such as Polyethylene oxide (PEO), Poly(acrylonitrile) (PAN), poly(methl methacrylate) (PMMA) or poly(vinylidene fluoride) (PVdF).

In the 1970s the original polymer design used a solid dry polymer electrolyte resembling a plastic-like film, replacing the traditional porous separator that is soaked with electrolyte.

The solid electrolyte can typically be classified as one of three types: dry SPE, gelled SPE and porous SPE. The dry SPE was the first used in prototype batteries, around 1978.

A typical cell has four main components: positive electrode, negative electrode, separator and electrode. The separator itself may be a polymer, such as a microporous film of polyethylene (PE) or polypropylene (PP); thus, even when the cell has a liquid electrolyte, it will still contain a "polymer" component. In addition to this, the positive electrode can be further divided into three parts: the lithium-transition-metal-oxide (such as LiCoO2 or LiMn2O4), a conductive additive, and a polymer binder of poly(vinylidene fluoride) (PVdF). The negative electrode material may have the same three parts, only with carbon replacing the lithium-metal-oxide. The main difference between lithium ion polymer cells and lithium ion cells is the physical phase of the electrolyte, such that LiPo cells use dry solid, gel-like electrolytes whereas Li-ion cells use liquid electrolytes.

A Lithium-ion battery, also known as the Li-ion battery, is a type of secondary (rechargeable) battery composed of cells in which lithium ions move from the anode through an electrolyte to the cathode during discharge and back when charging.

A lithium-ion polymer (LiPo) battery (also known as Li-pol, lithium-poly, and other names) is a type of Li-ion battery with a **polymer electrolyte instead of a liquid electrolyte**. All LiPo batteries use a high-conductivity gel polymer as the electrolyte. Lithium polymer cells have evolved from lithium-ion and lithium-metal batteries. The primary difference between lithium-ion and Li-pol is that instead of using a liquid lithium-salt electrolyte (such as LiPF6) held in an organic solvent, the battery uses a solid polymer electrolyte (SPE) such as polyethylene oxide (PEO), polyacrylonitrile (PAN), polymethyl methacrylate (PMMA) or polyvinylidene fluoride (PVdF). LiPos provide higher specific energies than other lithium batteries, often used in systems where weight is an important factor, such as mobile devices, drones, and some electric vehicles.

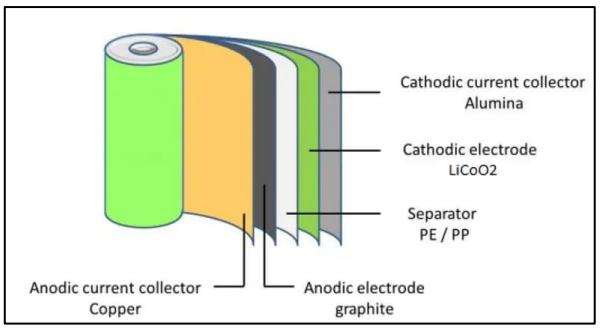
The cathode is made of a composite material (an intercalated lithium compound) and defines the name of the Li-ion battery cell. The anode is usually made out of porous lithiated graphite. The electrolyte can be liquid, polymer, or solid. The separator is porous to enable the transport of lithium ions and prevents the cell from short-circuiting and thermal runaway.

Chemistry, performance, cost, and safety characteristics vary across types of lithium-ion batteries. Handheld electronics mostly use lithium polymer batteries (with a polymer gel as electrolyte), a lithium cobalt oxide (LiCoO2) cathode material, and a graphite anode, which offer high energy density.

Li-ion batteries, in general, have a high energy density, no memory effect, and low self-discharge. One of the most common types of cells is 18650 battery, which is used in many laptop computer batteries, cordless power tools, certain electric cars, electric kick scooters, most e-bikes, portable power banks, and LED flashlights. The nominal voltage is 3.7 V.

Note that non-rechargeable primary lithium batteries (like lithium button cells CR2032 3V) must be distinguished from secondary lithium-ion or lithium-polymer, which are rechargeable batteries. Primary lithium batteries contain metallic lithium, which lithium-ion batteries do not.

Composition of Lithium Polymer Battery:



A typical lithium-ion cell contains:

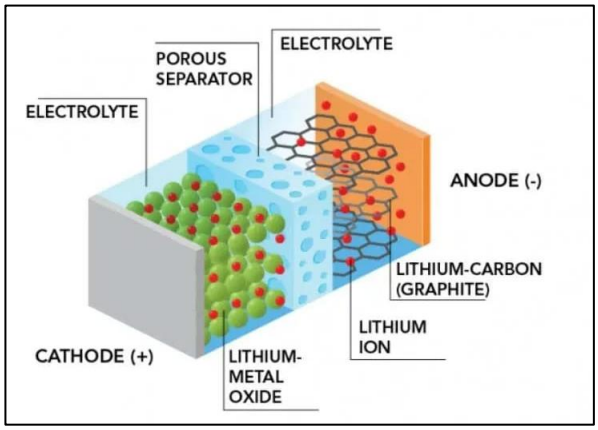
- **Cathode:** The cathode is the positive or oxidizing electrode that acquires electrons from the external circuit and is reduced during the electrochemical reaction. In the case of lithium batteries, cathode materials are generally constructed from LiCoO2 or LiMn2O4. For the cathode, it is important to hold a large amount of lithium without significant change in structure, have good chemical and electrochemical stability with electrolyte, be a good electrical conductor and diffuser of lithium ions, and be of low cost.

- **Anode**: The anode is the negative or reducing electrode that releases electrons to the external circuit and oxidizes during an electrochemical reaction. One of the most common anode materials used today is lithiated graphite, LixC6, which is composed of graphite sheets intercalated with lithium. New materials, such as those based on Silicon and other elemental blends, are being researched. Lithiated graphite has a unit cell with an HCP structure.

- **Separator**. A separator is a permeable membrane placed between a battery's anode and cathode. The main function of a separator is to keep the two electrodes apart to prevent electrical short circuits while also allowing the transport of ionic charge carriers that are needed to close the circuit during the passage of current in an electrochemical cell. Commercially available liquid electrolyte cells use microporous polyolefin materials, such as polyethylene (PE) or polypropylene (PP). Separators in Li-ions have to be electrochemically and chemically stable relative to the electrolyte and electrode materials. Functional separators that use MOF-coated membranes to perform the dual functions of the electrolyte and separator are being developed to support the design of high-performance Li-metal batteries for high-energy systems in electric vehicles and electric aircraft.

- **Electrolyte**: The choice of electrolyte in all batteries is critical for performance as well as safety. Most of the electrolytes used in commercial lithium-ion batteries are non-aqueous solutions, in which lithium hexafluorophosphate (LiPF6) salt dissolved in organic carbonates, in particular, mixtures of ethylene carbonate (EC) with dimethyl carbonate (DMC), propylene carbonate (PC), diethyl carbonate (DEC), and/or ethyl methyl carbonate (EMC). A good electrolyte must have low reactivity with other cell components, high ionic conductivity, low toxicity, a large window of electrochemical voltage stability (0-5V), and be thermally stable.

Internal Structure of a Lithium Polymer Battery:



Lithium Polymer Battery is a combination of a cylindrical and a rectangular shaped structure. The internal structure is bounded spirally that helps in creating a partition between the anode and the cathode portions of the battery by putting a concise and highly porous polyethylene layer between the two.

Lithium Polymer batteries work on the principle of using liquid electrolyte solution, so a portion of the battery is filled with organic liquid electrolyte solution. Also, to protect the battery from unprecedented incidents such as short circuit or explosions, safety valves, and PTC components are installed.

Followed by these, cells are installed either in a series or parallel to one another. The number of cells varies depending on the total voltage requirement of the battery. The voltage of one cell is 3.6V and the total voltage of the battery is the sum of the voltage of the cells installed.

The working principle of polymer lithium battery:

There are two types of lithium ion batteries: liquid lithium ion batteries and lithium polymer batteries. Among them, the liquid lithium ion battery refers to a secondary battery with Li+ intercalation compound as the positive and negative electrodes. The positive electrode adopts lithium compound LiCoO2, LiNiO2 or LiMn2O4, and the negative electrode adopts lithium carbon intercalation compound LixC6.

The principle of lithium polymer battery is the same as that of liquid lithium, but the main difference is that the electrolyte is different from liquid lithium. The main structure of the battery includes three elements:

- Positive electrode,
- Negative electrode
- Electrolyte

The so-called lithium polymer battery refers to the use of polymer materials as the main battery system in at least one or more of these three main structures. In the lithium polymer battery system developed, polymer materials are mainly used for the positive electrode and electrolyte.

Cathode materials include conductive polymers or inorganic compounds commonly used in lithium-ion batteries. The electrolyte may use a solid or gel polymer electrolyte or an organic electrolyte. Generally, lithium ion technology uses liquid or gel electrolyte, so it needs high strength. The secondary packaging contains flammable active ingredients, which increase weight and limit size flexibility.

The shape of the new generation of lithium polymer battery is theoretically achievable, and the shape is diversified, which improves the flexibility of battery shape design, so that it can meet product needs, and make some shape and capacity batteries for application equipment developers. Provide a high degree of design flexibility and adaptability in the power supply solution to maximize its product performance. At the same time, the unit energy of lithium polymer batteries is 10% higher than that of general lithium ion batteries. Compared with lithium-ion batteries, its capacity and cycle life have been greatly improved.

Specification:

- **Voltage level:** 3.6 to 3.7 V (average voltage at 50% discharge depth/0.2 C).
- **Charging:** Constant I / constant V, maximum charging voltage 4.2 V, for special cells up to 4.35/4.4 V, max. charging current 1 C, for larger cells 0.5 C.
- **Discharge**: Min. voltage 3.0 V, currents up to 1 C (in some cases 2 C).
- **Temperature range**: Charge: 0°C to +45°C, with reduced currents below 15°C.
- **Discharge:** -20°C to +60°C with suitably reduced voltage levels and capacities at low temperatures.
- **Cycles:** Charge/discharge at 0.5C/0.5C, 80% residual capacity after 500 cycles.
- **The energy density of LiPo batteries** ranges from 140 - 200+ Wh/kg in terms of weight and 250 - 350+ Wh/L for volume.

Differences between Li-Ion and Li-Polymer (LiPo) Battery:

Based on the composition of ions that carry the electrolyte materials, Lithium Batteries can be classified as Lithium Ion and Lithium Polymer. Following are the points of difference between the two:

- Li-ion batteries use liquid electrolytes whereas LiPo batteries use solid gel like polymers that substitute the electrolytes.
- Li-ion batteries are relatively cheaper because of the higher energy density.
- LiPo Batteries are safer and light weight.

- LiPo Batteries are rechargeable and have a longer span of life once charged as compared to the Li-ion batteries which tend to lose charge even when not in use.

The advantages of lithium polymer battery:

1. Safety performance is good

Lithium-ion polymer battery with aluminum composite flexible packaging in the structure, different from liquid metal case of batteries, in the event of a safety hazard, liquid batteries easy explosion, but lithium ion polymer battery most has meteorism.

2. More thin thickness, and can do more thin

Ultra-thin, batteries could assemble into a credit card. Ordinary liquid lithium battery use custom shell, after the plug is the method of the cathode, thickness of 3.6 mm the following technical bottlenecks, polymer batteries, there is no this problem, the thickness can be below 1 mm, can meet the demand of current mobile direction.

3. Light-weight

Lithium polymer battery using polymer electrolyte without metal shell as a protective outer packing. Same specifications of steel shell polymer battery weight is equal capacity lithium electric light 40%, a 20% aluminum battery light.

4. Big capacity

Lithium Polymer battery is the same size of the steel shell battery, the capacity is 10 to 15% more than high aluminum battery 5 ~ 10%, be the first choice of the color screen mobile phones and MMS, now on the market a new color and MMS phones are mostly used lithium polymer batteries

5. Small internal impedance

The internal resistance of the polymer batteries is less than liquid batteries, the current domestic internal resistance of the polymer batteries can even do below 35 m Ω, greatly reduced the battery power consumption, extend the standby time of mobile phone, can reach with international level. The support of large discharge current polymer li-ion battery, the ideal choice of remote control model to become the most promising alternative products of nimh batteries.

6. Shape can be customized

Manufacturers no need to limited to the standard shape, able to make the appropriate economic size. Lithium Polymer battery can increase or decrease the thickness of the batteries according to the customer's demand, the development of new batteries models, price cheap, open mold cycle is short, some even can be tailored according to the mobile phone shape, to make full use of the battery shell space, improve battery capacity

7. Better discharge characteristic

Li-Polymer battery using gel electrolyte, compared to liquid electrolyte, colloid electrolyte with smooth discharge characteristics and higher discharge platform.

8. Simple PCM design

Because of using polymer materials, which made batteries no fire, no explosion, batteries itself has enough safety, therefore, polymer battery protection circuit design can consider omitting PTC and fuse, thereby saving the cost of batter

Applications of lithium polymer battery:

LiPo cells provide manufacturers with compelling advantages. They can easily produce batteries of almost any desired shape. For example, the space and weight requirements of mobile devices and notebook computers can be met. They also have a low self-discharge rate, which is about 5% per month.

- **Drones, radio controlled equipment and aircraft**

  LiPo batteries are now almost ubiquitous when used to power commercial and hobby drones (unmanned aerial vehicles), radio-controlled aircraft, radio-controlled cars and large scale model trains, where the advantages of lower weight and increased capacity and power delivery justify the price. Test reports warn of the risk of fire when the batteries are not used in accordance with the instructions.
  The voltage for long-time storage of LiPo battery used in the R/C model should be 3.6~3.9V range per cell, otherwise it may cause damage to the battery.
  LiPo packs also see widespread use in airsoft, where their higher discharge currents and better energy density has very noticeable performance gain (higher rate of fire).

- **Personal electronics**

  LiPo batteries are pervasive in mobile devices, power banks, very thin laptops computers, portable media players,, wireless controllers for video game consoles, wireless PC peripherals, electric cigarettes, and other applications where small form factors are sought and the high energy density outweighs cost considerations.

- **Electric vehicles**

  Hyundai uses this type of battery in some of its battery electric and hybrid vehicles, as well as Kia Motors in their battery electric Kia Soul. The Bolloré Bluecar, which is used in car sharing schemes in several cities, also uses this type of battery.

- **Uninterruptible power supply systems**

  Lithium-ion batteries are becoming increasingly more commonplace in Uninterruptible power supply (UPS) systems. They offer numerous benefits over the traditional VRA battery and with stability and safety improvements confidence in the technology is growing. Their power to size and weight ratio is seen as a major benefit in many industries requiring critical power back up including data centers where space is often at a premium. The longer cycle life, usable energy (Depth of discharge), and thermal runaway are also seen as a benefit for using Li-po batteries over VRLA batteries.

- **Jump starter**

  The battery used to start a vehicle engine is typically 12V or 24V, so a portable jump starter or battery booster uses three or six LiPo batteries in series (3S1P/6S1P) to start the vehicle in an emergency, instead of the other jump-start methods . The price of a lead-acid jump starter is less but they are bigger and heavier than comparable lithium batteries, and so such products have mostly switched to LiPo batteries or sometimes lithium iron phosphate batteries.

4.2.8 Push Button Micro switch

What Are Pushbutton Switches?

It’s a type of electrical switch that’s used to control an electrical circuit by physically pressing a button. A Pushbutton switch, also known as a push switch, is a type of mechanical switch used to control the flow of electric current in an electronic circuit.

It is typically used for momentary, on-demand operations, where the switch is pressed to make or break the electrical connection, and it returns to its original position when released.

How Do They Function?

When you press the button, it makes or breaks an electrical connection, allowing or interrupting the flow of electricity through the circuit. Pushbutton switches are commonly used in various electronic devices, control panels and applications where a user needs to activate or deactivate a function or operation.

Pushbutton switches come in various shapes and sizes, but they typically consist of a housing or enclosure and a button or actuator that is designed to be pressed. The actuator can be made of various materials, such as plastic or metal, and it's connected to the switch mechanism inside the housing.

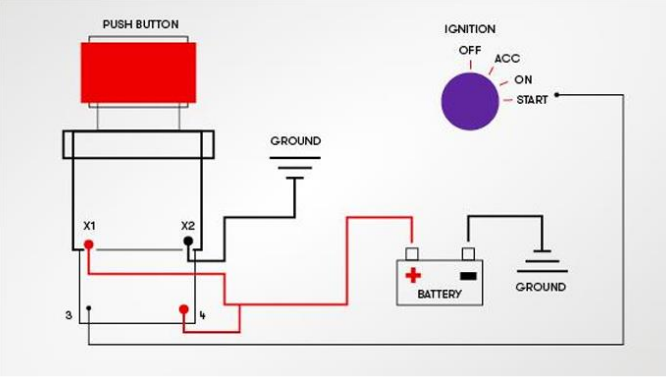
They can be either momentary or latching.

Momentary: In a momentary Pushbutton switch, the circuit is only connected (or disconnected) while the button is being actively pressed. As soon as you release the button, the connection is broken.

Latching: Latching Pushbutton switches have two stable states, and they stay in the position you push them into until you press them again to change the state. This means they can either maintain an open circuit or a closed circuit until manually switched to the other state.

Push Button Switch Diagram:

The diagram to the right demonstrates how a push button can be used alongside an ignition switch as part of a quick start ignition circuit. This schematic shows the basic circuit design and wire layout required for this particular application.



What's the Difference Between Pushbutton and Tactile Switches?

Tactile switches and Pushbutton switches are both types of mechanical switches used for controlling electrical circuits, but they differ in terms of their design, functionality and tactile feedback.

Tactile switches are designed to provide feedback to the user when the switch actuates. When you press a tactile switch, you can feel a noticeable physical "click" or resistance, which gives you confirmation that the switch has been activated. This tactile feedback is often preferred in applications where precise control is important as it helps prevent accidental presses. Pushbutton switches can vary in terms of tactile feedback. Some push switches may provide a tactile click similar to tactile switches, but not all of them do. There are Pushbutton switches that offer a smoother and quieter actuation without a distinct tactile feel.

Tactile switches are typically momentary, meaning the electrical connection is established only while the switch is actively pressed. When you release the button, the connection is broken. Pushbutton switches can be either momentary or latching. Momentary push switches are similar to tact switches in that they are pressed to make and released to break the electrical connection. Latching Pushbutton switches, on the other hand, can maintain their state (either open or closed) until manually switched to the other state.

Applications of Push Button Micro switch:

- Telecommunications
- Networking
- Computers/servers
- Performance audio
- Instrumentation
- Low power on/off designs
- External hard drives and modems
- Consumer electronics
- Medical devices

4.2.9.1k 1/4W resistor

Resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. Carbon film resistors are a fixed form type resistor. They are constructed out of a ceramic carrier with a thin pure carbon film around it, that functions as restive material. Resistors act to reduce current flow, and, at the same time, act to lower voltage levels within circuits. In electronic circuits, resistors are used to limit current flow, to adjust signal levels bias active elements, and terminate transmission lines among other uses.

Features:

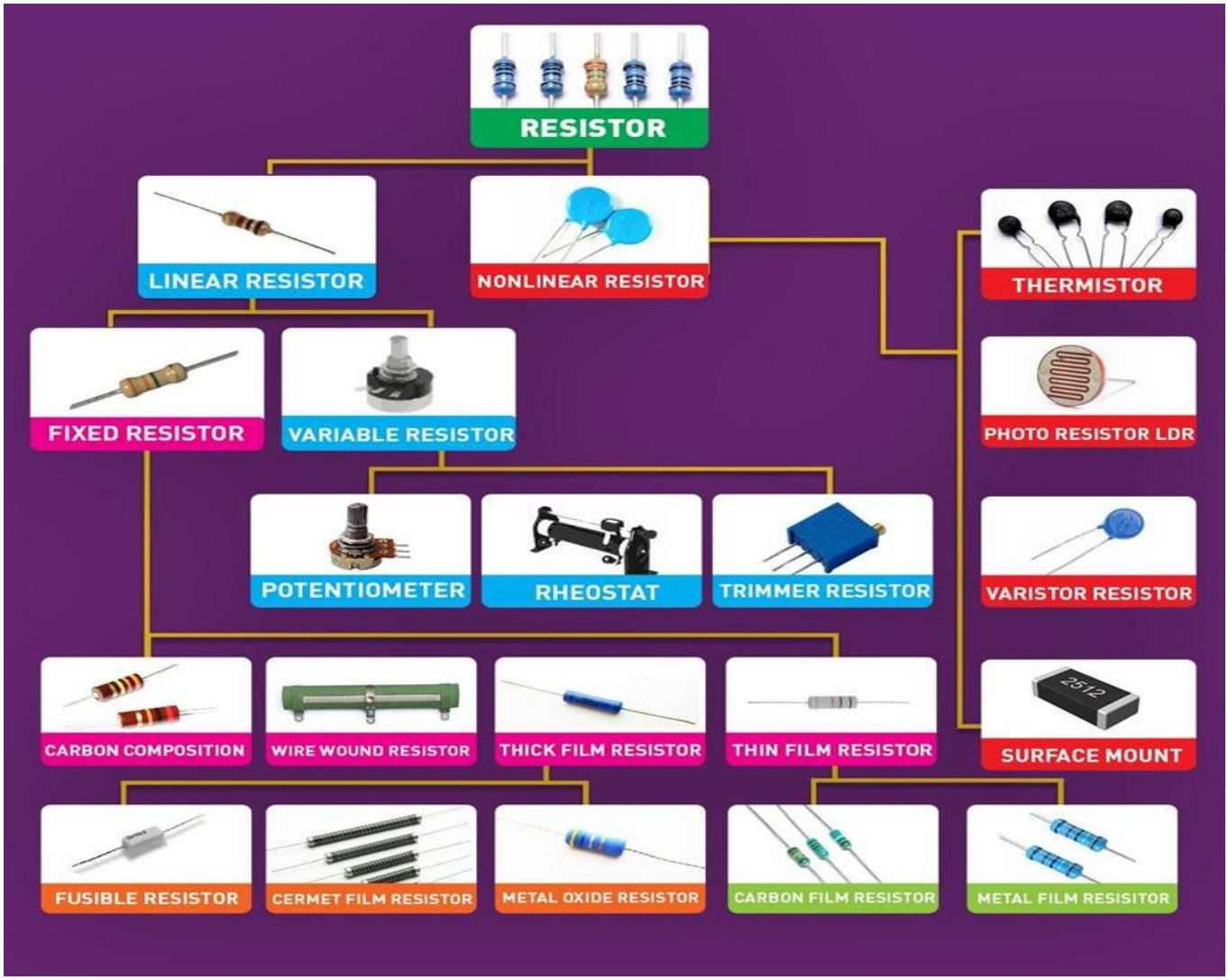
- Product Type: Metal Film Resistor
- Resistance Value: 1K Ohm
- Power Rating: 250 mW (1/4 W)
- Tolerance: ±1%
- Maximum Operating Temperature: +155°C
- Number of Terminations: 2
- Package / Case: Axial

Types of Resistors

Resistors are available in different shapes and sizes. Common types that are available are through-hole and surface mount. A resistor might be static, standard resistor, special, or a pack of variable resistors.

There are two basic types of resistors as follows:

- Linear resistor
- Non-linear resistor
- Linear resistors



Linear resistors: The resistors whose values change with change in applied temperature and voltage are known as linear resistors. There are two types of linear resistors:
Fixed resistors: These resistors have a specific value and these values cannot be changed. Following are the different types of fixed resistors:

- Carbon composition resistors
- Wire wound resistors
- Thin film resistors
- Thick film resistors

Variable resistors: These resistors do not have a specific value and the values can be changed with the help of dial, knob, and screw. These resistors find applications in radio receivers for controlling volume and tone. Following are the different types of variable resistors:

1. Potentiometers
2. Rheostats
3. Trimmers

Non-linear resistors:

The resistor values change according to the temperature and voltage applied and is not dependent on Ohm's law. Following are the different types of non-linear resistors:

1. Thermisters
2. Varisters
3. Photo resistors

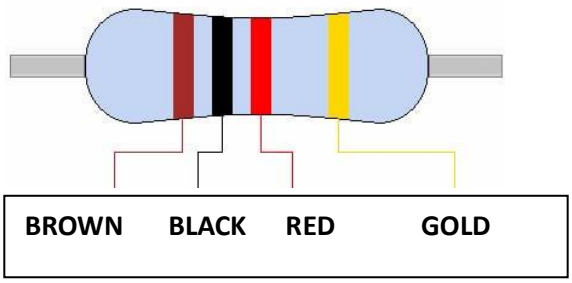
Colour code of resistor:

1. The first band indicates the first significant digit of the resistor value.
2. The second band indicates the second significant digit in the resistor value.
3. The third band serves as the decimal multipliers.
4. The fourth band indicates the tolerance level of the resistor, which shows what level above which the resistor can still function normally.
5. Sometimes, a fifth band might be present, which symbolizes the temperature coefficient of the resistor.

Image showing 1k resistor color code–

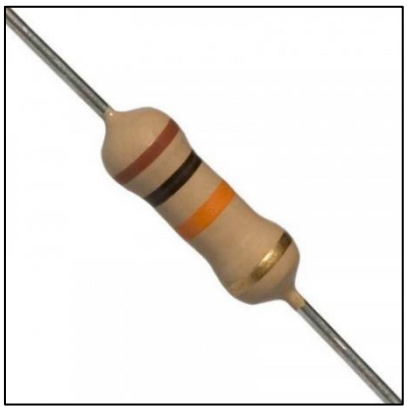
1k resistor color code is as shown in the image it is brown/black/red/gold, colour code of resistors does not depend on the power rating of resistor, the power rating of the resistor depends on its physical size and comes under standard wattage rating of 1/4 W,1/2W,1W,10W, etc.

Representational Image Of 1k ohm resistor color code:



Band	Colour	Value
1st brown		1
2nd black		10
3rd red		100
4th gold		±5%

1×10×100= 1k±5%Ω



Applications of Resistor:

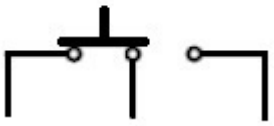
Following are the applications of resistors:

- Wire wound resistors find applications where balanced current control, high sensitivity, and accurate measurement are required like in shunt with ampere meter.
- Photoresistors find application in flame detectors, burglar alarms, in photographic devices, etc.
- Resistors are used for controlling temperature and voltmeter.
- Resistors are used in digital multi-meter, amplifiers, telecommunication, and oscillators.
- They are also used in modulators, demodulators, and transmitters.

4.2.10 SPDT mini slide switch

What is Slide Switch?

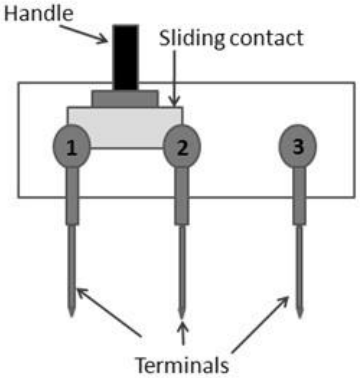
Definition of a slide switch is: It is a mechanical switch that is used to control the flow of current in a circuit by sliding the slider from the OFF (open) position to the ON (close) position known as a slide switch. This switch simply controls the current within a circuit without cutting a wire manually. These switches will stay in one position until changed into another position manually. The **slide switch symbol** is shown below.



Slide switch symbol

Slide Switch Construction:

The construction of the slide switch can be done by using metal slides that contact the plane metal elements on the switch. When the slider in the switch is moved then metal slide contacts will slide from one set of metal contacts to the other for activating the switch. This switch includes terminals, sliding contact, and handle which are discussed below.



- **Terminals:** This switch includes three terminals like one common pin & remaining two pins which compete for connection toward the common. These are best used for choosing between two power sources &swapping inputs.

- **Sliding Contact:** A sliding contact in a switch is an electrical contact where current flows throughout this contact. These contacts mainly include a stationary part like a brush & a rotating part like a commutator or slip-ring. The materials used in these contacts mainly need constant, low contact voltage, and low wear rate.

- **Handle:** A slide switch includes a bar handle, used to slide from one position to another to change its state.

How Does Slide Switch Work?

Slide switches work by using a slider to move from the OFF position to the ON position. They control the flow of current within a circuit in small projects. These switches are designed in two ways by using a metal slide and a metal seesaw.

The metal slides are used in most common designs that make contact through the flat metal parts on the switch. When the slider on the switch is moved, then metal slide contacts can move from one set of metal contacts to the other to activate the switch.

Similarly, the secondary design utilizes a metal seesaw. The slider of this switch includes a spring that pushes down on one face of the metal seesaw otherwise the other. These are maintained-contact switches, so they wait in one state until activated into the latest state and after that stay in that state until performed upon again.

Types of Slide Switches:

There are different types of slide switches available in the market which is used based on the requirement. These switches are discussed below.

SPDT Slide Switch

Sometimes, SPDT switches are known as three-way switches which are used to make or break the single conductor connection with any of two other single conductors. Generally, these switches include three terminals which are normally used in pairs.

This is a simple switch that is used as an ON/OFF or a normal control switch. The space between the pins of these switches is 0.1but they are very thin so these are not recommended for breadboard use. These switches are rated for 0.3A at 50VDC. These switches are very smaller so it occupies less space on PCB.

SPST Slide Switch

SPST slide switches has two terminals, with one terminal connected to one side of the circuit, and the other connected to the other side. When no force is applied to the slider, it remains in its "off" position, making both sides of the circuit disconnected from each other. When the slider is moved toward one terminal, it becomes "on", closing that terminal (and thus connecting it to one side).

DPST Slide Switch

These switches are mainly used to make or break the two circuit conductor's connection within a single branch circuit. Generally, these switches include four terminals. The contacts are mechanically separated by a spring, but electrically connected by a center pin (which can be removed to make the switch momentary). When the slider is moved to one side (either up or down), the center pin connects one pole to the other so that current can flow between them. When the slider is moved to another side (again up or down), however, this connection is broken and no current flows through the device.

A DPST slide switch can have many variations in its design. For example, there may be indicator lights to show which position has been selected; there may be more than two poles on each side of the switch; there may be multiple throws within each pole; etc. To know more on how a DPST switch works click here.

DPDT Slide Switch

These switches are used to make or break the two conductors into two separate circuits connection. Generally, these switches include six terminals . Two of the terminals are connected together internally inside the switch, so they are effectively tied together. The other four terminals are connected to external electrical circuits and devices in various combinations. When you apply voltage to one pair of terminals and ground to another pair, the circuit will conduct electricity in only one direction. If you apply voltage to both pairs of terminals at once, the circuit will conduct electricity in both directions simultaneously — this is known as a short circuit.

Advantages and Disadvantages:

The advantages of slide switch include the following.

- These are famous mechanical switches due to their simple functionality.
- These switches provide the visual advantage to the operator to identify exactly at which condition this switch is operating.
- These are available in standard & small sizes.

The disadvantages of slide switch include the following.

- Frequently moving of slider can lose its sharpness and fail.
- The mechanical component can exhaust.
- Issues within traffic broadcasting.
- Defenseless.

Applications:

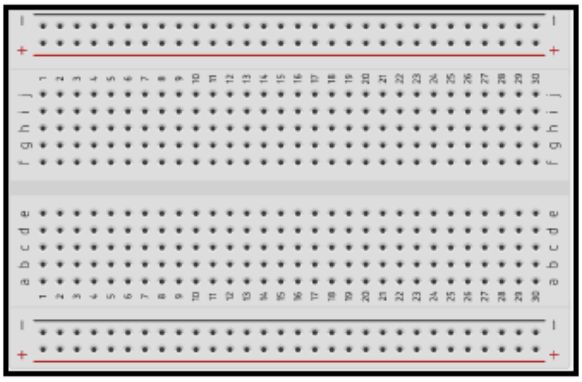
The applications of slide switch include the following.

- Generally, these switches are used in small size projects.
- This switch plays a key role in connecting two additional circuits to an existing one and connecting them optionally.
- These are applicable when switching is required between one position or another.
- These types of switches are extensively used in consumer & industrial applications: like peripherals of computers, consumer electronics devices, household appliances, test equipment, smart home systems, communication hardware, etc.
- These are used in network devices like routers, PCs, monitors, etc
- These switches are used in Telecommunication devices like modems, phones, transmitters & receivers, etc.
- Consumer products like thermostats, headphones, smartphones, remote controls, TV sets, etc.
- These are used in Test & data acquisition equipment, power tools like sanders, drills, grinders routers, jigsaws, etc.

4.2.11 Breadboard

The breadboard is a white rectangular board with small embedded holes to insert electronic components. It is commonly used in electronics projects. We can also say that breadboard is a prototype that acts as a construction base of electronics.

The breadboard is shown in the below image:



A breadboard is derived from two words bread and board. The word breadboard was initially used to slice the bread pieces. But, it was further named as a breadboard for its use in electronics around the 1970s. Hence, the term breadboard refers to these boards only and provides a quick electrical connection.

A breadboard is also categorized as a **Solderless board**. It means that the component does not require any soldering to fit into the board. Thus, we can say that breadboard can be reused. We can easily fit the components by plugging their end terminal into the board. Hence, a breadboard is often called a **plugboard**.

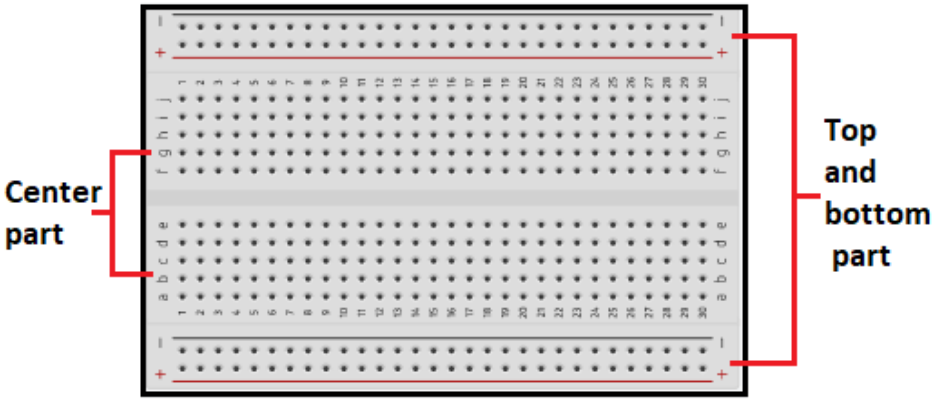
Materials used:

White plastic is the material that is used to create breadboards. Today, most of the breadboards are Solderless breadboards. We can directly plug in the electronic components and connect it with the external power supply.

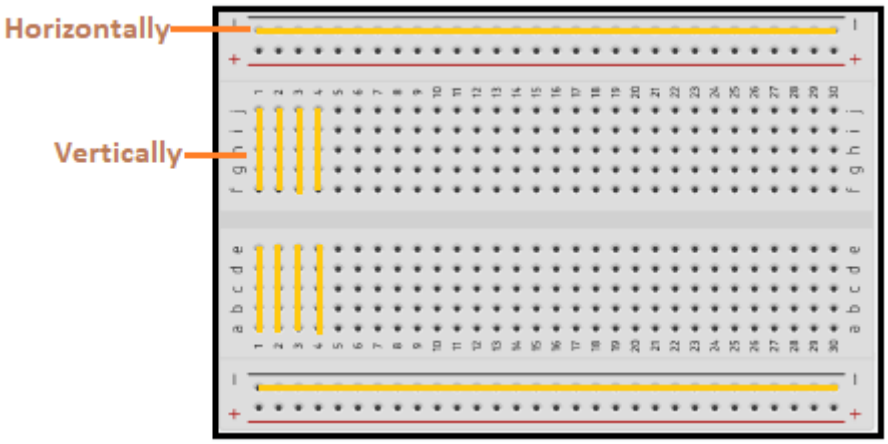
The breadboards are available as per the specified point holes. For example, 400 point breadboard, 830 point breadboard, etc.

How breadboard enables the connection when the leads of different components are plugged in?

There are three parts in a breadboard, as shown below:

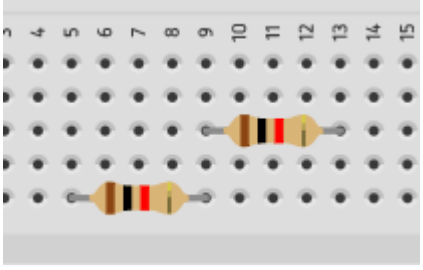


The top and bottom holes of a row in a breadboard are connected horizontally, and the center part is connected vertically, as shown below:



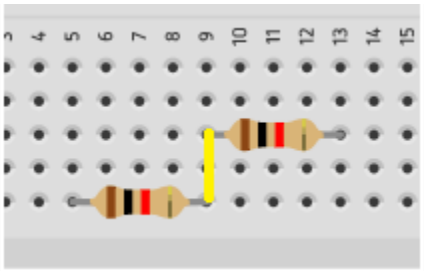
It means a single horizontal line of a breadboard has the same connection. It is because the metal strip underneath the breadboard at the top and bottom are connected horizontally. Hence, it provides the same connection in a row. The two top and bottom parts of a breadboard are generally used for power connections.

The vertical connection of the center part means a single vertical line in a breadboard provides the same connection. It is useful when we need to connect the different components in series.



For example, let's connect two resistors in series. These two resistors can be connected in series in different ways, as shown below:

It is because the metal strips underneath the breadboard at the center are connected vertically. Hence, it provides similar connectivity through a particular column, as shown below:



The connection between two different components can be created by inserting a lead in common. For example, a jump wire that acts as a connection between the LED and battery terminal can be connected in any hole in the same vertical line.

Types of Breadboard:

There are two types of the breadboard, namely **Solderless** and **solder breadboard**.

Solderless breadboards - As the name implies, Solderless boards do not require any soldering after the electronic components are plugged in.

The leads or ends of the components are inserted into the holes of a breadboard for its functioning.

Solder breadboard - The solder breadboard is also a board that has a tiny hole embedded into it. We can insert the terminal of the electronic components into the board. After the connection is rechecked, we can solder these components.

The common difference between solder and Solderless breadboards is the **reusability**

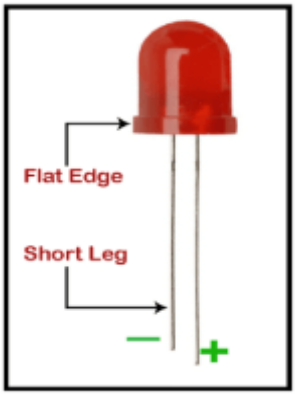
Connection setup through a breadboard:

Here, we will discuss the connection setup through the breadboard with the help of few examples. It will help us to understand different connections in it.

Example 1: Blinking an LED

The components required for the above example are an LED (any color), a breadboard, two jump wires, a resistor, and a battery. Here, we have chosen an LED of red color. The resistor is connected in series with the LED to limit the current across the LED.

LED has two terminals, namely cathode (negative terminal) and anode (positive terminal). The structure of LED is shown below:

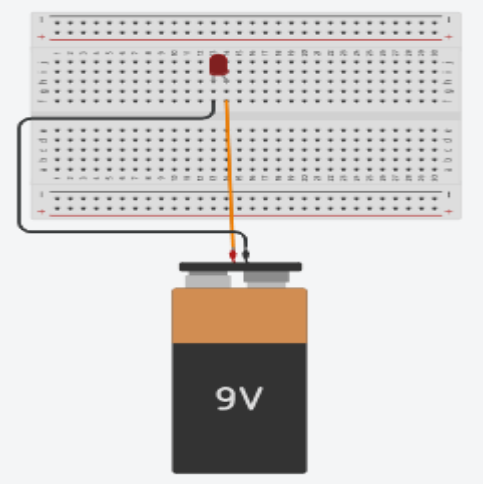


Connection Setup:

The connection setup is listed in the below steps:

1. Plug-in the two terminals of the LED into the two tiny holes of the breadboard. We can add or skip the resistor because the battery already provides a limiting current.
2. Connect one end of a jump wire to the anode of an LED and the other end of the jump wire to the positive terminal of the battery. Similarly, connect the cathode of an LED to the negative terminal of the battery.

As soon as the circuit is complete after the terminals are connected to the battery, the LED will light. The circuit so formed will appear like the image shown below:



The above circuit connection depicts that the LED and resistor are connected in series. Similarly, we can easily create various projects and circuits with the help of the breadboard.

Advantages of Breadboard:

The advantages of using breadboard are listed below

- **Temporary prototype**-We can build a temporary prototype for the projects with the help of a breadboard.
- **Reusable**-Today, Solderless boards are mostly used in various applications. It does not require any soldering to fix the components. Hence, it can be reused.
- **Lightweight**-The breadboard is made of white plastic, which is light in weight.
- **Easy experimentation**- We can quickly insert the leads of the components into the tiny holes of the breadboard. The circuit can be created using various components and circuit design.
- **Inexpensive**-The breadboards are easily available. It also cost less.
- **Easy to use**-It does not involve any complex parts. We can easily insert the required number of components.

- **No drilling required**- The hoes are already embedded in the board. Hence, we do not require any drilling to insert the electronic components.
- **Quick modifying capability**- We can easily switch or remove the components from the board.
- **Available in various sizes**- The breadboards are available in various sizes. We can select the desired size as per the number of components.
- **Easy to adjust** -The breadboard is easy to adjust in the project or connection setup.

Disadvantages of Breadboard:

The disadvantages or limitations of the breadboard are listed below:

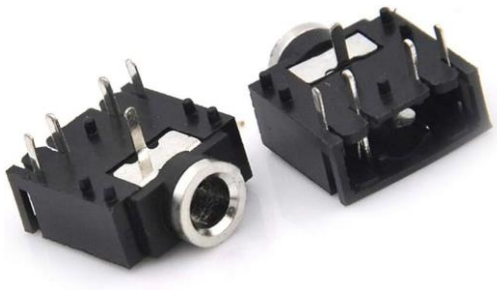
- Not suited for high current applications
- Low-frequency Solderless boards are limited to low-frequency applications.
- Requires more physical space for simple circuits.
- A high number of connections in the Solderless board make the circuit messy due to a greater number of wires.
- The circuit design does not work well for high-speed design.
- The plugging and unplugging can disturb the other connections.
- Less reliable connections.
- Limited signaling.

4.2.12 3.5mm Female Headphone jack

A common connector for plugging in a standard pair of music headphones such as the ones found on music players, computers and most other electronic devices with audio outputs.

It can support stereo and/or microphone, depending on the number of separate connector rings on the jack.

Some phones offer only a 2.5 mm jack, which is a smaller variety of the same principle.



Headphones supplied with mobile phones usually have a mic somewhere along the cable and a remote button that allows for managing calls without using the phone.

Some manufacturers opt for placing a 3.5mm audio jack on this remote control instead of directly on the phone itself. The reason for this is that 3.5mm jacks take up quite a lot of internal space; plus, in this way the user gets to keep the remote control/mic functionality while using third-party headphones.

Origin Of The 3.5 Mm Audio Jack

The origins of the 3.5mm jack can be traced all the way back to the 19th century. Back in 1878, a predecessor to the 3.5mm jack, a 6.35mm jack (also called the ¼" inch jack) was developed as a 'phone connector' used by telephone operators to direct calls manually. Back then, you couldn't just dial a number and be connected straight away. You had to speak to an actual person (the operator) and request that they connect your call forward.

In the late 19th century, Thomas Edison made a primitive version of modern headphones using vacuum tubes connected to the phone using the quarter-inch jack. This was perhaps the first time one could have a private telecommunication experience. However, it wasn't until much later that the jack was actually used for the purpose of listening to music.

How Does The 3.5 Mm Audio Jack Work?

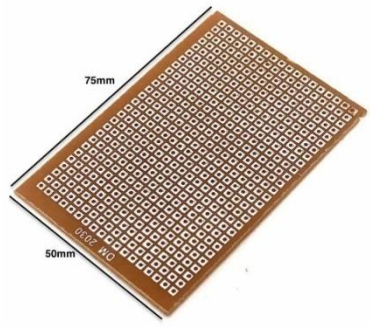
The 3.5mm jack that we so widely use today is technically referred to as a TRS (Tip Ring Sleeve) connector. The tip, the ring and the sleeve are three integral parts of the jack. These are labelled below:

The tip transmits a current to the left speaker/earpiece, the ring transmits to the right speaker/earpiece, and the sleeve grounds the port. The black bands in between are called isolation grommets, which ensure that there is no unwanted mixing of sound between the right and left channels. If you pay attention, some 3.5mm jacks have one isolation grommet, some have two, while some even come with three. One isolation grommet means the connector comes with just a tip and a sleeve without a ring, which leads to a mono sound output. If you had earphones with a single ring, you would essentially have sound coming from only one of the earpieces. One-grommet jacks are primarily used for guitars.

Most 3.5mm jacks have 2 grommets that create a stereo sound, giving the user some semblance of surround sound. Jacks with 3 grommets, typically found in Apple's old wired earphones, have the extra metal ring for the microphone input. So, the next time you're at an electronics store looking for a good pair of earphones, look at the number of rings to determine which ones actually support mic functionality.

4.2.13 Custom Printed Circuit Board (PCB)

What is a Printed Circuit Board?



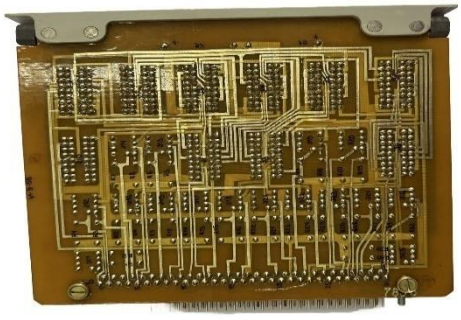
A printed circuit board (PCB) is an electronic assembly that uses copper conductors to create electrical connections between components. PCBs

also provide mechanical support for electronic components so that a device can be mounted in an enclosure.

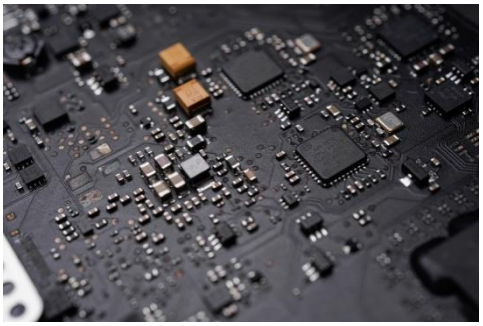
PCBs Past and Present:

In the past, electronics were designed and assembled from small integrated circuits (ICs) and discrete components, which were connected together using wires, and the components were mounted to a rigid substrate. This original substrate was initially a material called bakelite, which was used to replace the top ply on a sheet of plywood. The number of wires was so great that they could get tangled or inhabit a large space within a design. Debugging was difficult and reliability suffered. Manufacturing was also slow, where multiple components and their wired connections were manually soldered.
Old PCBs often looked like the system shown below, where copper traces are exposed and many through-holes are used to hold large, bulky components.

Today, standard designs can have many small components, such as tiny ICs, very small passive components, and advanced chips with very high pin counts. It's impossible to manually connect all of these components together with soldered wires, so the copper connections are deposited directly on insulating substrates as described in the above manufacturing process. Many of today's devices are advanced high density interconnect (HDI) designs with thousands of connections and multiple electrical interfaces, powering everything from smartphones to heart rate monitors to rockets.



Older printed circuit board



Modern printed circuit board

Types of PCBs:

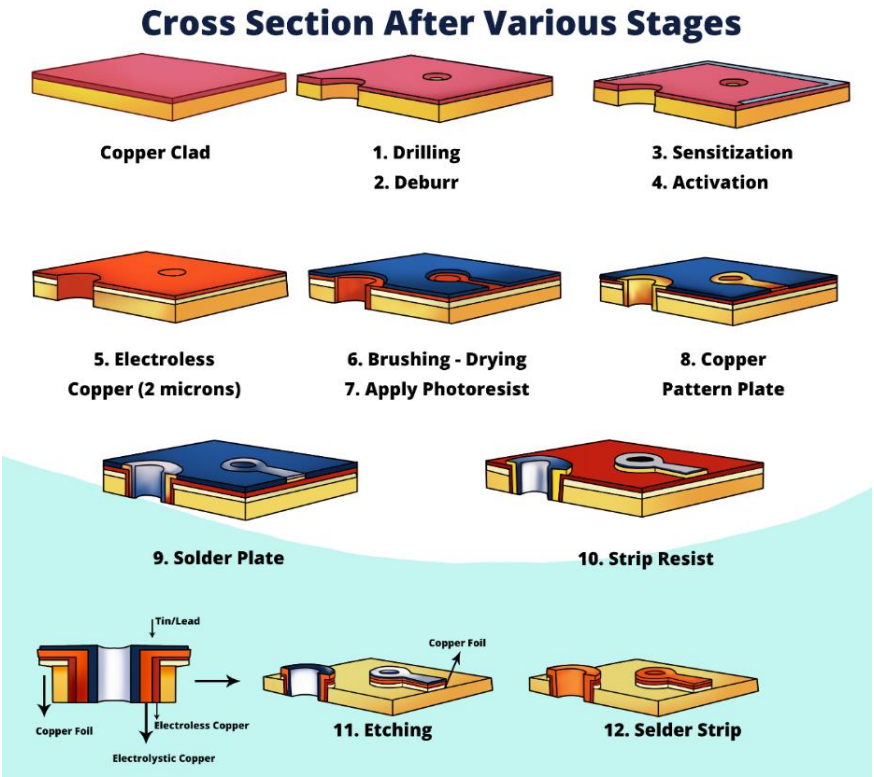
The common types of these are:

- **Single-sided** - This board only has components mounted on one surface. The back surface is typically fully copper (ground) and coated with a solder mask.
- **Double-sided** - This type of circuit board has components mounted on both surfaces. Each surface is defined as a signal layer in the PCB stack-up, so the surfaces will contain traces that carry signals between components.
- **Multi-layer PCBs** - These boards have conductors on internal layers that carry electrical signals between components, or the internal layers could be conductive plane layers. Multi-layer PCBs may be single-sided or double-sided.

- **Rigid-flex PCBs** - Rigid-flex PCBs use a flexible polyimide ribbon that connects two or more rigid sections in a printed circuit board assembly. A rigid-flex board might be used when the design must have some movable element, such as a folding or bending enclosure.
- **Flex PCBs** - Fully flexible PCBs do not use any rigid materials and are made entirely of flexible polyimide ribbons. These boards can have components mounted and soldered on the, just like rigid and rigid-flex printed circuit boards.
- **Printed flex PCBs** - These PCBs use a flexible material as the base, and copper conductors are printed onto the flexible material in an inkjet process or in a similar additive process. The resulting boards are very similar to flex PCBs.
- **Metal-core PCBs (or insulated metal substrate (IMS) PCBs)** - These boards use a metal slab in the core layer (normally aluminum) in order to provide much greater rigidity and heat dissipation than in typical rigid printed circuit boards. The metal-core PCB design manufacturing process is quite different from the standard rigid PCB design manufacturing process, and there are a few design points to consider to ensure solvability. These boards are common in high-power lighting and some industrial applications.
- **Ceramic PCBs** - These boards are less common and are used in applications that require very high thermal conductivity such that the board can dissipate large amounts of heat away from components.
- **HDI PCBs** - These PCBs use very high pin count components that require a specialized manufacturing process and specialized materials to accommodate a very high density of copper connections.
- **UHDI and Substrate-like PCBs** - These PCBs are so small and dense that they have bypassed capabilities of subtractive etching, and instead they require a specialized additive manufacturing process used to build IC packages.

Stages of PCB design:

All PCBs are built from alternating layers of conductive copper with layers of electrically insulating material. Conductive features on printed circuit boards include copper traces, pads, and conductive planes. The mechanical structure is made up of the insulating material laminated between the layers of conductors. The overall structure is plated and covered with a non-conductive solder mask, and silk screen is printed on top of the solder mask to provide a legend for electronic components. After these fabrication steps are completed, the bare board is sent into printed circuit board assembly, where components are soldered to the board and the PCBA can be tested.



During manufacturing, the inner copper layers are etched, leaving the intended traces of copper for connecting components in the circuit board. Multiple etched layers are laminated in succession until the printed circuit board stack-up is complete. This is the overall process used in PCB design fabrication, where the bare board is formed before passing through a printed circuit board's assembly process.

4.2.14 Communication Channels

4.2.14.1 Male to Male Jumper wires - It is mainly used to connect the peripherals with the Arduino Controller as well as to connect the peripherals with each other also. But sometime it can also be used as any bus cable for its beautiful arrangement.

4.2.14.2 Female Header Pins - Female headers are the corresponding parts of headers that mate directly to male pin headers. Female headers (often referred to as socket headers) are made of a plastic housing with many sockets where the male pins are meant to fit. Just like pin headers, female headers are offered in one to four row configurations with a varying number of pins per row. They, paired with pin headers, can also be applied to applications in various industries that require PCBs to be connected. Headers, therefore, provide effective, cost-saving solutions for board-to-board interconnects.

4.2.14.3 Male Header Pins - Male pin header consists of one or more rows of metal pins moulded into a plastic base, often 2.54 mm (0.1 in) apart, though available in many spacings. Male pin headers are cost-effective due to their simplicity.

4.2.14.4 Mini USB Cable 2.0 - It is used for coding the Arduino. The +5V power can also be given to run the Arduino through this cable from any USB device.

❑ **Non – Electronic Components :**

The abbreviation 'Non-electronic' tells that it is not related to the electricity and hence not related to the electrons. So they are basically physically components used to make the structural body of any project.

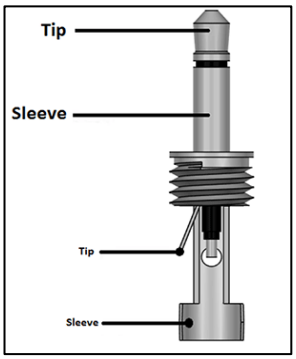
4.2.15 3.5mm Earphone:

The meaning of Earphone is a device that converts electrical energy into sounds waves and is worn over or inserted into the ear.

Brief Description:

Now-a-days 3.5mm is the universal audio jack size to be found in Smartphones, PC and Laptops. Also, for hobbyists 3.5mm audio jack is a useful components for projects that plug into headphone jacks. There are different types of 3.5mm audio jack available with different application like TS, TRS, and TRRS, but the most common that we see in daily life is TRS and TRRS.

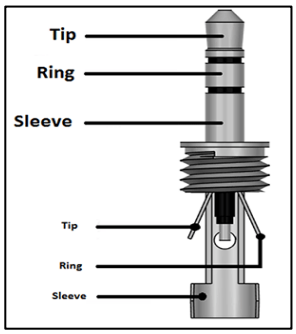
Types of 3.5mm Jack



1. TS Type Male Audio jack-These types of audio jacks does not support stereo sound and microphone, which means there is no left and right. You will get same sound from both the sides. Below is the pinout of TS type male audio jack.

Application: Still used on musical equipment (especially in electric guitars) and aviation radios.

2. TRS Type Male Audio jack

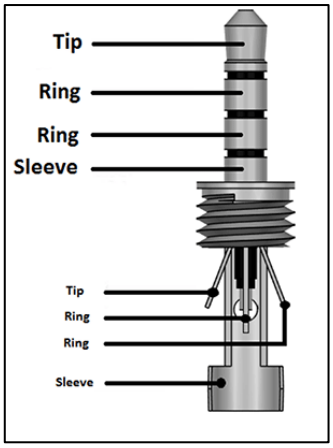


Here, in the name of TRS, 'T' stands for Tip, 'R' stands for Ring and 'S' stands for Sleeve. These types of audio jacks supports stereo sound and but doesn't supports microphone. So, using this you can only listen music but can't talk on calls. Below is the pinout of TRS type male audio jack.

Application: Speakers, microphone, keyboards and etc.

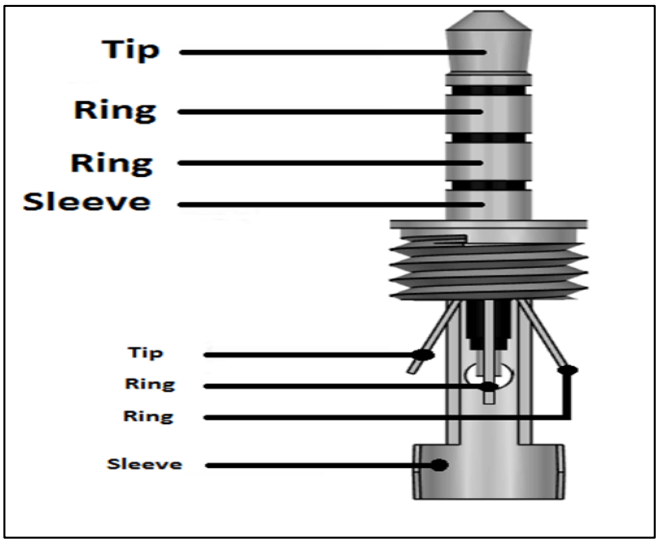
3. TRRS Type Male Audio jack

A TRRS type audio jack have four conductors and are most popular with smartphones and tablets. The sequence of TRRS type audio jack is Tip-Ring-Ring-Sleeve and it includes both stereo and microphone functionality. There are number of standards to be used while making these audio jacks such as OMTB and CTIA. This is the reason behind your smartphone doesn't supports other brands of headphone. Below is the pinout of TRS type male audio jack.



Application: Used in many branded companies headphones like Apple, Nokia, Samsung, Panasonic and etc.

3.5mm Earphone Pinout Configuration:

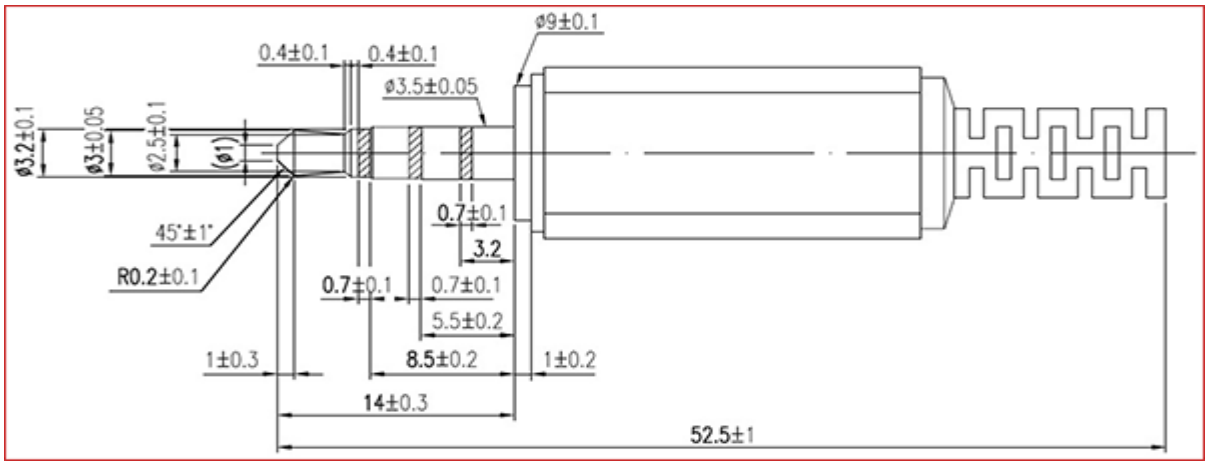


How to Use a Audio jack?

For using a 3.5mm audio male jack for your projects or Prototypes, you have to solder wires with the pins of the jack. Remove the above plastic casing and you will see the pins of the jack as shown in the images above. Now use multi-strand wires to solder with the pins and then again cover it with the Plastic casing.

2D Model

Pin No.	Pin Name	Description
1.	Tip	Left
2.	Ring	Right
3.	Ring	Ground
4.	Sleeve	Microphone



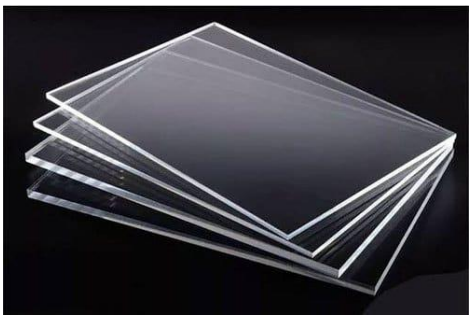
4.2.16 16 GB Memory Card

A memory card is an electronic data storage device used for storing digital information, typically using flash memory. These are commonly used in digital portable electronic devices, such as Digital cameras as well as in many early games consoles such as the Nintendo Wii. They allow adding memory to such devices using a card in a socket instead of protruding USB flash drives.

Common types of flash memory card include SD cards (including microSD), Sony's Memory Stick and CompactFlash. However, SD cards are by far the most common type of memory cards nowadays.

4.2.17 Transparent Acrylic Sheet

Acrylic (polymethyl methacrylate or PMMA) is a transparent plastic that is known to be more common by the trade name “Plexiglass” The material is similar to polycarbonate because it is suitable for use as an alternative to collision against the glass. It was first produced in 1928 and was taken to market five years later by Rohm and Haas Company. This is generally considered the most obvious plastic available.



The Physical Properties of Acrylic Sheets:

Acrylic plastic is a transparent thermoplastic homopolymer that is often colloquially referred to as plexiglass. This plastic has unique properties that make it ideal for a variety of purposes, ranging from basic household items to fiber optic cables that power the world. These properties have allowed acrylic to enjoy immense popularity in manufacturing as well as DIY projects:

- High impact resistance
- High optical clarity
- Innate weatherability and UV resistance
- Excellent dimensional stability
- Lightweight
- Excellent chemical resistance

What is Acrylic Used For?

Whether you're looking for strength, longevity, or clarity, acrylic plastic offers several advantages over other materials and is extremely versatile—meaning it can be used in a variety of different applications. Here are a few examples:

- Skylights and commercial windows
- Retail signage and displays
- Aquariums and terrariums
- Protective barriers for manufacturing

4.2.18 Carrying Box

ITOS365 Handmade Wooden Small Box, perfect for organize any project . This beautiful prohject organizer is meticulously crafted from high-quality wood, and features intricate hand carvings that add a touch of elegance and charm to any dressing table. This small but sturdy box has enough room to store your favorite pieces. The lid of the box is adorned with a delicate floral pattern, and opens to reveal a velvet-lined interior that keeps project safe and protected.

- Elegant & Stylish- Beautiful wooden handmade box. It can be the source for your styling. Store your valuables project work.
- Hand Carved Design-The intricate hand-carved design make the box even more special and unique. This design include anything from floral patterns to geometric shapes, making it a beautiful.
- Premium & Exquisitely Carved- This comes with a velvet inner lining and to keep the material scratch free and safe. It is one of the most premium, exquisitely carved and crafted available anywhere.
- Small & Compact Size- The box designed for project is small and compact enough to fit on a dresser or bedside table without taking up too much space. This makes it easy to access quickly and easily.
- Unique & Perfect Gift- The classy wooden box can be used as a gift .

4.3 Platform

In this section, we will explain all the detail referred to the application code developed and the programming language selected to deploy it.

4.3.1 Arduino 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 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.

Verify Checks your code for errors compiling it.

Upload Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer".

New Creates a new sketch.

Open Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.

Save Saves your sketch.

Serial Monitor Opens the serial monitor.

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 below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
- Add File... Adds a supplemental file to the sketch (it will be copied from its current location). The file is saved to the data subfolder of the sketch, which is intended for assets such as documentation. The contents of the data folder are not compiled, so they do not become part of the sketch program.

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

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

Help

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

- 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).

Before compiling the sketch, all the normal Arduino code files of the sketch (.ino, .pde) are concatenated into a single file following the order the tabs are shown in. The other file types are left as is.

Uploading

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

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The 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 #include statements from the top of your code. There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

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

For details on creating packages for third-party hardware, see the Arduino Platform specification.

Serial Monitor

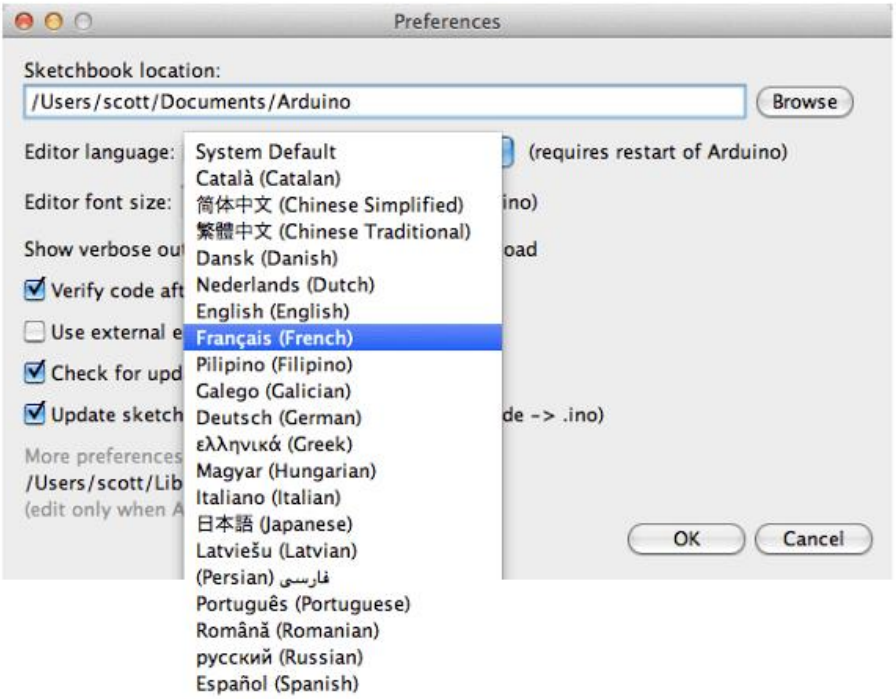
This displays serial sent from the Arduino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux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

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

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

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.

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager 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.

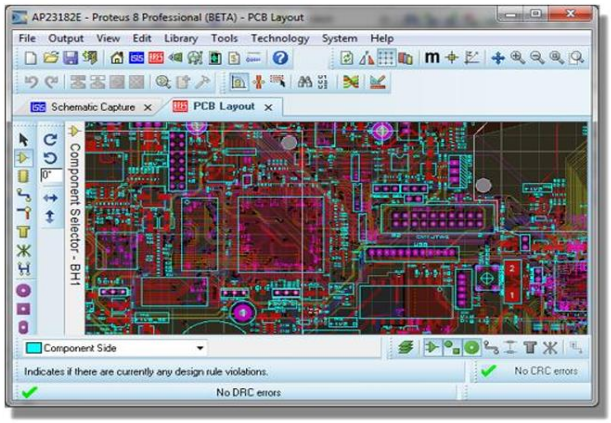
- Arduino Yún An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino Uno An ATmega328P running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Diecimila or Duemilanove w/ ATmega168 An ATmega168 running at 16 MHz with auto reset.
- Arduino Nano w/ ATmega328P An ATmega328P running at 16 MHz with auto-reset. Has eight analog inputs.
- Arduino Mega 2560 An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Mega An ATmega1280 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Mega ADK An ATmega2560 running at 16 MHz with auto-reset, 16 Analog In, 54 Digital I/O and 15 PWM.
- Arduino Leonardo An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino Micro An ATmega32u4 running at 16 MHz with auto-reset, 12 Analog In, 20 Digital I/O and 7 PWM.
- Arduino Esplora An ATmega32u4 running at 16 MHz with auto-reset.
- Arduino Mini w/ ATmega328P An ATmega328P running at 16 MHz with auto-reset, 8 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Ethernet Equivalent to Arduino UNO with an Ethernet shield: An ATmega328P running at 16 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Fio An ATmega328P running at 8 MHz with auto-reset. Equivalent to Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328P, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino BT w/ ATmega328P ATmega328P running at 16 MHz. The bootloader burned (4 KB) includes codes to initialize the on-board Bluetooth® module, 6 Analog In, 14 Digital I/O and 6 PWM..
- LilyPad Arduino USB An ATmega32u4 running at 8 MHz with auto-reset, 4 Analog In, 9 Digital I/O and 4 PWM.
- LilyPad Arduino An ATmega168 or ATmega132 running at 8 MHz with auto-reset, 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328P An ATmega328P running at 16 MHz with auto-reset. Equivalent to Arduino Duemilanove or Nano w/ ATmega328P; 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino NG or older w/ ATmega168 An ATmega168 running at 16 MHzwithout auto-reset. Compilation and upload is equivalent to Arduino Diecimila or Duemilanove w/ ATmega168, but the bootloader burned has a slower timeout (and blinks the pin 13 LED three times on reset); 6 Analog In, 14 Digital I/O and 6 PWM.
- Arduino Robot Control An ATmega328P running at 16 MHz with auto-reset.
- Arduino Robot Motor An ATmega328P running at 16 MHz with auto-reset.
- Arduino Gemma An ATtiny85 running at 8 MHz with auto-reset, 1 Analog In, 3 Digital I/O and 2 PWM.

4.3.2 Proteus 8.11 Professional

This is a brief introductory review of the app Proteus. I have tried to be honest with the review as I always do. Hope this review will give a good insight into using the app easily. The special features of the app and the Pros and Cons have been listed below. Hope this article is useful.

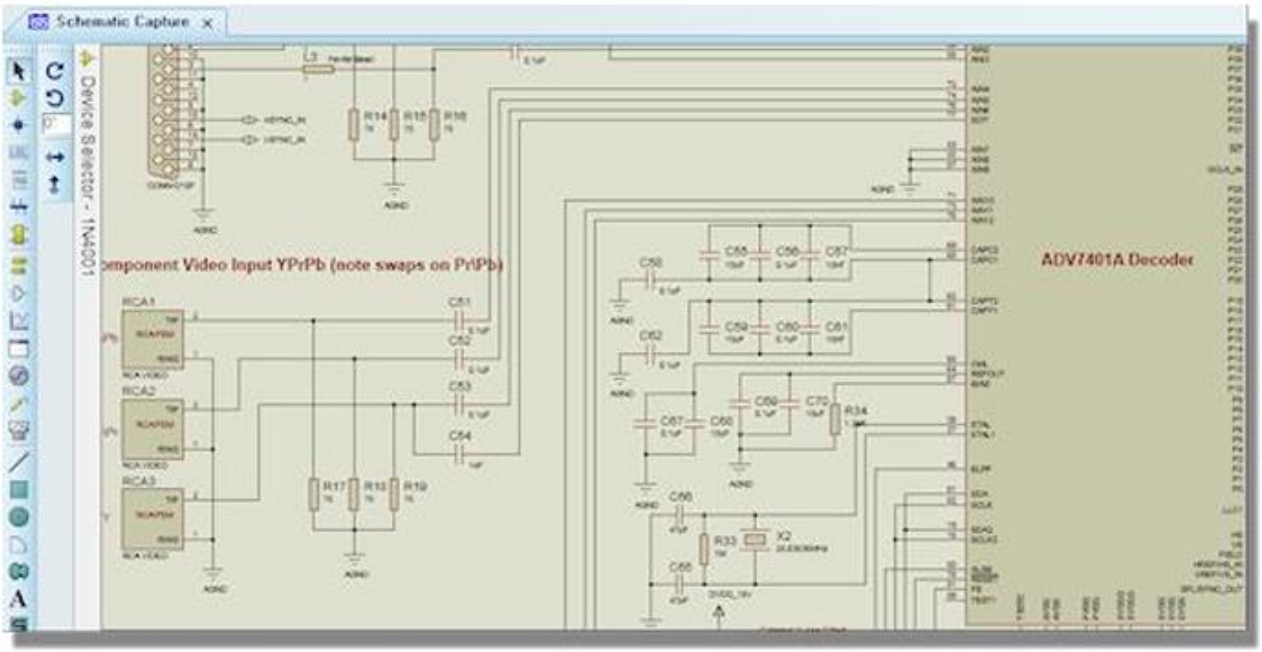
Introduction

Want to test your circuit diagram? Breadboarding it can be a good option, but it is easy to get confused if your circuit is big. You can try creating a PCB and testing on it, but is a very time-consuming task and even takes a lot of effort. Simulating it is the best idea. You can use Proteus 8 Professional. Proteus 8 Professional is a software which can be used to draw schematics, PCB layout, code and even simulate the schematic. It is developed by Labcenter Electronic Ltd.



Features:

Schematic drawing –

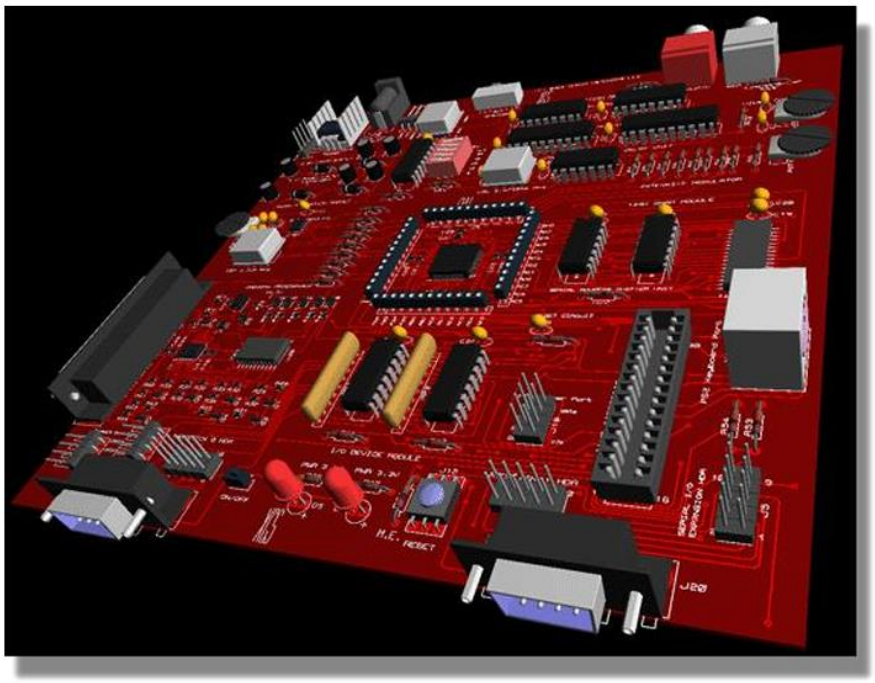


Drawing the schematic is very easy using Proteus. You can click the "Pick devices" button and select the desired component. You can draw wires by clicking on the terminal of the component or Vcc, Ground, etc.

Simulation: Circuit ready for testing? You can test it using Proteus's simulation feature. Many of the components in Proteus can be simulated. There are two options for simulating: Run simulator and advance frame by frame. The "Run simulator" option simulates the circuit in a normal speed (If the circuit is not heavy). "Advance frame by frame" option advances to next frame and waits till you click this button for the next time. This can be useful for debugging digital circuits. You can also simulate microcontrollers. The microcontrollers which can be simulated include PIC24, dsPIC33, 8051, Arduino, ARM7 based microcontrollers. You can download the compilers for Proteus or use different compiler and dump the hex files in the microcontroller in Proteus. You can even interact in real-time with the simulation using switches, resistors, LDRs, etc. There are even virtual voltmeter, ammeter, oscilloscope, logic analyzer, etc.

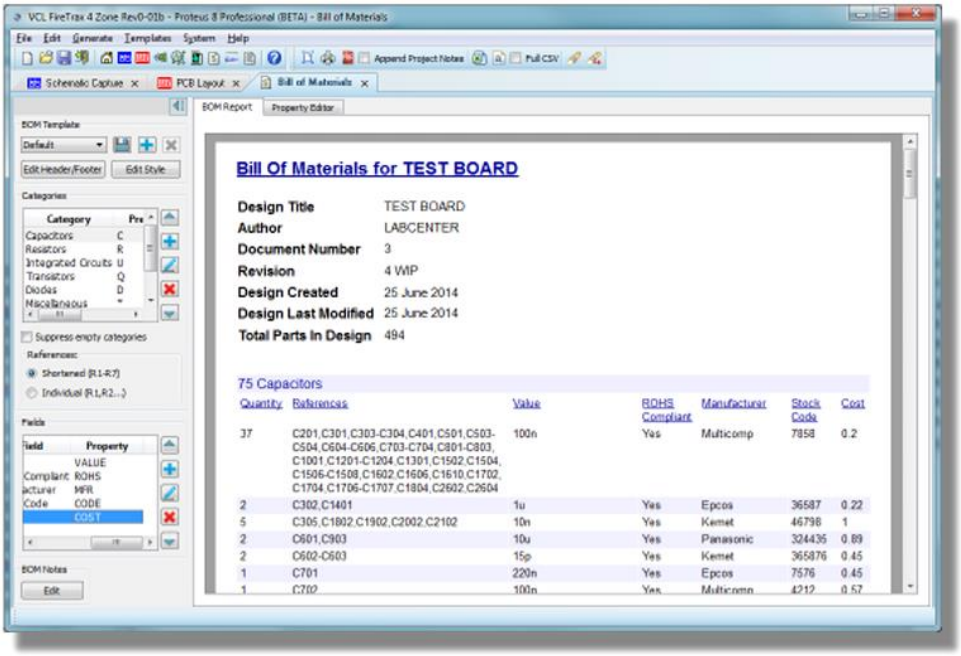
Designing PCB: Designing PCBs are easy using Proteus. You can make your own design or let Proteus do that for you. Making your own design is simple, you just have to place the components used in the schematic and draw traces over them. Don't worry about violating any design rules because it automatically detects design rule (DRC) errors. You can also let Proteus do the work for you. You can place the components on their respective places and select the "auto route" option. This will automatically draw multiple variations of traces and selects the best one. There is also an "Auto placer" option present; this option needs you to specify the board dimensions by drawing the shape and size of the board so that, it can place the components within the board boundaries. So, all you have to do is to make the schematic.

3D visualization:



You can now look at your finished PCB design without actually making one. The 3D visualization feature virtually creates a 3D model of the PCB. You can use this to verify whether it will look like you imagined.

Bill of Materials:



Tired of keeping track of the budget? Now you can use Proteus 8 professional's "Bill of materials" feature. You just have to specify the cost of each of components. After specifying the cost for all items, Proteus automatically makes a list of all components used with their individual price and total price.

Pros and cons:

Pros:

•You can interact with the running simulation, using switches, etc.

• Virtual electronic measurement instruments are available.

Cons:

• User interface could have been made better.

The conclusion: Proteus 8 Professional is software which can be used to draw schematics, PCB layout, code and even simulate the schematic. You can simulate your work and be more efficient in completing the task at hand. Hope this articles about Proteus 8.11 Professional is useful. Thank you for reading.

4.3.3 Easy EDA

What is Easy EDA?

Easy EDA is one kind of PCB design & Simulation Online Tool especially for Electronic Circuit Design in engineering projects. For designing all these, we need to login otherwise sign up with the Easy EDA website. This online tool supports any operating system like Windows or Linux or Mac as well as any Browser like Internet Explorer or Firefox or Chrome or Safari. Once a circuit designing is done, we can also edit many times if any component or connection is misplaced because the data can be stored on the

website. So the designed file can be accessed anytime. So day by day, several features are including in this tool, so this tool is very helpful for beginners as well as engineers.

Basics/Features of Easy EDA:
The main features of EasyEDA include the following.

- This tool is very simple, user friendly, easier, and very powerful to draw the circuits
- Schematic Capture can be done
- PCB Layout
- It can be operated anytime, anywhere on any device
- Team Cooperation in Real-time
- Online Sharing can be possible
- Hundreds of Open Source Projects can be designed
- Script Support
- Millions of Free Libraries
- Incorporated printed circuit board fabrication
- Components selection can be done easily through direct links
- Open-source designs
- A single tool for all the manufacturing processes of PCB
- Simulation
- Automatic Upgrade of Libraries
- Free or Low-cost subscription

Steps to Design Circuits in Easy EDA:

The following steps are used to design circuits in Easy EDA.

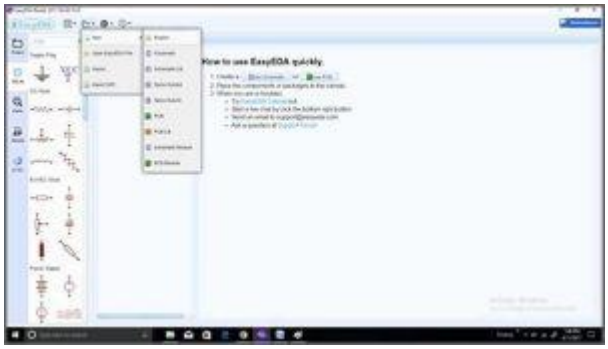
- Getting Started with EasyEDA for PCB Design
- Design a Schematic Diagram
- Checking the Connections
- Simulating the Circuit
- After saving the circuit, click on “Green Button” on top of the screen for Simulation and choose “Run the document”.
- Conversion of PCB Layout through EasyEDA
- Advantages & Disadvantages as Compared to Other Tools
- How to Place an Order
- Conclusion

The EDA is the abbreviation of electronic design automation and it is one kind of tool used to design the desired PCB quickly and effortlessly. This tool is very suitable for beginners and engineers who want to experiment or try different circuits with free of cost. This tool can be downloaded or run online. First, you need to log in to the EasyEDA website like https://easyeda.com.

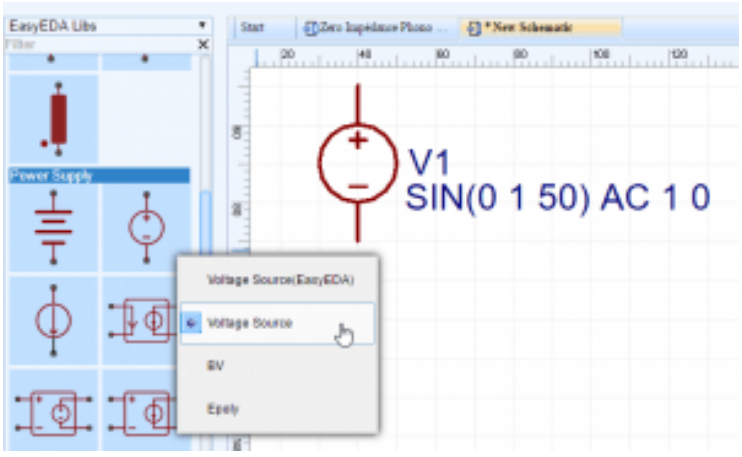
So the screenshot of the website is shown above.

Design a Schematic Diagram:

Once the website login is done by entering login credentials, start working on the circuit diagram by beginning a new project. Once the new project is created, you need to go into the drawing board & select all the required components from the libraries which are required. If you cannot discover the required component, then the option like MORE LIBRARIES is there to search for your required component. Select all the required components from the left side of the monitor and start drawing the schematic diagram. Click on the required component to choose & click on the canvas once more to drag that component.



Once the component is placed then click the Esc button or Right-click to exit. To connect the components, wiring is very important. So it can be done by dragging the wire among the terminals of components. And also you can change the component's properties otherwise attributes by clicking over the component. On the top menu, you can found some shortcut keys also to use. Once the drawing is done, the schematic diagram can be saved with the name of the circuit in the project folder so that simulation can be done easily.



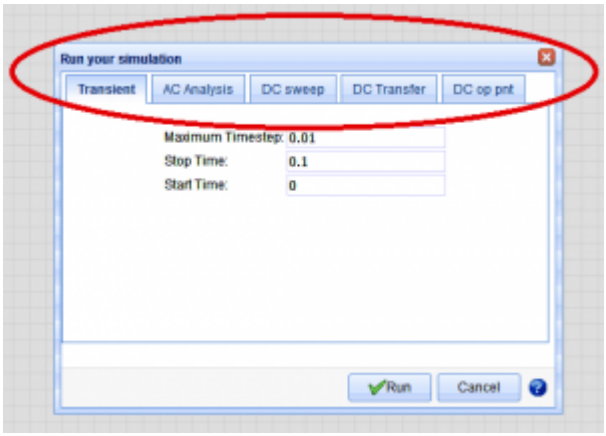
Checking the Connections

Once the circuit designing is finished, you need to cross-check each and every connection. For that, you need to open the folder like Nets on the Design Manager tab. For instance, if the connections of components within the circuit are not properly connected so automatic warnings can be generated. While designing the circuit, the following errors may occur.

- Partly cover the new path contact region of the element.
- Overlay of contact pads for elements, because they appear to be connected, however, they are not.
- There is no require underestimating the circuit size. Reducing the PCB board to save the size, cost of the device to reduce the circuit.
- The paths are vertical to the contacts. Once you move anycomponent, you can condensethe paths without observing.
- You need to verify the connections of tracks always to contacts of the components over the Design Manager tab.

Simulating the Circuit:

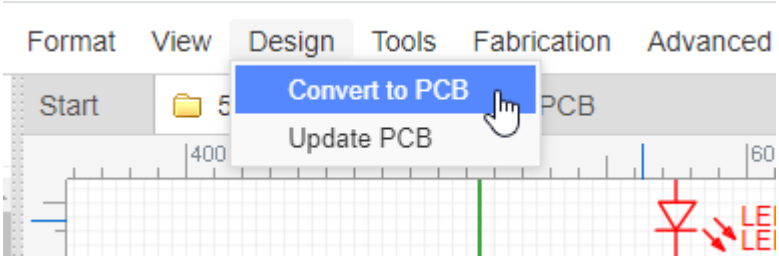
Once the circuit is saved, look at the green button located on the top of the menu. So, click that green button for Simulation & immediately select the button “Run the document”. Afterward, you have to arrange the simulation you wish to run from the available types of simulation.



Once the simulation is done, we can notice the diagram at the selected terminal that must be selected by bringing the probe toward that end. You can choose the probe to drag it toward the end where you wish to observe the chart.

Conversion of PCB Layout through Easy EDA:

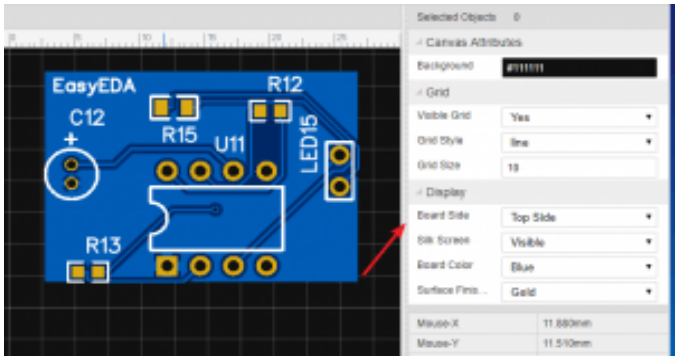
Once we enter into the schematic board then click over the option line PROJECT TO PCB.



When you click over that then the PCB design board page will be opened. If there are any components over the schematic diagram which doesn't contain PCB traces, then it asks to select the most suitable one to continue. Select the suitable one depending on your observation & click on submit.

Once you entered into the PCB design, all the used components will be dispersed approximately the PCB model board.

Connect all the required components in order. When the arrangement of every component is done you must include input at one ending whereas output on the other ends.



PCB Layout

Now trace the light blue color lines over the PCB board without interrupting another one, you can perform several loops. Once the tracing is finished, select a PRINT option within the FILE menu, so that you will find the PCB board trace.

Print the suitable layer by choosing the option like top layer or bottom. Finally, you can print the image onto sleek paper for etching the PCB otherwise you can provide the module to makers for mass production.

Advantages and Disadvantages:

The advantages & disadvantages of Easy EDA as compared to other kinds of software mainly include the following.
The advantages are

- This kind of tool is very simple to utilize & also user-friendly.
- An enormous library for different components lets you illustrate schematic diagrams rapidly.
- This tool permits you to download a schematic diagram as well as other files from other tools like Eagle, KiCad, etc
- No need for installation, we can directly operate through a web browser
- It supports all types of operating systems
- It can access from any system, anywhere even on mobiles also by connecting it to the internet.
- Once the server software is updated then automatically this tool will be updated and applied to all the users.
- Design can be easily shared with other clients online.
- You can get the new designs from the community.

The disadvantages are

- This is a new tool as compared to KiCad so it takes some time to get awareness.
- Without the internet, this tool cannot be operated.

How to Place an Order?

On the Easy EDA site, you can simply place an order for PCB by visiting the website. One can easily login through g-mail or by creating a free account. If you are using a desktop client, utilize similar data to log in there. On the website of EasyEDA, a page will be opened by the software. From there, PCB can be easily ordered by using the reasonable prices offered.

Thus, this is all about an overview of the Easy EDA tool. This tool is very helpful for engineers, teachers, students & hobbyists to make powerful schematic capture, PCB layout & mixed-mode spice simulation. This circuit designing software has many features EasyEDA has all the features which take your design from beginning to production. This tool mainly targets to give a good experience to every beginner or user. This is the reason to call this tool as easy.

4.3.4 Fritzing

One of the most important aspects of creating new projects is the initial design and vetting process to figure out whether your circuit will work or not. Fritzing is a free, open-source design tool that provides users with circuit design, schematics, and PCB layout creation. This article serves as an introductory overview of what Fritzing has to offer.

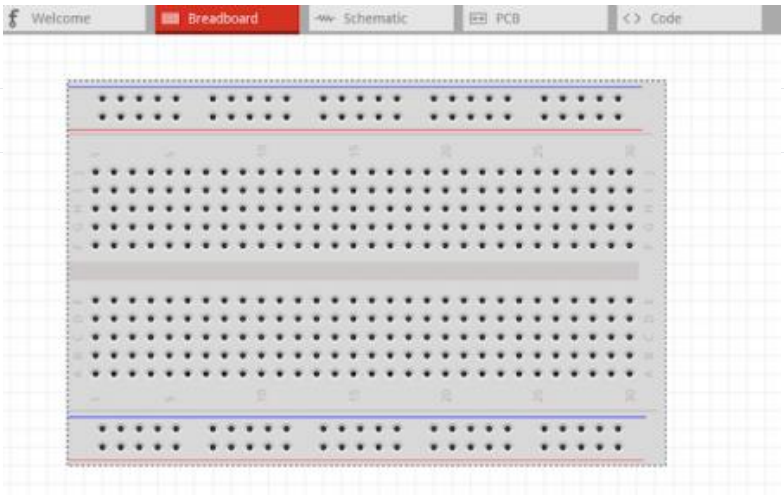
What is Fritzing?

Fritzing is a free design program that can be used to design circuits on breadboards, create schematics, and even develop PCBs. Unlike a lot of other circuit design suites, Fritzing is polished and heavily uses vectorized images, producing beautiful looking circuits.

While Fritzing includes many popular components such as the Arduino range of boards and modules, it is geared for hobby and maker use rather than professional use. Along with the free design features, there is also a community forum and a projects page where users can upload their projects.

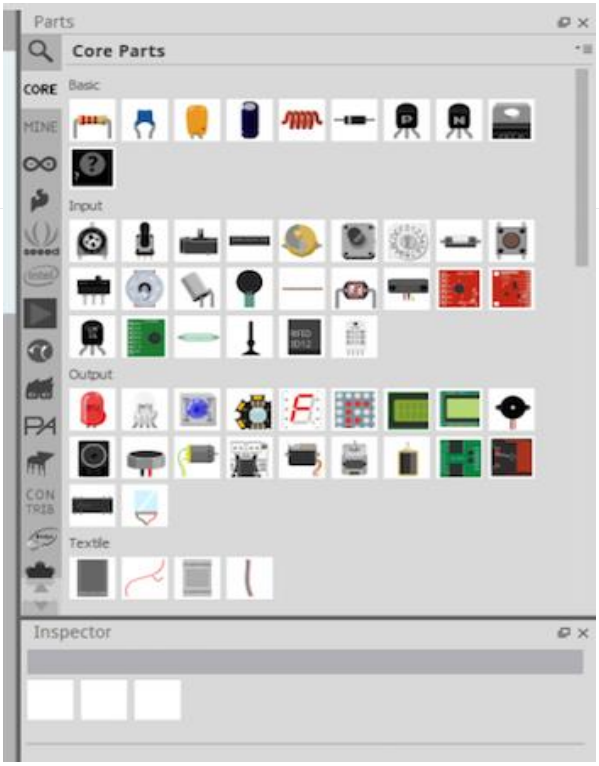
Designing a Project on a Breadboard:

The first main window in the Fritzing environment is the breadboard view which allows the creation of breadboard circuits. The breadboard size can be adjusted, and the set up is interactive, allowing users to drop components and wires to individual pins.



Breadboard view in Fritzing

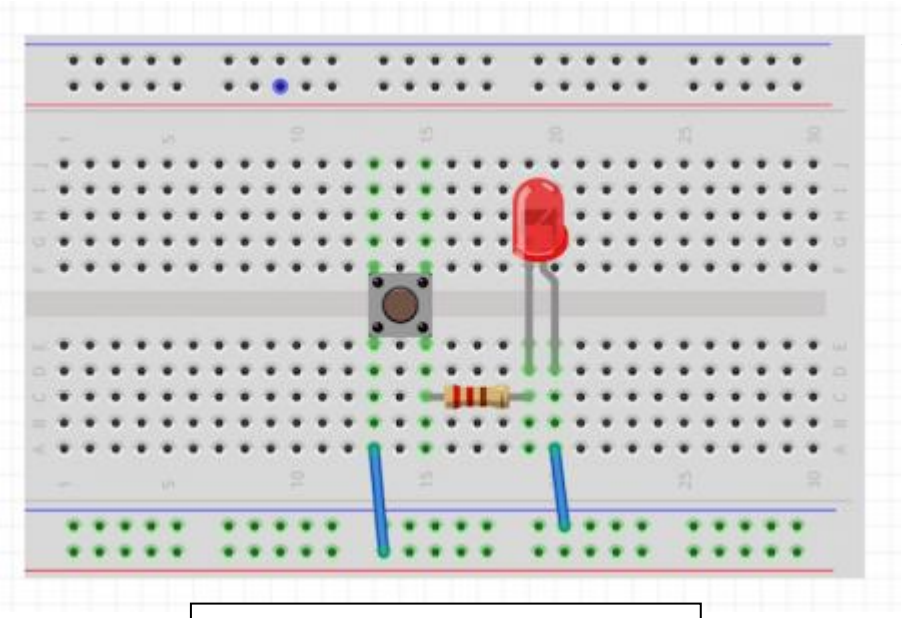
The window on the right-hand side of the screen allows the user to find and place a variety of components onto the breadboard and a search makes finding specific parts easier.



Component pick and place

Unlike more advanced circuit suites (such as KiCad), the list of parts available is limited to the most commonly used modules and microcontrollers (for example, there are plenty of Arduino boards but no PIC devices).

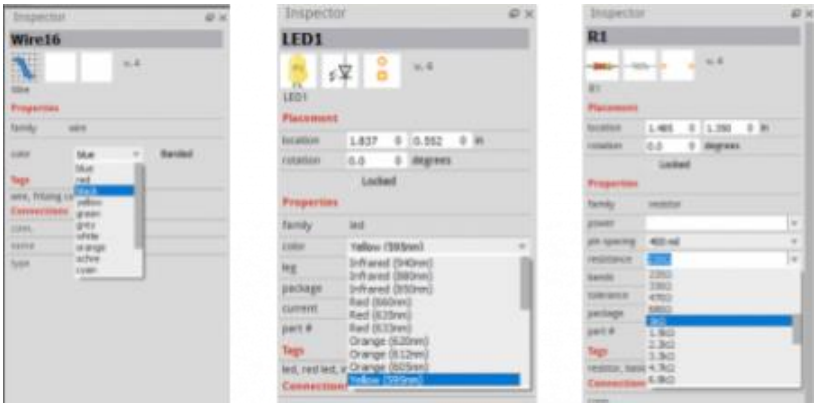
Parts from the search window are dragged into the breadboard area and should snap into holes as the legs cross them. When a part is dropped into place, the rows that are electrically connected to that component's legs show up. This helps to make identifying rows easier, and wires placement clearer. Wires can be made by click and holding— they are also snapped to holes on the breadboard.



Inspector windows in Fritzing

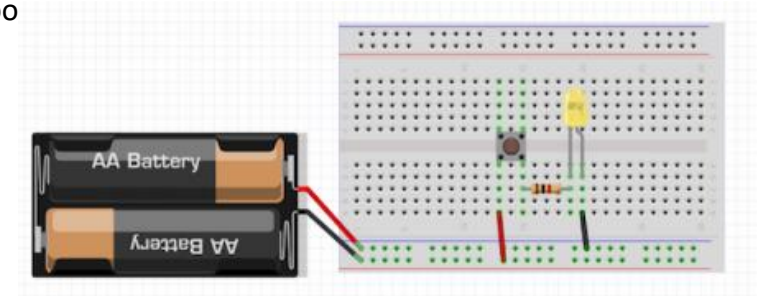
Users can select wires and change their color as needed. Changing the color of wires is very good practice with designs that heavily rely on wires.

The inspector window, which is directly below the component search window, can be used to edit components including the breadboard.



Inspector windows in Fritzing

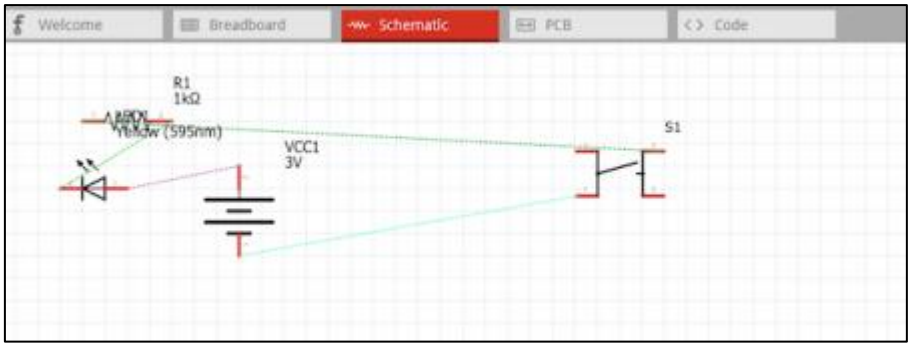
When the options above are applied to our simple LED circuit we can see that the LED is now yellow, the wires color-coded red (for power), and black (for ground), and the resistor is set to 1K. A battery was also included as a source of po



Color-coded circuit design

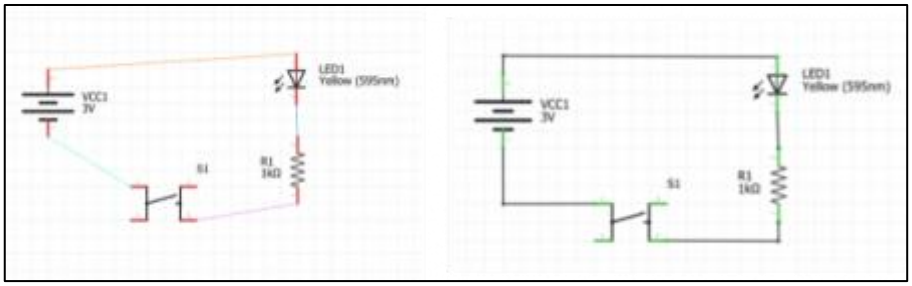
Converting Circuits into Schematics:

Once a circuit is created on the breadboard, it can then be converted into a schematic. One of the really cool features in Fritzing is that as you construct your breadboard circuit, it automatically adds parts into the schematic and forms connections between those parts according to their placement on the breadboard. The schematic, however, will not be neat or complete and requires some rearranging of the components to form the connections.



The schematic window in Fritzing

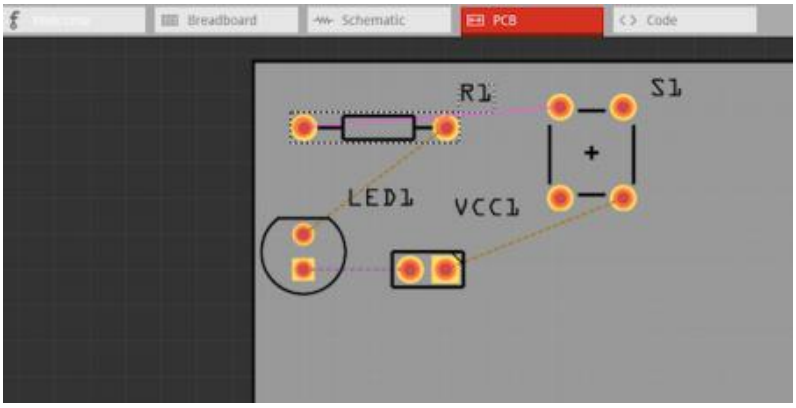
Dotted lines indicate connections between components. Components can be dragged around by clicking and holding. Parts are easily rotated by clicking the part and the pressing Ctrl + R. Wires can be added by clicking the dotted lines. Dotted lines in schematic view indicate connections between components.



Creating PCB Layouts:

The PCB layout is the last stage of creating a circuit in Fritzing. Like the schematic stage, the parts on the PCB are automatically placed on the board. Again, the user is required to orient and position these components.

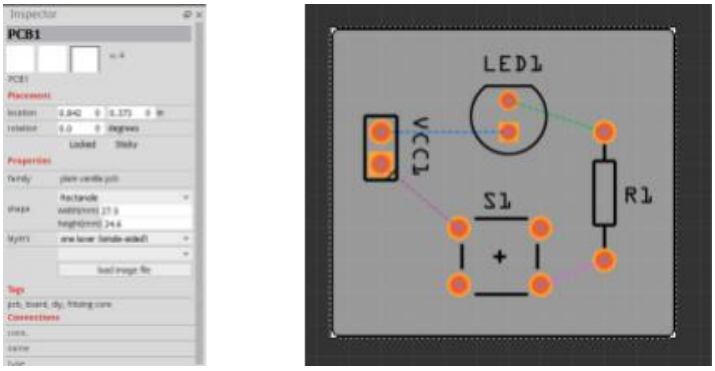
The PCB stage is optional because not every project is going to use a PCB. One of the major advantages of using Fritzing is that the software works in conjunction with the Fritzing PCB service which easily turns a design on Fritzing into a low-cost PCB.



The PCB design window in Fritzing

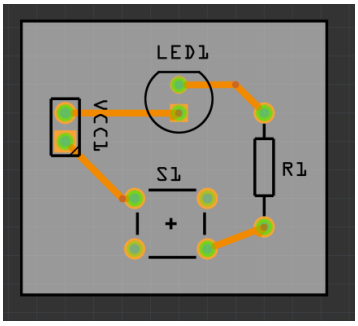
Just like in the schematic layout, components and their neighboring indents can be dragged and rotated. The PCB itself can also be easily resized by dragging the corners. The inspector window provides additional options such as the number of layers on the PCB and the specific shape (such as an Arduino shield shape).

In the example below, the PCB has been shrunk down and set to be a single-sided PCB so that there are only copper traces on the underside.



A single-sided PCB design alongside its settings in the Inspector window

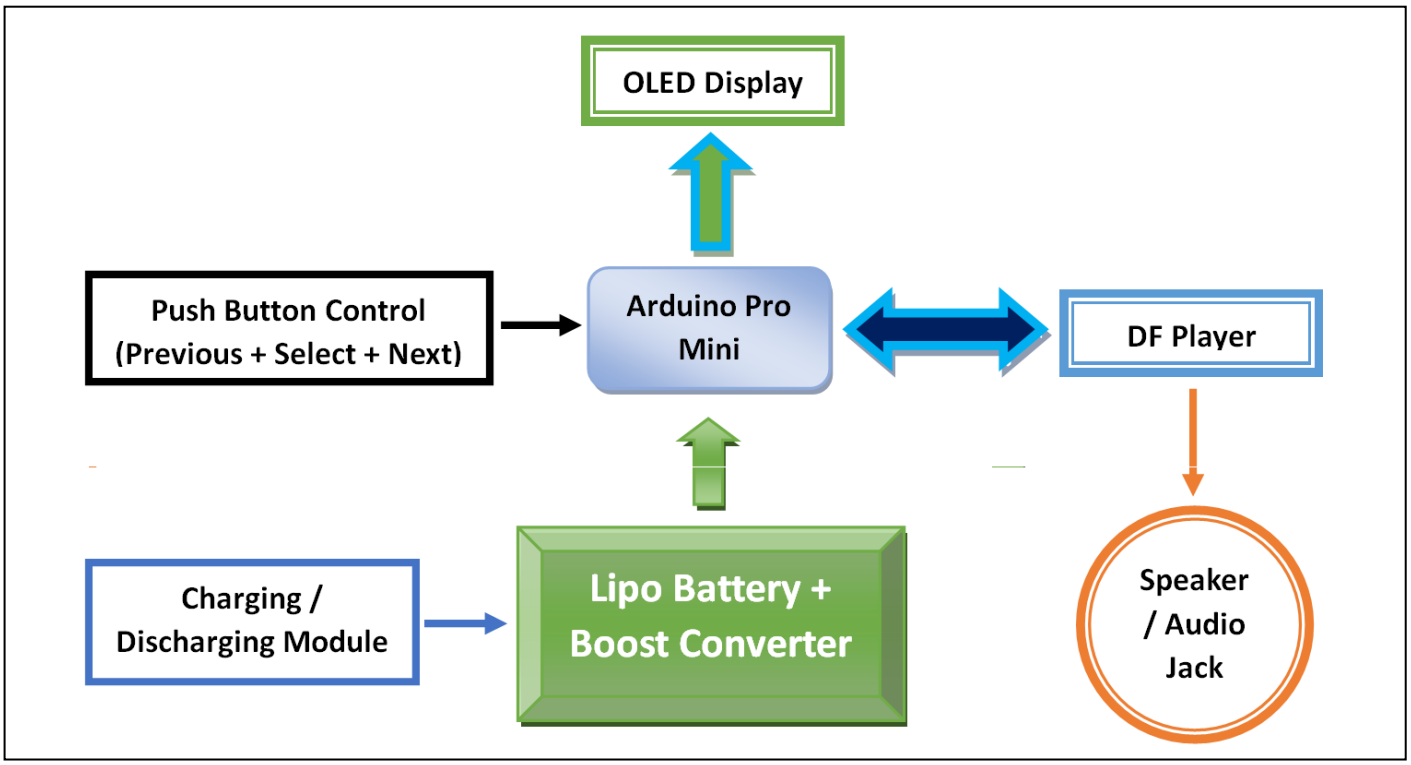
Copper tracks can be routed either by double-clicking the dotted line or clicking and dragging from the pads. If manual routing proves too difficult, Fritzing also comes with a simple auto-router that can route your PCBs automatically.



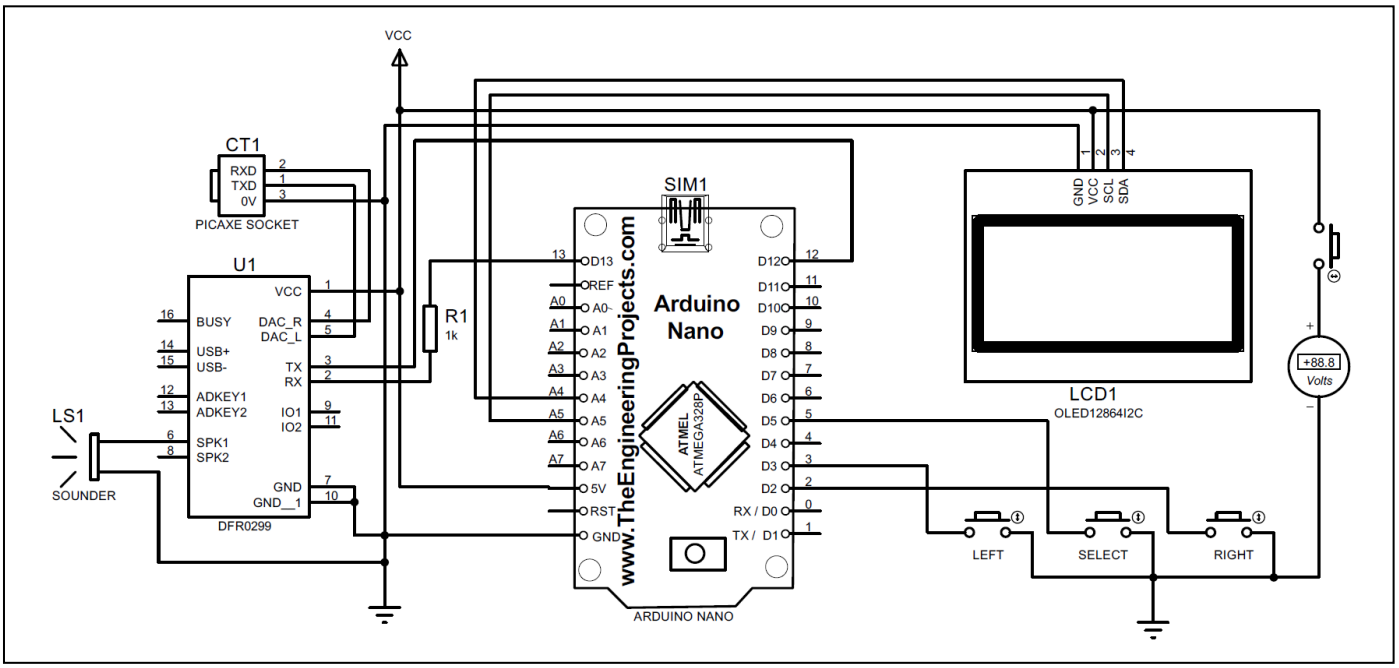
Orange lines represent copper traces in the PCB design window.

4.4 Overall System

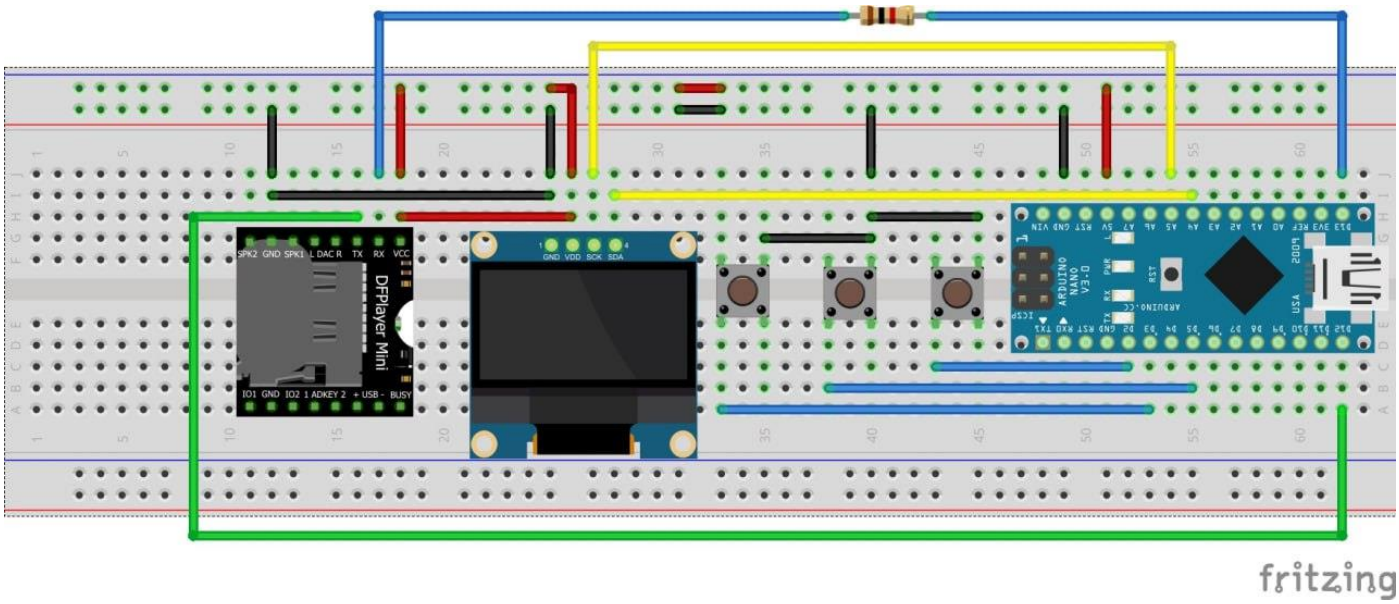
Block Diagram:



Circuit Diagram:



Breadboard Assembly (Fritzing):



Project Description:

This project is all about a portable mp3 player which is based upon a Arduino microcontroller. This tiny mp3 player has some incredible features –

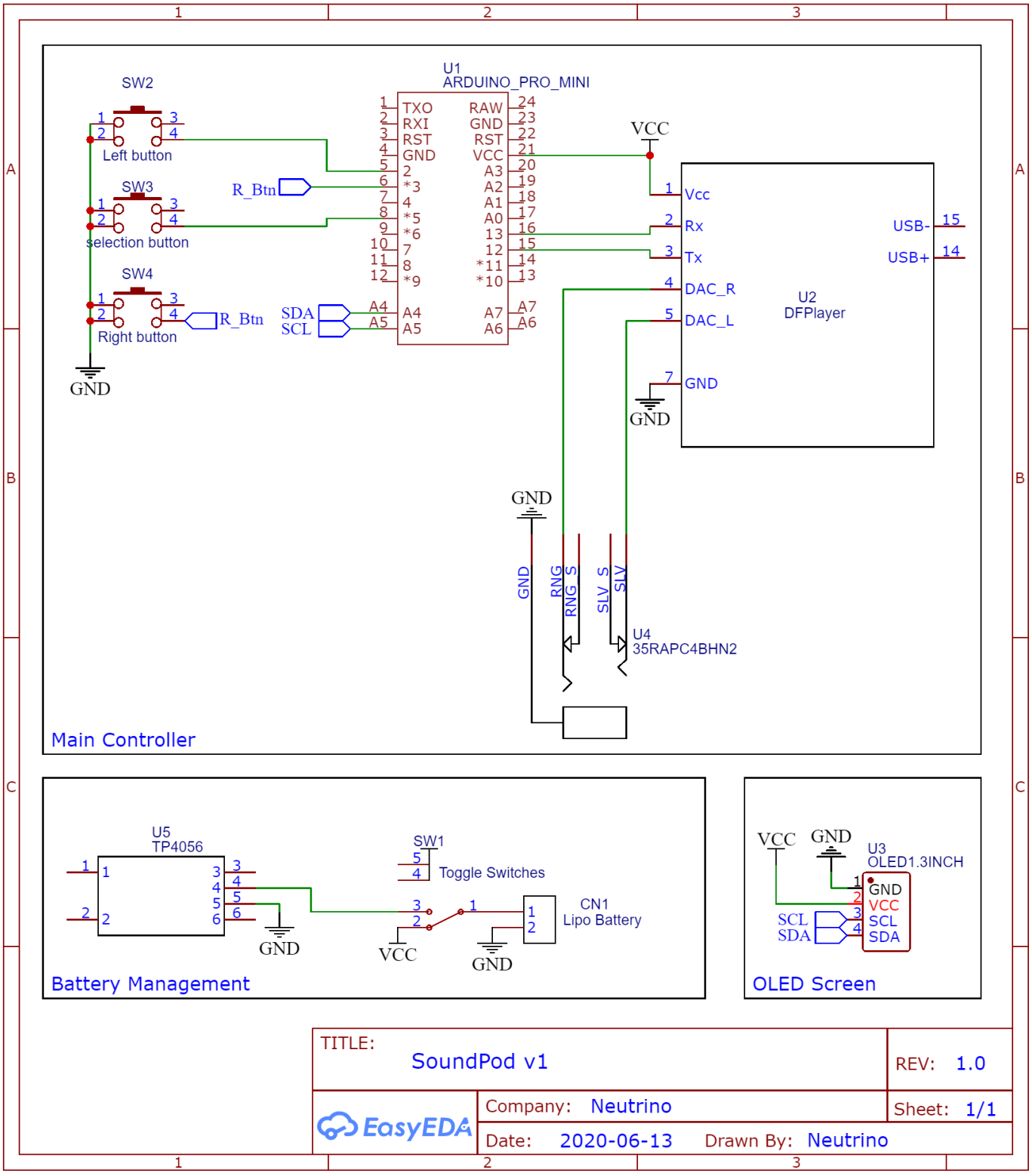
1. It has a OLED display for visual control. By the display you can see the control volume, EQ, Previous track, Next track, Play and Pause the music.
2. External memory card supports up to 32 GB.
3. Live Battery level monitoring using digital voltmeter.
4. Transparent and durable frame.
5. 430 MAH Lipo battery for backup up to 3 hours
6. 3.5mm Headphone jack
7. Type C Charging port
8. Can be repaired easily

Using this mp3 player one can listen to music via headphone from the SD card that helps to store the favourite music files. There are even 6 Equaliser modes that come with this player. Those are –

1. Mode 0 : Normal
2. Mode 1 : Pop
3. Mode 2 : Rock
4. Mode 3 : Jazz
5. Mode 4 : Classic
6. Mode 5 : Bass

By selecting proper EQ mode from the display, One can listen to his / her favourite songs with their favourite taste.

PCB Schematic (Easy EDA):

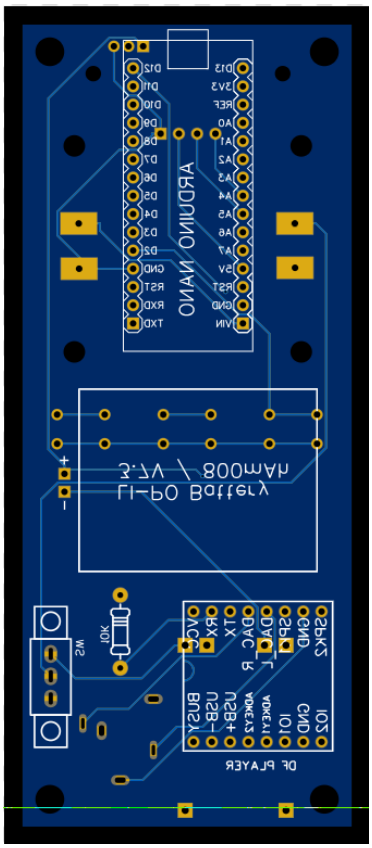


Working principle:

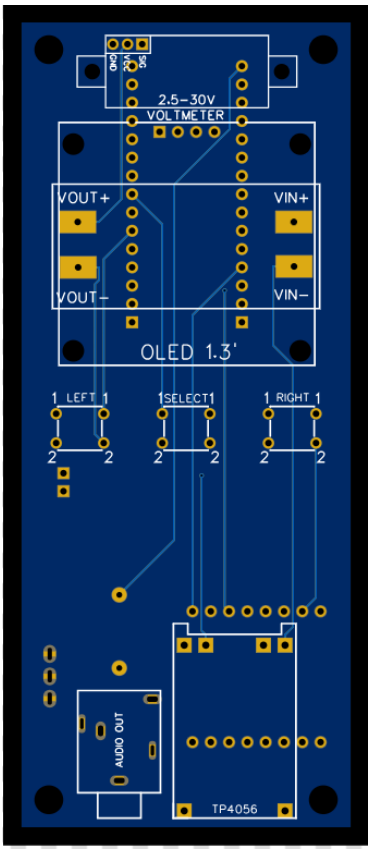
The internal working principle has been described step by step –

- ✓ Step 1 → When the Arduino gets power then it powers all the accessories.
- ✓ Step 2 → OLED Screen gets the Clock from Arduino via 'SCL' pin to run its own digital circuitry.
- ✓ Step 3 → Now data is provided to the OLED screen via 'SDA' pin and OLED screen boots up.
- ✓ Step 4 → Now Arduino sends serial signals TX & RX to the DF Player to check whether there is memory / there are mp3 files in the SD card or not.
- ✓ Step 5 → If SD Card is present then Arduino gets an Acknowledgement signal via TX pin of DF player.
- ✓ Step 6 → Then Arduino triggers DF player and display together and music plays from the DF player via a 3.5 mm Earphone.
- ✓ Step 7 → But if No SD card is inserted or Negative acknowledgement is received to the Arduino then Arduino stalls.
- ✓ Step 8 → Now during playing any button, then an interrupt is received to the Arduino. Then immediately Arduino stops executing the main program and attend that interrupt and executes the corresponding programmes and simultaneously do the serial communication with the OLED and DF player.
- ✓ Step 9 → Battery voltage is monitored from the DC voltmeter.
- ✓ Step 10 → Boost converter is used to power up the Arduino Nano boosting the battery voltage from 3.7V to 9V.

PCB View:



Bottom View



Top View

Code:

[Folder : Soundpod]

```
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <DFRobotDFPlayerMini.h>
#include <U8g2lib.h>
#include <Wire.h>
#include <EEPROM.h>
#include "OneButton.h"

// 65 - Audio Settings Icon
// 66 - Flash screen Spectrum
// 67 - Battery empty
// 68 - Battery full
// 69 - left adjustment button
// 70 - right adjustment button
// 71 - back button
// 72 - pause button
// 73 - play button
// 74 - previous button
// 75 - Next button
// 76 - Music Icon
// 77 - volume full
// 78 - volume half
// 79 - volume low

extern const uint8_t soundpod_icon_pack_font[350] U8G2_FONT_SECTION("soundpod_icon_pack_font") =
  "\17\0\4\4\5\6\3\3\7\34 \0\0\16\0\16\0\0\0\0\0\1AA&p\341\241\22\64\214\310"
  "!BE\210\11\24\206\210 !D\4\11!\42H\10\21ABD\10\25\42TL\320 \0BZ\34"
  "$\301\17\202<H\362 \311\203$\17\202\20y\20\204\310\203 D\36\4!\362 \10QH\210BB"
  "\24\22\42\204\210\42D\24!\242\10\21E\210(BD\21\42\212\20\321\220\20\205\204($D!!\12"
  "\11\221\7A\210<\10B\344A\20\42\17\222<H\362 \311\203$\10\0C\17\310`\221p\42T\210"
  "PS\205\70\1\0D\14\310`\221p\342\304\3\24'\0E\12\303x\221\22\2\205\220\0F\13\303p"
  "\221\20D\4\212 \0G\16\250\240\221\21(\304\220\21fH\211\1H\12\214\261\241@\10\377G\4I"
  "\24\213\261\241\20t`\261CI\36<\10r\250\330\300\240\0J\14\307`\221 h\14\11k\10\5K"
  "\15\307`\221\20\210\214\11\23d\6\11L\35\20\42\241yj\21G\304\4\11\24$P\220@A\242\12"
  "\11\62Q\210W$\314\222\5M\24\10!\221\23L\204\10\42&B\224\10QD\12A\201\0N\16\6"
  ")\221\23H\10\25\25\211\12\2\0O\13\4\61\221\23\344\301\20\61\1\0\0\0\4\377\377\0";

const uint8_t leftButtonPin = 2;
const uint8_t selectionButtonPin = 5;
const uint8_t rightButtonPin = 3;

volatile uint8_t sMenuSelection = 2;
volatile uint8_t selection = 1;
volatile bool updateScreen = true;
```

```
// Variables
uint8_t filecounts;                // total number of files in current folder
uint8_t foldercounts;              // total number of folders on sd-card

uint8_t volume = 20;               // current volume (0 .. 30)
uint8_t folder = 1;                // current sd-card folder
uint8_t file   = 1;                // curent file in current folder
uint8_t eq = 0;

uint8_t batteryLevel = 0;
uint8_t mins = 0;
unsigned long lastBatteryRead = 0;

boolean playing = false;
boolean inSideMenuSelection = true;

OneButton PreviousBTN(leftButtonPin, true);
OneButton PlayBTN(selectionButtonPin, true);
OneButton NextBTN(rightButtonPin, true);

SoftwareSerial customSoftwareSerial(12,13);
DFRobotDFPlayerMini myDFPlayer;

U8G2_SH1106_128X64_NONAME_1_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE);

void setup(void) {

 //being hardware serial for debugging
 Serial.begin(9600);

// Initializing u8g2 library
 u8g2.begin();
 u8g2.firstPage();
 do{

   flashPage();

 }while(u8g2.nextPage());

 PreviousBTN.attachClick(preivousButtonClicked);
 PlayBTN.attachClick(playButtonClicked);
 NextBTN.attachClick(nextButtonClicked);

 PreviousBTN.setDebounceMs(50);
 PlayBTN.setDebounceMs(50);
 NextBTN.setDebounceMs(50);

 //Initializing software serial
 customSoftwareSerial.begin(9600);
```

```
 if (!myDFPlayer.begin(customSoftwareSerial)) {  //Use softwareSerial to communicate with mp3.
  Serial.println(F("Please insert the SD card!"));
 }
 volume = EEPROM.read(0);
 if(volume > 30)
 volume = 30;

 eq     = EEPROM.read(1);
 if(eq > 5)
 eq = 5;

 file   = EEPROM.read(2);
 if(file >= 255)
 file = 1;

 delay(1000);
 myDFPlayer.volume(volume);  //Set volume value. From 0 to 30
 delay(500);
 foldercounts = myDFPlayer.readFolderCounts();
 startFolderPlay();
//
}

void loop() {

 //Check constantly for button inputs
 PreviousBTN.tick();
 PlayBTN.tick();
 NextBTN.tick();
 //Update the display as required
 updateDisplay();
 updateDFplayer();

}
```

[Folder : Input]

```
void preivousButtonClicked()
{

 updateScreen = true;
 if(inSideMenuSelection && sMenuSelection < 2)
 {
  sMenuSelection++;
 }
 else if(selection < 4 && sMenuSelection == 1)
 {
```

```
    selection++;
  }
  else if(selection == 1 && volume < 30 &&sMenuSelection == 2 && !inSideMenuSelection)
  {
    volume++;
  }
  else if(selection == 2 && eq < 5 && sMenuSelection == 2 && !inSideMenuSelection)
  {
    eq++;
  }

}

void nextButtonClicked()
{
  updateScreen = true;
  if(inSideMenuSelection && sMenuSelection > 1)
  {
   sMenuSelection--;
  }
  else if(selection > 1 && sMenuSelection == 1)
  {
   selection--;
  }
  else if(selection == 1 && volume > 0 &&sMenuSelection == 2)
  {
    volume--;
  }
  else if(selection == 2 && eq > 0 && sMenuSelection == 2)
  {
    eq--;
  }
}

void playButtonClicked()
{
  //Selection button
  if(inSideMenuSelection)
  {
     inSideMenuSelection = false;
     updateScreen = true;
     delay(100);
  }
  else if(!inSideMenuSelection && sMenuSelection == 1)
  {
      if(selection == 1)
      {
       if(file > 1)
       {
```

```
      //previous audio
      myDFPlayer.previous();
      file--;
      if(!playing)
        playing = true;
      EEPROM.write(2, file);
     }
    }
    else if(selection == 2)
    {
     //pause / play
     if(playing)
     {
      myDFPlayer.pause();
     }
     else
     {
      myDFPlayer.start();
     }
     playing = !playing;
    }
    else if(selection == 3)
    {
      //next audio
      file++;
       myDFPlayer.next();
      if(!playing)
       playing = true;
       EEPROM.write(2, file);
    }
    else if(selection == 4)
    {
     //back to side menu
     selection = 1;
     inSideMenuSelection = true;
    }
    updateScreen = true;
    delay(200);
}else if(!inSideMenuSelection && sMenuSelection == 2)
{
  if(selection == 1)
  {
     selection = 2;
     myDFPlayer.volume(volume);
     EEPROM.write(0, volume);
  }
  else if(selection == 2)
  {
    selection = 4;
```

```
      myDFPlayer.EQ(eq);
      EEPROM.write(1, eq);
    }
    else if(selection == 4)
    {
       //back to side menu
       selection = 1;
       inSideMenuSelection = true;
    }
    updateScreen = true;
    delay(200);
  }

}
```

[Folder : Audio]

```
// starts to play the actual file in the actual folder
void startFolderPlay() {
  filecounts = myDFPlayer.readFileCountsInFolder(folder);
  myDFPlayer.playFolder(folder, file);
  playing = false;
}

void updateDFplayer()
{
   // check player status
  if (myDFPlayer.available()) {
    uint8_t type = myDFPlayer.readType();
    int value = myDFPlayer.read();

    switch (type) {
      case DFPlayerPlayFinished:
        if (file < filecounts) {
          file++;
          myDFPlayer.playFolder(folder, file);
          EEPROM.write(2, file);
          updateScreen = true;
        }
        break;
      default:
        break;
    }
  }


}
```

[Folder : Display]

```
void updateDisplay() {
  //Updating the dispaly
  if (updateScreen) {
    u8g2.firstPage();
    do {
      int ch = (sMenuSelection);
      switch (ch) {
        case 1:
          player();
          break;
        case 2:
          settings();
          break;
        default:
          Serial.println(F("Default Screen"));
      }
      updateScreen = false;
    } while (u8g2.nextPage());
  }
}

void settings() {
  sideMenu();
  topMenu();
  u8g2.setFont(u8g2_font_glasstown_nbp_tf);

  u8g2.setCursor(65, 17);
  u8g2.print(F("Setting"));

  u8g2.setFontMode(0);

  u8g2.setCursor(47, 40);
  u8g2.print(F("Volume"));
  if (selection == 1 && !inSideMenuSelection) {
    u8g2.setFont(soundpod_icon_pack_font);
    if (volume > 0)
      u8g2.drawGlyph(85, 40, 69); // drawing left ajustment button next to the volume
    if (volume < 30) {
      if (volume < 10)
        u8g2.drawGlyph(100, 40, 70); // drawing right ajustment button next to the volume for single digit
volume
      else
        u8g2.drawGlyph(105, 40, 70);  // drawing right ajustment button next to the volume for double digit
volume
    }
  }
  u8g2.setFont(u8g2_font_glasstown_nbp_tf);
```

```
  u8g2.setCursor(95, 40);
  u8g2.print(volume);
  u8g2.setDrawColor(1);

  u8g2.setCursor(67, 60);
  u8g2.print(F("EQ"));
  if (selection == 2 && !inSideMenuSelection) {
    u8g2.setFont(soundpod_icon_pack_font);
    if (eq > 0)
      u8g2.drawGlyph(85, 60, 69); // drawing left ajustment button next to the volume
    if (eq < 5)
      u8g2.drawGlyph(100, 60, 70); // drawing right ajustment button next to the volume
  }
  u8g2.setFont(u8g2_font_glasstown_nbp_tf);
  u8g2.setCursor(95, 60);
  u8g2.print(eq);
  u8g2.setDrawColor(1);

  if (selection == 4 && !inSideMenuSelection) {
    u8g2.setDrawColor(0);
  }
  u8g2.setFont(soundpod_icon_pack_font);
  u8g2.drawGlyph(120, 60, 71); // drawing back button to the bottom off the screen
  u8g2.setDrawColor(1);
}

void flashPage() {
  drawIcon(soundpod_icon_pack_font, u8g2.getDisplayWidth() / 2 - 12, u8g2.getDisplayHeight() - 22, 66);
//drawing spectrum for splash screen
  u8g2.setFont(u8g2_font_glasstown_nbp_tf);
  u8g2.setCursor(50, 63);
  u8g2.print(F("Soundpod"));
}

void topMenu() {
  u8g2.setFont(soundpod_icon_pack_font);

  if (volume >= 25)
    u8g2.drawGlyph(119, 9, 77);
  else if (volume > 10 && volume < 25)
    u8g2.drawGlyph(119, 9, 78);
  else if (volume <= 10)
    u8g2.drawGlyph(119, 9, 79);
  //
  //    u8g2.setFont(u8g2_font_glasstown_nbp_tf);
  //    u8g2.setCursor(25,9);
  //    u8g2.print(batteryLevel);
  //    u8g2.setCursor(35,9);
  //    u8g2.print("%");
```

```
}

void sideMenu() {
 const uint8_t menuListGlpy[2] = { 77, 64 };

 u8g2.setFontMode(0);

 //Audio player selection
 if (abs(sMenuSelection) == 1 && inSideMenuSelection) {
  u8g2.drawRBox(0, 12, 20, 21, 3);
  u8g2.setDrawColor(0);
 }

 //Audio player
 drawIcon(soundpod_icon_pack_font, 2, 30, 76);

 u8g2.setDrawColor(1);

 //setting Selection
 if (abs(sMenuSelection) == 2 && inSideMenuSelection) {
  u8g2.drawRBox(0, 33, 20, 19, 3);
  u8g2.setDrawColor(0);
 }
 //settings
 drawIcon(soundpod_icon_pack_font, 2, 52, 65);
 //    u8g2.setFont(menuList[1][1]);
 //    u8g2.drawGlyph(2,52,menuList[1][0]);

 u8g2.setDrawColor(1);
 u8g2.drawLine(22, 0, 22, 68);
}

void player() {
 sideMenu();
 topMenu();

 u8g2_uint_t midOriginX = 64;
 u8g2_uint_t midOriginY = 44;

 u8g2.setFontMode(0);
 u8g2.setCursor(45, 25);
 u8g2.setFont(u8g2_font_glasstown_nbp_tf);
 u8g2.print("Track : ");
 u8g2.setCursor(78, 25);
 u8g2.print(file);
 u8g2.setCursor(88, 25);
 u8g2.print('/');
 u8g2.setCursor(95, 25);
 u8g2.print(filecounts);
```

```
 if (selection == 1 && !inSideMenuSelection) {
  u8g2.drawRBox(midOriginX - 7, midOriginY - 5, 11, 9, 2);
  u8g2.setDrawColor(0);
 }
 u8g2.setFont(soundpod_icon_pack_font);
 u8g2.drawGlyph(midOriginX - 5, midOriginY + 4, 74);
 u8g2.setDrawColor(1);

 if (selection == 2 && !inSideMenuSelection) {
  u8g2.drawRBox(midOriginX + 7.5, midOriginY - 8.5, 16.5, 15, 3);
  u8g2.setDrawColor(0);
 }
 u8g2.setFont(soundpod_icon_pack_font);
 if (playing) {
  u8g2.drawGlyph(midOriginX + 7.5, midOriginY + 7.5, 72);
 } else {
  u8g2.drawGlyph(midOriginX + 7.5, midOriginY + 7.5, 73);
 }
 u8g2.setDrawColor(1);

 if (selection == 3 && !inSideMenuSelection) {
  u8g2.drawRBox(midOriginX + 25, midOriginY - 5, 12, 9, 2);
  u8g2.setDrawColor(0);
 }
 u8g2.setFont(soundpod_icon_pack_font);
 u8g2.drawGlyph(midOriginX + 27, midOriginY + 4, 75);
 u8g2.setDrawColor(1);

 if (selection == 4 && !inSideMenuSelection) {
  u8g2.setDrawColor(0);
 }
 u8g2.setFont(soundpod_icon_pack_font);
 u8g2.drawGlyph(120, 60, 71);
 u8g2.setDrawColor(1);
}

void drawIcon(const uint8_t* iconName, u8g2_uint_t x, u8g2_uint_t y, uint16_t glyph) {
 u8g2.setFont(iconName);
 u8g2.drawGlyph(x, y, glyph);
}
```

Chapter 4 : System Testing and Result

After completion of circuit assembly and before soldering the components on the PCB it is highly recommended to test the circuit individually. Now from now we will concentrate on the individual testing of components first. Then we will test the total circuit and will see that whether the out is desired or not.

5.1 Individual testing

First of all the discrete components will be tested individually. Hence we will be familiar with all these components and also it will be checked that whether all the components are working fine or not. Now let's check one by one.

5.1.1 Testing of Arduino Nano

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P. It offers the same connectivity and specs of the UNO board in a smaller form factor.

The Arduino Nano is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards and running both online and offline. For more information on how to get started with the Arduino Software visit the Getting Started page.

Use your Arduino Nano on the Arduino Web IDE

All Arduino boards, including this one, work out-of-the-box on the Arduino Web Editor, you only need to install Arduino Create Agent to get started.

The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow this simple guide to start coding on the browser and upload your sketches onto your board.

Use your Arduino Nano on the Arduino Desktop IDE

If you want to program your Arduino Nano while offline you need to install the Arduino Desktop IDE To connect the Arduino Nano to your computer, you'll need a Mini-B USB cable. This also provides power to the board, as indicated by the blue LED (which is on the bottom of the Arduino Nano 2.x and the top of the Arduino Nano 3.0).

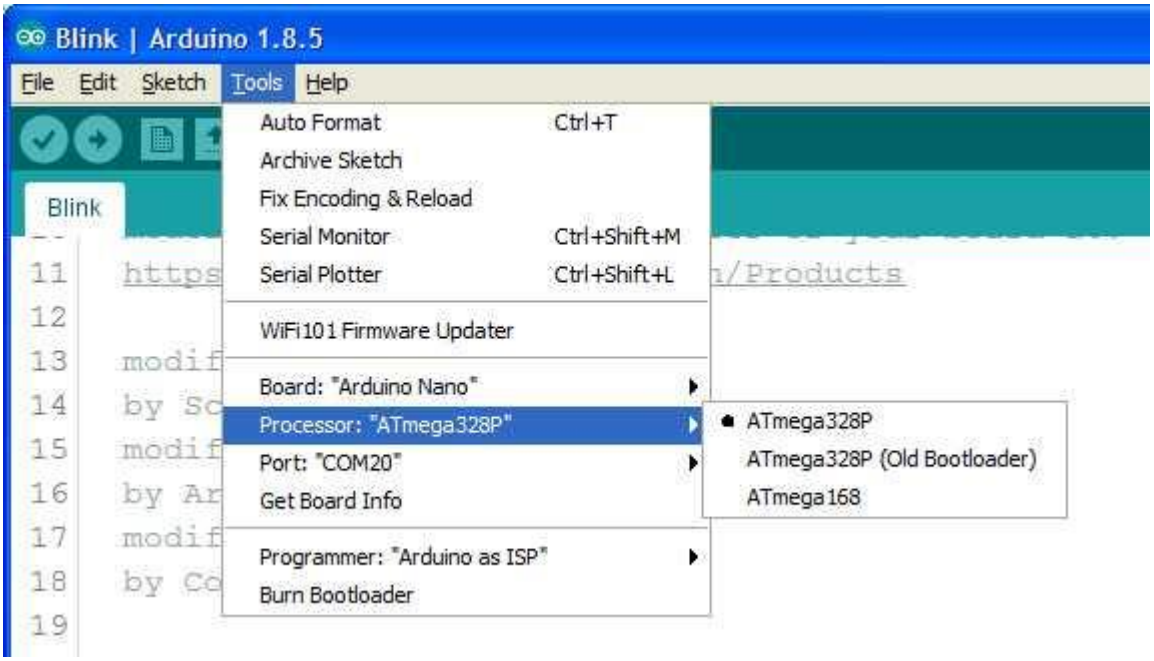
Open your first sketch

Open the LED blink example sketch: **File > Examples > 01.Basics > Blink**.

Select your board type and port

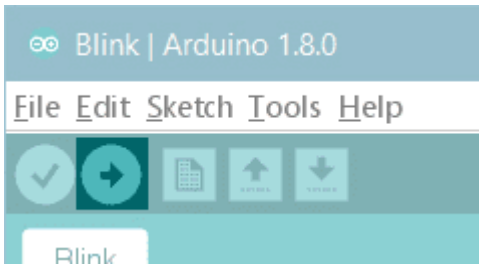
Select **Tools > Board > Arduino AVR Boards > Arduino Nano**.

NOTE: We have updated the Nano board with a fresh bootloader. Boards sold by us from January 2018 have this new bootloader, while boards manufactured before that date have the old bootloader. First, check that **Tools > Board > Boards Manager** shows you have the Arduino AVR Boards 1.16.21 or later installed. Then, to program the NEW Arduino NANO boards you need to chose **Tools > Processor > ATmega328P**. To program old boards you need to choose **Tools > Processor > ATmega328P (Old Bootloader)**. If you get an error while uploading or you are not sure which bootloader you have, try each **Tools > Processor** menu option until your board gets properly programmed.



Upload and Run your first Sketch

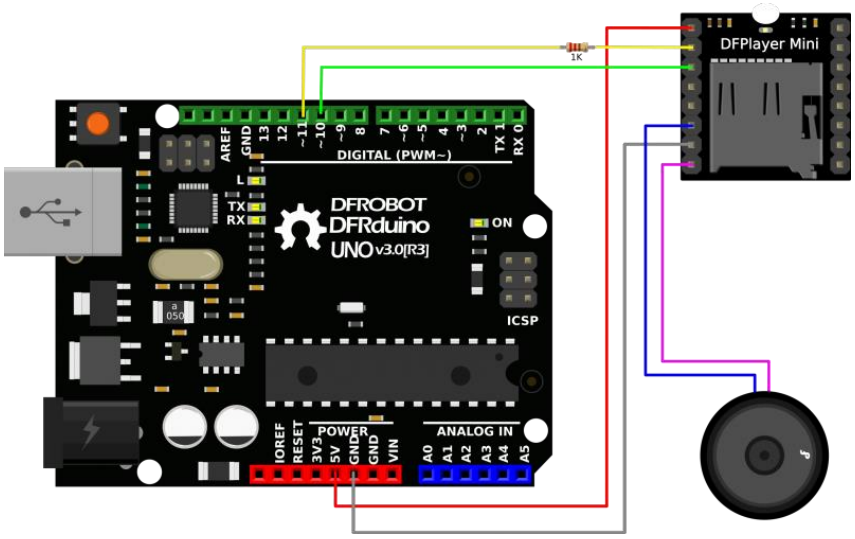
To upload the sketch to the Arduino Nano, click the **Upload** button in the upper left to load and run the sketch on your board:



Wait a few seconds - you should see the RX and TX LEDs on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.

5.1.2 Testing of DF Player Mini

Connection Diagram



Note: For simple use ,the upper diagram is ready,But if you find the noise is quite loud, then you could attach an 1K resistor to the TX pin.

Copy your mp3 into you micro SD card

NOTE: The order you copy the mp3 into micro SD card will affect the order mp3 played , which means play(1) function will play the first mp3 copied into micro SD card.

For Mac User

NOTE: If you are using Mac OS X to copy the mp3, the file system will automatically add hidden files like: "._0001.mp3" for index, which this module will handle as valid mp3 files. It is really annoying. So you can run following command in terminal to eliminate those files.

dot_clean /Volumes/<SDVolumeName>

Please replace the to the volume name of your SD card.

Sample Code

We've created an Arduino library for DFPlayer Mini to simplify the method for you to make it work. Connect the hardware as the picture above shown and play with the sample code. You can download the latest library here: DFRobotDFPlayerMini.

Sample code "GetStarted", switching to next song every 3 seconds

```
/**************************************************
DFPlayer - A Mini MP3 Player For Arduino
 <https://www.dfrobot.com/product-1121.html>

 **************************************************
 This example shows the basic function of library for DFPlayer.

 Created 2016-12-07
 By [Angelo qiao](Angelo.qiao@dfrobot.com)

 GNU Lesser General Public License.
 See <http://www.gnu.org/licenses/> for details.
 All above must be included in any redistribution
 **************************************************/

/***********Notice and Trouble shooting***************
 1.Connection and Diagram can be found here
 <https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299#Connection_Diagram>
 2.This code is tested on Arduino Uno, Leonardo, Mega boards.
 **************************************************/

#include "Arduino.h"
#include "DFRobotDFPlayerMini.h"
```

```
#if (defined(ARDUINO_AVR_UNO) || defined(ESP8266))   // Using a soft serial port
#include <SoftwareSerial.h>
SoftwareSerial softSerial(/*rx =*/4, /*tx =*/5);
#define FPSerial softSerial
#else
#define FPSerial Serial1
#endif

DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

void setup()
{
#if (defined ESP32)
  FPSerial.begin(9600, SERIAL_8N1, /*rx =*/D3, /*tx =*/D2);
#else
  FPSerial.begin(9600);
#endif

  Serial.begin(115200);

  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

  if (!myDFPlayer.begin(FPSerial, /*isACK = */true, /*doReset = */true)) {  //Use serial to communicate with
mp3.
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while(true){
      delay(0); // Code to compatible with ESP8266 watch dog.
    }
  }
  Serial.println(F("DFPlayer Mini online."));

  myDFPlayer.volume(10);  //Set volume value. From 0 to 30
  myDFPlayer.play(1);  //Play the first mp3
}

void loop()
{
  static unsigned long timer = millis();

  if (millis() - timer > 3000) {
    timer = millis();
    myDFPlayer.next();  //Play next mp3 every 3 second.
  }
```

```
  if (myDFPlayer.available()) {
    printDetail(myDFPlayer.readType(), myDFPlayer.read()); //Print the detail message from DFPlayer to
handle different errors and states.
  }
}

void printDetail(uint8_t type, int value){
  switch (type) {
    case TimeOut:
      Serial.println(F("Time Out!"));
      break;
    case WrongStack:
      Serial.println(F("Stack Wrong!"));
      break;
    case DFPlayerCardInserted:
      Serial.println(F("Card Inserted!"));
      break;
    case DFPlayerCardRemoved:
      Serial.println(F("Card Removed!"));
      break;
    case DFPlayerCardOnline:
      Serial.println(F("Card Online!"));
      break;
    case DFPlayerUSBInserted:
      Serial.println("USB Inserted!");
      break;
    case DFPlayerUSBRemoved:
      Serial.println("USB Removed!");
      break;
    case DFPlayerPlayFinished:
      Serial.print(F("Number:"));
      Serial.print(value);
      Serial.println(F(" Play Finished!"));
      break;
    case DFPlayerError:
      Serial.print(F("DFPlayerError:"));
      switch (value) {
        case Busy:
          Serial.println(F("Card not found"));
          break;
        case Sleeping:
          Serial.println(F("Sleeping"));
          break;
        case SerialWrongStack:
          Serial.println(F("Get Wrong Stack"));
          break;
        case CheckSumNotMatch:
          Serial.println(F("Check Sum Not Match"));
```

```
        break;
      case FileIndexOut:
        Serial.println(F("File Index Out of Bound"));
        break;
      case FileMismatch:
        Serial.println(F("Cannot Find File"));
        break;
      case Advertise:
        Serial.println(F("In Advertise"));
        break;
      default:
        break;
    }
    break;
  default:
    break;
 }

}
```

Copy

Sample code "FullFunction", including all the functions. Please read the comments and documents in detail

```
/**************************************************
 DFPlayer - A Mini MP3 Player For Arduino
 <https://www.dfrobot.com/product-1121.html>

 ***************************************************
 This example shows the all the function of library for DFPlayer.

 Created 2016-12-07
 By [Angelo qiao](Angelo.qiao@dfrobot.com)

 GNU Lesser General Public License.
 See <http://www.gnu.org/licenses/> for details.
 All above must be included in any redistribution
 ****************************************************/

/***********Notice and Trouble shooting***************
 1.Connection and Diagram can be found here
<https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299#Connection_Diagram>
 2.This code is tested on Arduino Uno, Leonardo, Mega boards.
 ****************************************************/

#include "Arduino.h"
#include "DFRobotDFPlayerMini.h"

#if (defined(ARDUINO_AVR_UNO) || defined(ESP8266))   // Using a soft serial port
```

```
#include <SoftwareSerial.h>
SoftwareSerial softSerial(/*rx =*/4, /*tx =*/5);
#define FPSerial softSerial
#else
#define FPSerial Serial1
#endif

DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

void setup()
{
#if (defined ESP32)
  FPSerial.begin(9600, SERIAL_8N1, /*rx =*/D3, /*tx =*/D2);
#else
  FPSerial.begin(9600);
#endif

  Serial.begin(115200);

  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

  if (!myDFPlayer.begin(FPSerial, /*isACK = */true, /*doReset = */true)) {  //Use serial to communicate with
mp3.
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while(true);
  }
  Serial.println(F("DFPlayer Mini online."));

  myDFPlayer.setTimeOut(500); //Set serial communictaion time out 500ms

  //----Set volume----
  myDFPlayer.volume(10);  //Set volume value (0~30).
  myDFPlayer.volumeUp(); //Volume Up
  myDFPlayer.volumeDown(); //Volume Down

  //----Set different EQ----
  myDFPlayer.EQ(DFPLAYER_EQ_NORMAL);
//  myDFPlayer.EQ(DFPLAYER_EQ_POP);
//  myDFPlayer.EQ(DFPLAYER_EQ_ROCK);
//  myDFPlayer.EQ(DFPLAYER_EQ_JAZZ);
//  myDFPlayer.EQ(DFPLAYER_EQ_CLASSIC);
//  myDFPlayer.EQ(DFPLAYER_EQ_BASS);

  //----Set device we use SD as default----
```

```
//  myDFPlayer.outputDevice(DFPLAYER_DEVICE_U_DISK);
  myDFPlayer.outputDevice(DFPLAYER_DEVICE_SD);
//  myDFPlayer.outputDevice(DFPLAYER_DEVICE_AUX);
//  myDFPlayer.outputDevice(DFPLAYER_DEVICE_SLEEP);
//  myDFPlayer.outputDevice(DFPLAYER_DEVICE_FLASH);

  //----Mp3 control----
//  myDFPlayer.sleep();     //sleep
//  myDFPlayer.reset();     //Reset the module
//  myDFPlayer.enableDAC();  //Enable On-chip DAC
//  myDFPlayer.disableDAC();  //Disable On-chip DAC
//  myDFPlayer.outputSetting(true, 15); //output setting, enable the output and set the gain to 15

  //----Mp3 play----
  myDFPlayer.next();  //Play next mp3
  delay(1000);
  myDFPlayer.previous();  //Play previous mp3
  delay(1000);
  myDFPlayer.play(1);  //Play the first mp3
  delay(1000);
  myDFPlayer.loop(1);  //Loop the first mp3
  delay(1000);
  myDFPlayer.pause();  //pause the mp3
  delay(1000);
  myDFPlayer.start();  //start the mp3 from the pause
  delay(1000);
  myDFPlayer.playFolder(15, 4);  //play specific mp3 in SD:/15/004.mp3; Folder Name(1~99); File
Name(1~255)
  delay(1000);
  myDFPlayer.enableLoopAll(); //loop all mp3 files.
  delay(1000);
  myDFPlayer.disableLoopAll(); //stop loop all mp3 files.
  delay(1000);
  myDFPlayer.playMp3Folder(4); //play specific mp3 in SD:/MP3/0004.mp3; File Name(0~65535)
  delay(1000);
  myDFPlayer.advertise(3); //advertise specific mp3 in SD:/ADVERT/0003.mp3; File Name(0~65535)
  delay(1000);
  myDFPlayer.stopAdvertise(); //stop advertise
  delay(1000);
  myDFPlayer.playLargeFolder(2, 999); //play specific mp3 in SD:/02/004.mp3; Folder Name(1~10); File
Name(1~1000)
  delay(1000);
  myDFPlayer.loopFolder(5); //loop all mp3 files in folder SD:/05.
  delay(1000);
  myDFPlayer.randomAll(); //Random play all the mp3.
  delay(1000);
  myDFPlayer.enableLoop(); //enable loop.
  delay(1000);
  myDFPlayer.disableLoop(); //disable loop.
```

```
  delay(1000);

  //----Read imformation----
  Serial.println(myDFPlayer.readState()); //read mp3 state
  Serial.println(myDFPlayer.readVolume()); //read current volume
  Serial.println(myDFPlayer.readEQ()); //read EQ setting
  Serial.println(myDFPlayer.readFileCounts()); //read all file counts in SD card
  Serial.println(myDFPlayer.readCurrentFileNumber()); //read current play file number
  Serial.println(myDFPlayer.readFileCountsInFolder(3)); //read file counts in folder SD:/03
}

void loop()
{
  static unsigned long timer = millis();

  if (millis() - timer > 3000) {
    timer = millis();
    myDFPlayer.next();  //Play next mp3 every 3 second.
  }

  if (myDFPlayer.available()) {
    printDetail(myDFPlayer.readType(), myDFPlayer.read()); //Print the detail message from DFPlayer to
handle different errors and states.
  }
}

void printDetail(uint8_t type, int value){
  switch (type) {
    case TimeOut:
      Serial.println(F("Time Out!"));
      break;
    case WrongStack:
      Serial.println(F("Stack Wrong!"));
      break;
    case DFPlayerCardInserted:
      Serial.println(F("Card Inserted!"));
      break;
    case DFPlayerCardRemoved:
      Serial.println(F("Card Removed!"));
      break;
    case DFPlayerCardOnline:
      Serial.println(F("Card Online!"));
      break;
    case DFPlayerUSBInserted:
      Serial.println("USB Inserted!");
      break;
    case DFPlayerUSBRemoved:
      Serial.println("USB Removed!");
      break;
```

```
  case DFPlayerPlayFinished:
   Serial.print(F("Number:"));
   Serial.print(value);
   Serial.println(F(" Play Finished!"));
   break;
  case DFPlayerError:
   Serial.print(F("DFPlayerError:"));
   switch (value) {
     case Busy:
      Serial.println(F("Card not found"));
      break;
     case Sleeping:
      Serial.println(F("Sleeping"));
      break;
     case SerialWrongStack:
      Serial.println(F("Get Wrong Stack"));
      break;
     case CheckSumNotMatch:
      Serial.println(F("Check Sum Not Match"));
      break;
     case FileIndexOut:
      Serial.println(F("File Index Out of Bound"));
      break;
     case FileMismatch:
      Serial.println(F("Cannot Find File"));
      break;
     case Advertise:
      Serial.println(F("In Advertise"));
      break;
     default:
      break;
   }
   break;
  default:
   break;
 }

}
```

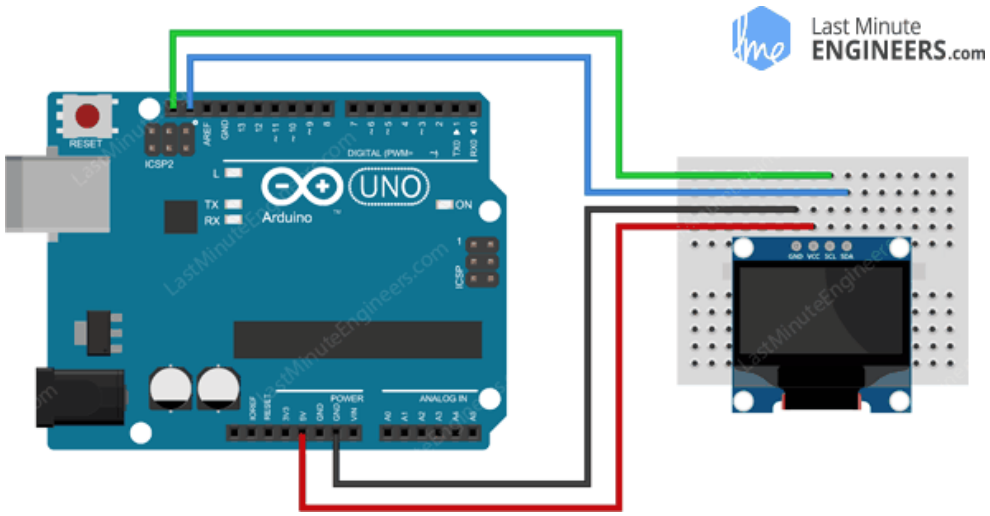
Copy

NOTE: The folder name needs to be mp3, placed under the SD card root directory, and the mp3 file name needs to be 4 digits, for example, "0001.mp3", placed under the mp3 folder. If you want to name it in Both English and Chinese, you can add it after the number, for example, "0001hello.mp3" or "0001后来.mp3".

5.1.3 Testing of 1.3 inch OLED

You will need an Arduino board, four wires, and an OLED display.

Step 1: Connecting Your Display



The OLED display should be connecting to the Arduino like the included picture.

Step 2: Identifying What Kind of Your Display Is

The types of displays may affect the codes that are uploaded to. So after connecting the display to your Arduino, you might need to run a code. Which is included.

https://playground.arduino.cc/Main/I2cScanner/

If the display is connected and the codes are uploaded, open up serial monitor, it should show something like the picture.

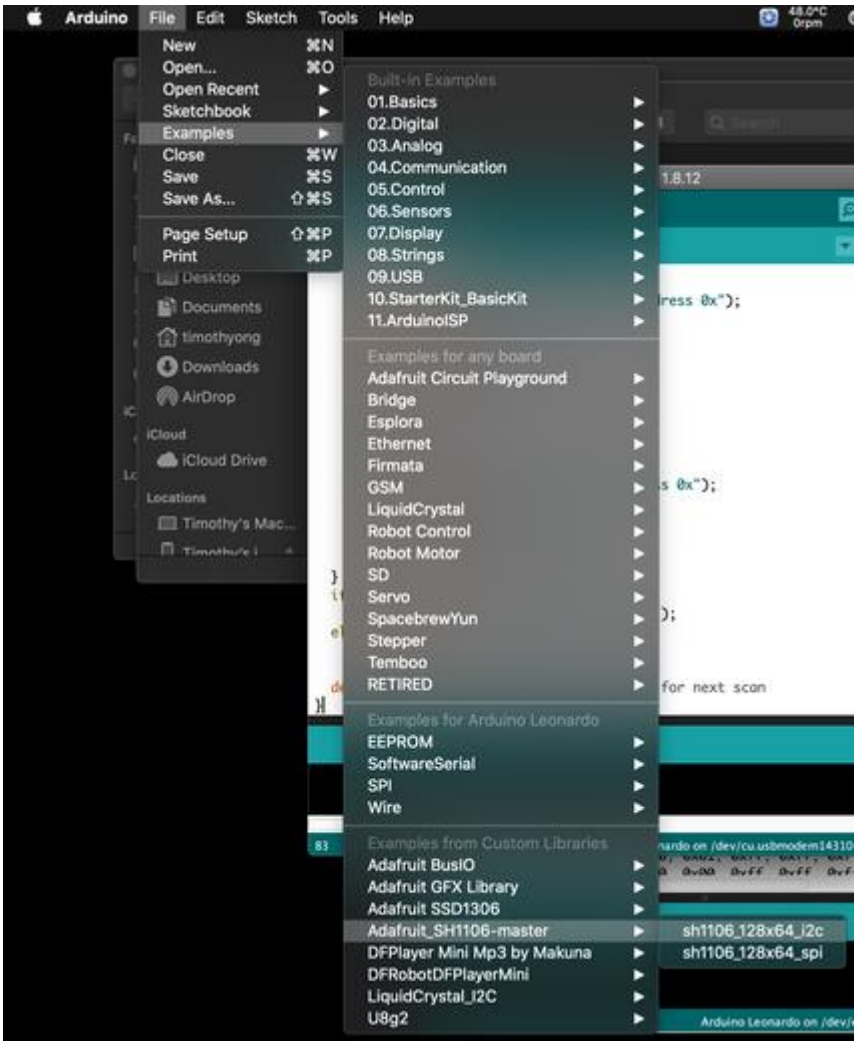
Step 3: Installing Libraries

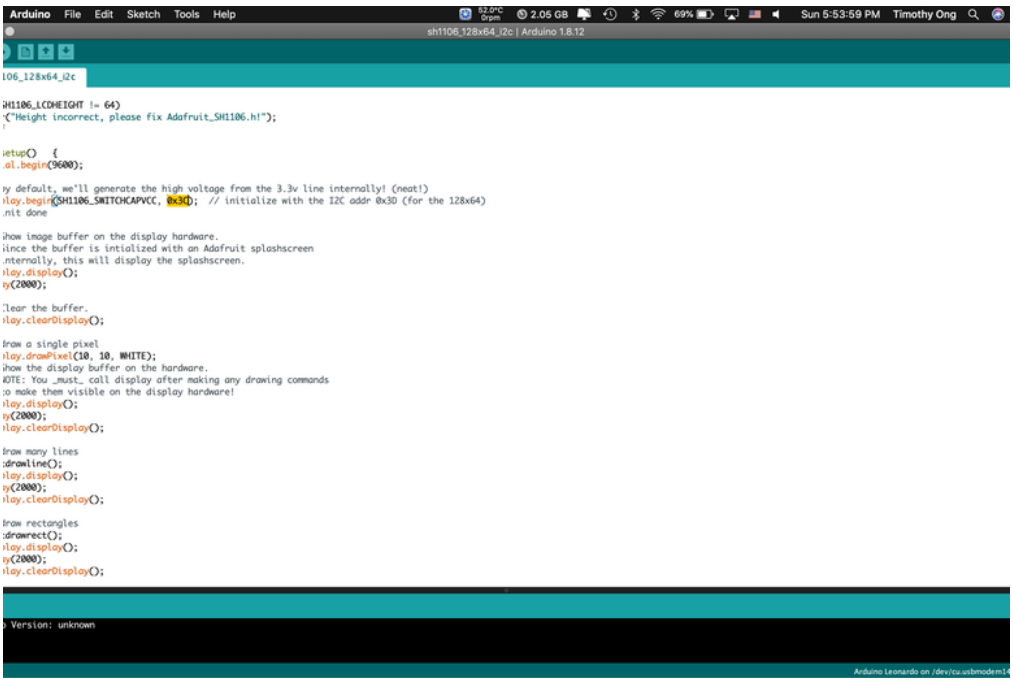
The libraries that I use are

SH1106: https://github.com/winneymj/SH1106

GFX: https://github.com/adafruit/Adafruit-GFX-Library

Step 4: Testing Your Display

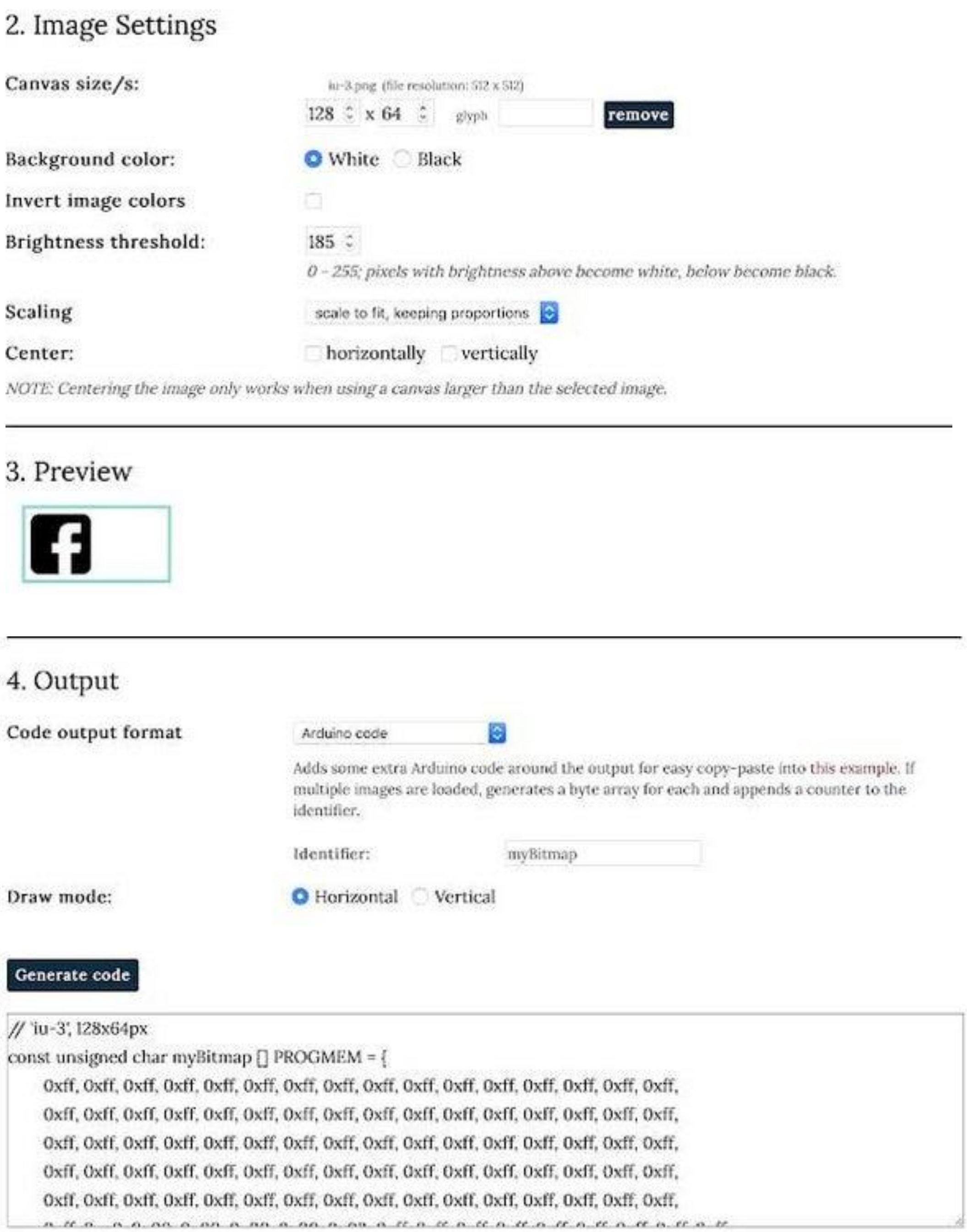




To test your display, there are testing files that can be reached by following the picture.

After opening, make sure you change the highlighted part to the result that the scanner scanned. In my case, is 0X3C.

Step 5: Uploading Your Own Picture



Till this point, your OLED display should glow up and display the example file. If not, head back and check if you are following the right instructions.

So first find a picture that you would like to display to your OLED.

Next, head to https://diyusthad.com/image2cpp and follow the steps on the picture.

Then, copy the code and paste it into the part that writes " //Paste your bitmap here". Example file is also included.

https://create.arduino.cc/editor/timothyong16/2167...

Step 6: Enjoy Your OLED Display!

If the tutorial does not work for you or isn't clear enough, the reference link might be easier to understand than mine. Or you can comment down below so I can solve your problem!

5.1.4 Testing of DC Voltmeter

In all of the experiments in this book, you will be using some test equipment to measure aspects of electricity you cannot directly see, feel, hear, taste, or smell. Electricity—at least in small, safe quantities—is insensible by our human bodies.

Your most fundamental “eyes” in the world of electricity and electronics will be a device called a *multimeter*. Multimeters indicate the presence of and measure the quantity of electrical properties such as voltage, current, and resistance. In this experiment, you will familiarize yourself with the measurement of voltage.

Voltage is the measure of electrical “push” ready to motivate charges to move through a conductor. In scientific terms, it is the specific energy per unit charge, mathematically defined as joules per coulomb. It is analogous to pressure in a fluid system: the force that moves fluid through a pipe and is measured in the unit of the Volt (V).

Digital Versus Analog Multimeters

As illustrated in Figure 3, there are two types of multimeters: digital and analog.

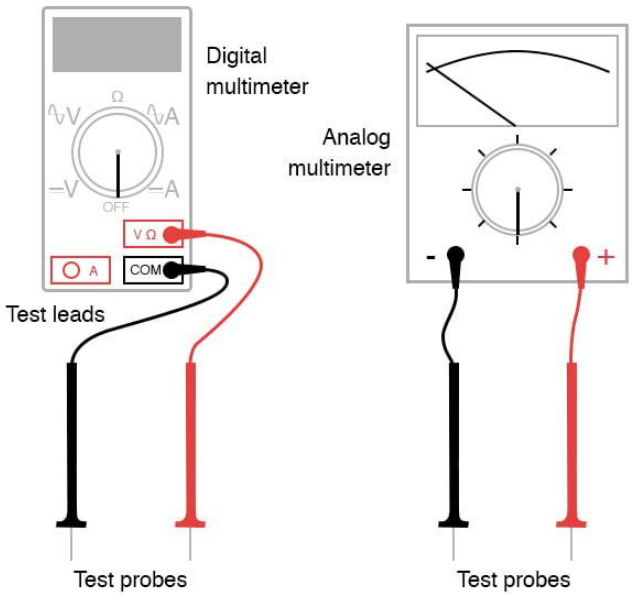


Figure 3. Digital and analog multimeters for measuring voltage, current, and resistance

Digital multimeters have numerical displays, like digital clocks, for indicating the quantity of voltage, current, or resistance. Analog multimeters indicate these quantities by means of a moving pointer over a printed scale. Your multimeter should come with some basic instructions. Read them well!

If your multimeter is digital, it will require a small battery to operate. If it is analog, it does not need a battery to measure voltage.

Manual Versus Autoranging Multimeters

Manual-ranging meters have several different selector positions for each basic quantity: several for voltage, several for current, and several for resistance. For voltage, these ranges could be µV, mV, and V.

Some digital multimeters are autoranging. An autoranging meter has only a few selector switch (dial) positions. Autoranging is to manual ranging as an automatic transmission is to a manual transmission in a car. Basically, an autoranging meter "shifts gears" automatically to find the best measurement range to display the particular quantity being measured.

Instructions

Step 1: Insert your test probes into the voltmeter. By convention, black probes are used with the negative, or common, input of the voltmeter, while red probes are used with the positive input (see Figure 3).

Step 2: Set your multimeter's selector switch to the highest-value DC volt position available. Autoranging multimeters may only have a single position for DC voltage, in which case you need to set the switch to that one position.

Step 3: Touch the red test probe to the positive (+) side of a battery and the black test probe to the negative (-) side of the same battery. The voltmeter should now provide you with an indication of the battery's voltage. If your meter is a manual-range type, and the selector switch has been set to a high-range position, the indication will be small.

Step 4: Reverse the test probe connections to the battery if the meter's indication is negative (on an analog meter, a negative value is indicated by the pointer deflecting left instead of right).

Step 5: If you have a manual-ranging voltmeter, move the selector switch to the next lower DC voltage range setting and reconnect to the battery. The indication should be stronger now, as indicated by a greater deflection of the analog meter pointer (needle) or more active digits on the digital meter display.

For best results, move the selector switch to the lowest-range setting that does not over-range the meter. An over-ranged analog meter is said to be "pegged," as the needle will be forced all the way to the right-hand side of the scale, past the full-range scale value. Additionally, an over-ranged digital meter sometimes displays the letters "OL" or a series of dashed lines. This indication is manufacturer-specific.

Step 6: What happens if you only touch one of the voltmeter test probes to one end of a battery? How does the meter have to connect to the battery to provide an indication? What does this tell us about voltmeter use and the nature of voltage? Is there such a thing as voltage "at" a single point?

Step 7: Be sure to measure more than one size of the battery, and learn how to select the best voltage range on the multimeter to give you the maximum indication without over-ranging.

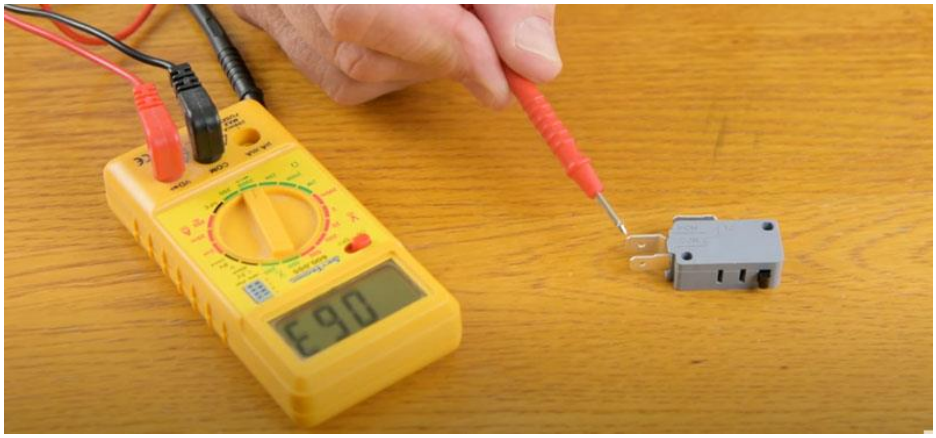
5.1.5 Testing of Lithium Polymer Battery

One way to check the voltage of a LiPo battery is using a multimeter. You can set a multimeter to measure the voltage and connect to the terminals of the JST connector. Of course, you can also connect to the +VBATT and GND pins on a PCB if the battery is connected to a board as well. Just be careful not to accidentally create a short between the two terminals when measuring with the probes.

Using a Multimeter to test the voltage on a LiPo Battery.

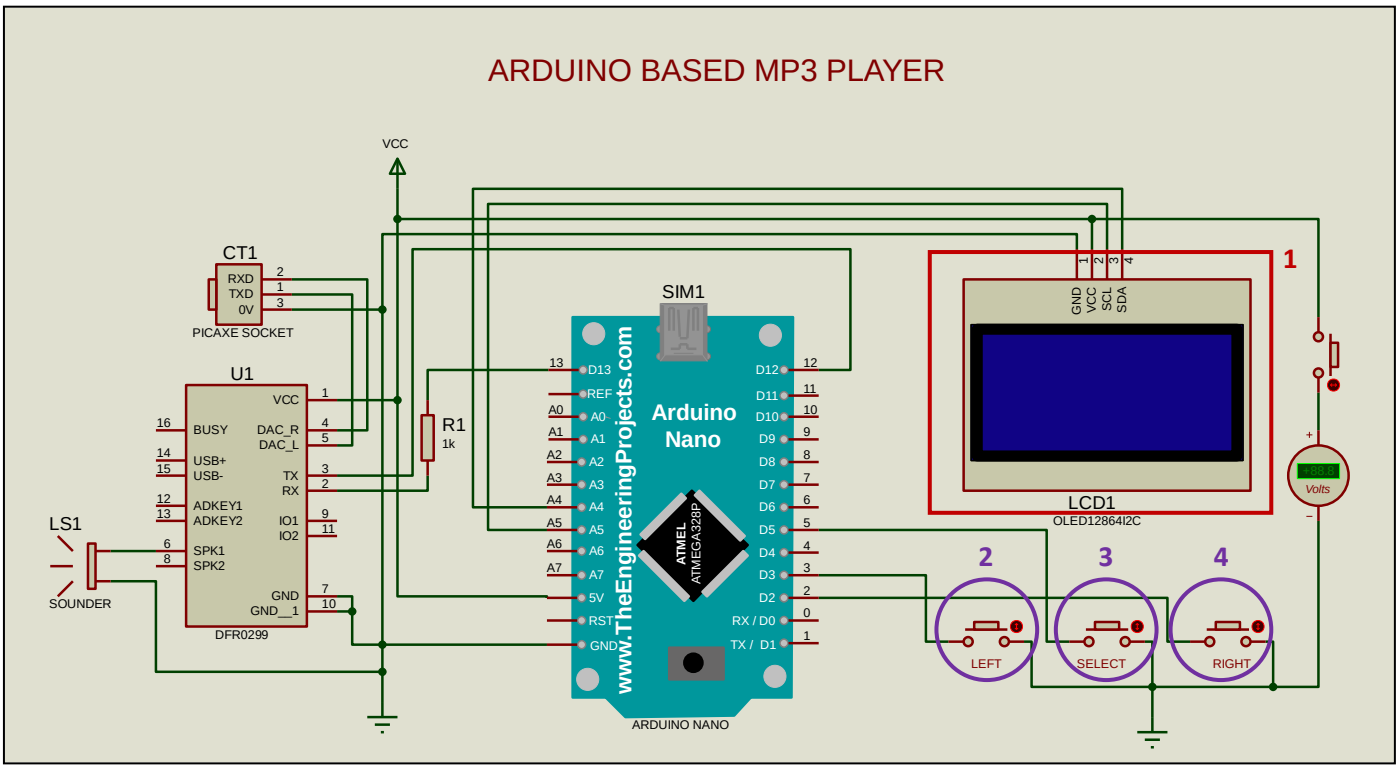
5.1.6 Testing of Push Button

To test a push button micro switch, you can use a multimeter set to continuity or resistance mode and touch the probes to the switch's terminals. Then, press or release the actuator while observing the multimeter reading. If the multimeter doesn't show continuity or gives an inconsistent reading, the switch might have an electrical fault. You can also measure the switch's contact resistance, as excessive contact resistance can affect the switch's performance.



5.2 Final testing of total Project

5.2.1 Testing in Software



Here are some steps through which we can check whether our system is giving desired output or not –

1. Click on Run Button.
2. Check whether display is working or not.
3. Clicking on the LEFT , RIGHT & SELECT button respectively we can check the corresponding operations like 'Changing the volume' , 'Changing the EQ' , 'Switching the tracks' & 'Play / Pause' are working or not.

5.2.2 Testing in Hardware

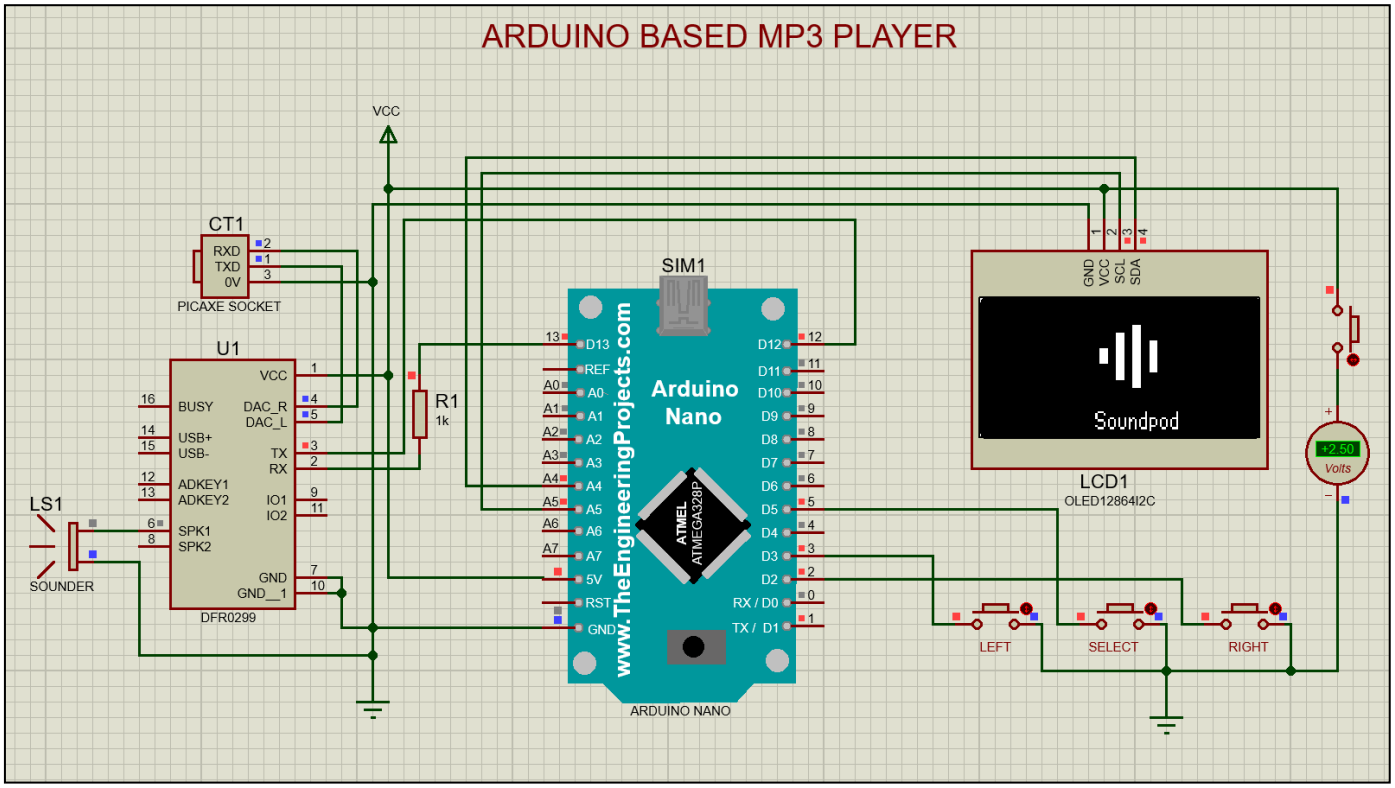
In case of Hardware also before checking the whole circuit in a glance it's highly preferred to go though some procedures –

1. To test our project before assembling components on the customized PCB, it is highly recommended to check the connections of the PCB. Briefly check the all those sources and sinks are perfect or not.
2. If 'yes' then after doing PCB assembly check the connections again according to the connection diagram given previous.
3. Power on the Circuit.
4. Clicking on the LEFT , RIGHT & SELECT button respectively we can check the corresponding operations like 'Changing the volume' , 'Changing the EQ' , 'Switching the tracks' & 'Play / Pause' are working or not.
5. Check the Headphone jack is working or not means whether you can listen to music through the headphone jack or not.

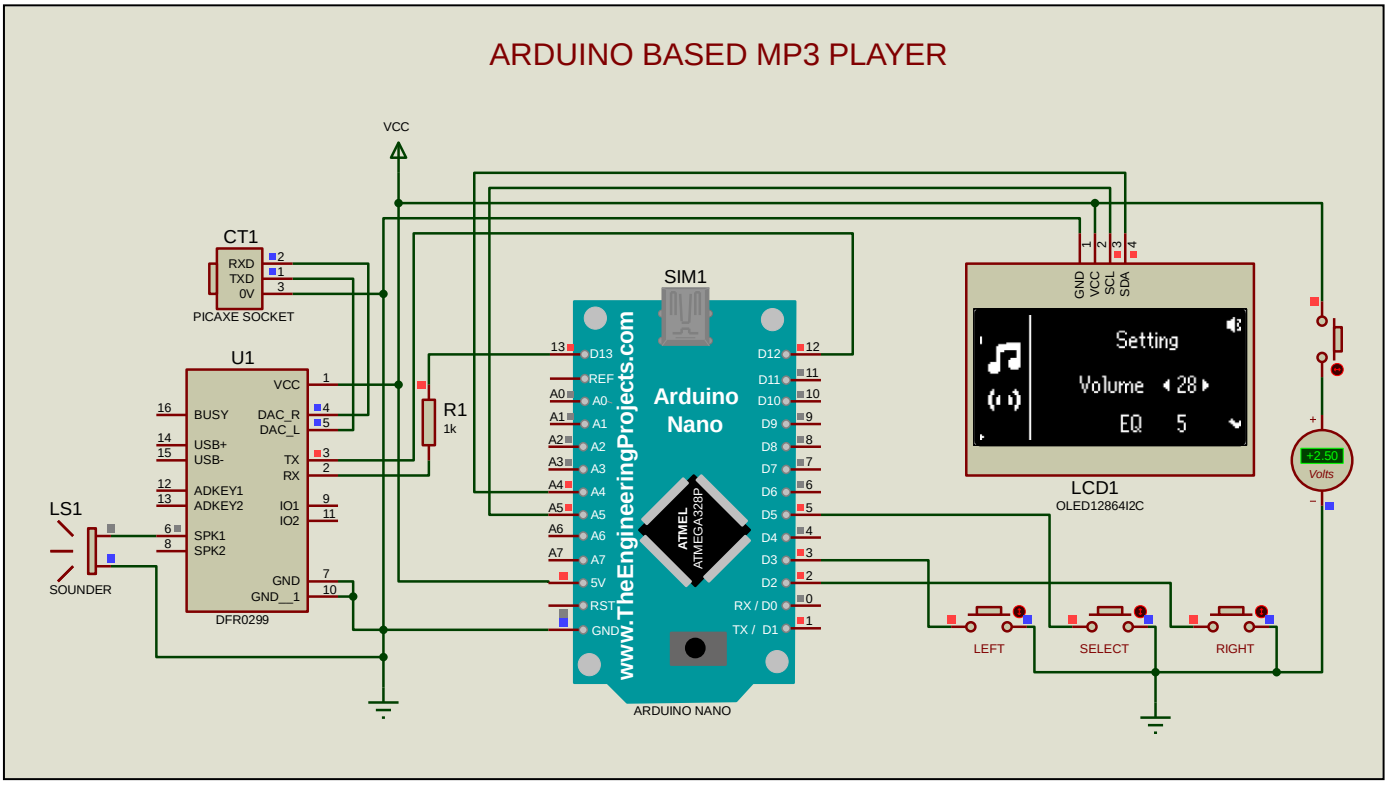
5.3 Test Results

5.3.1 Result in Software Test

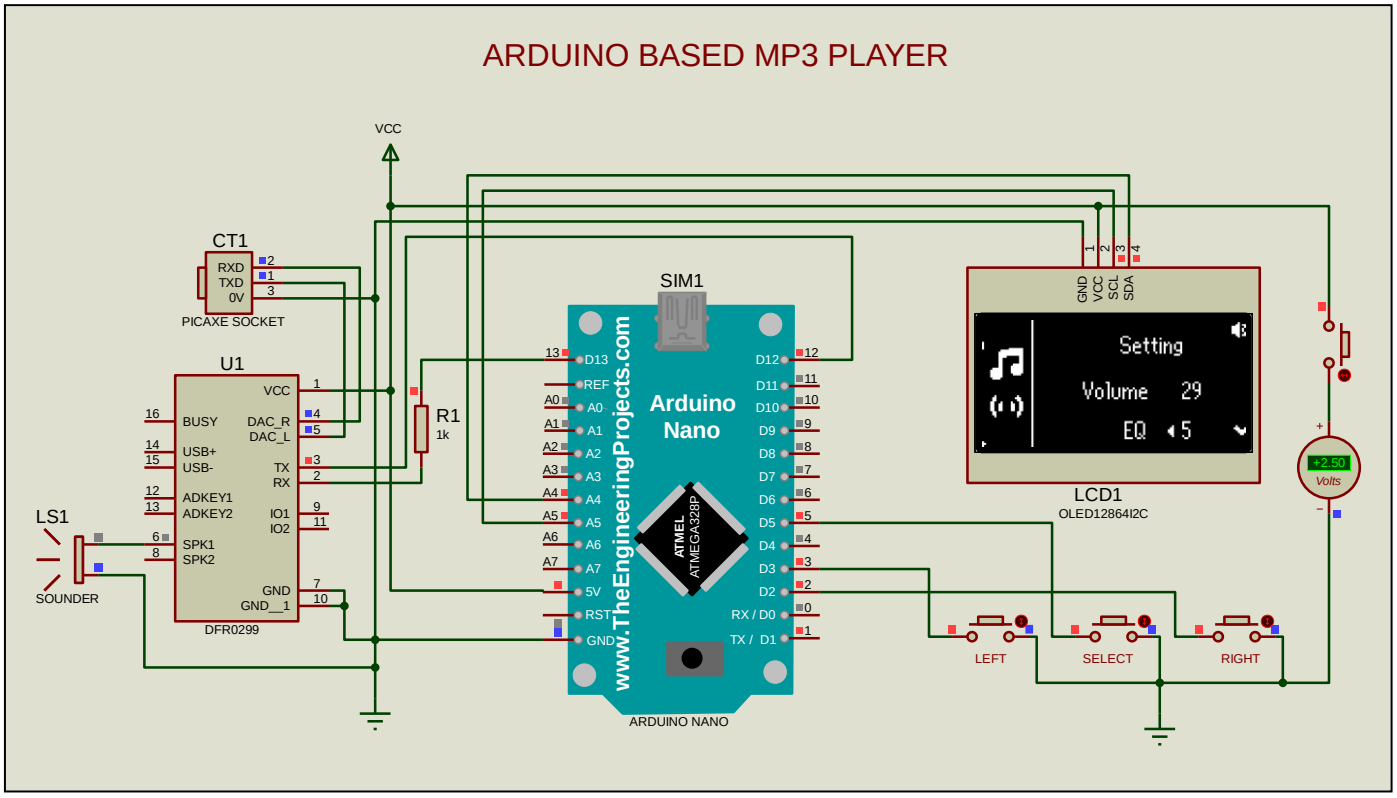
1. OLED is working / Booting….



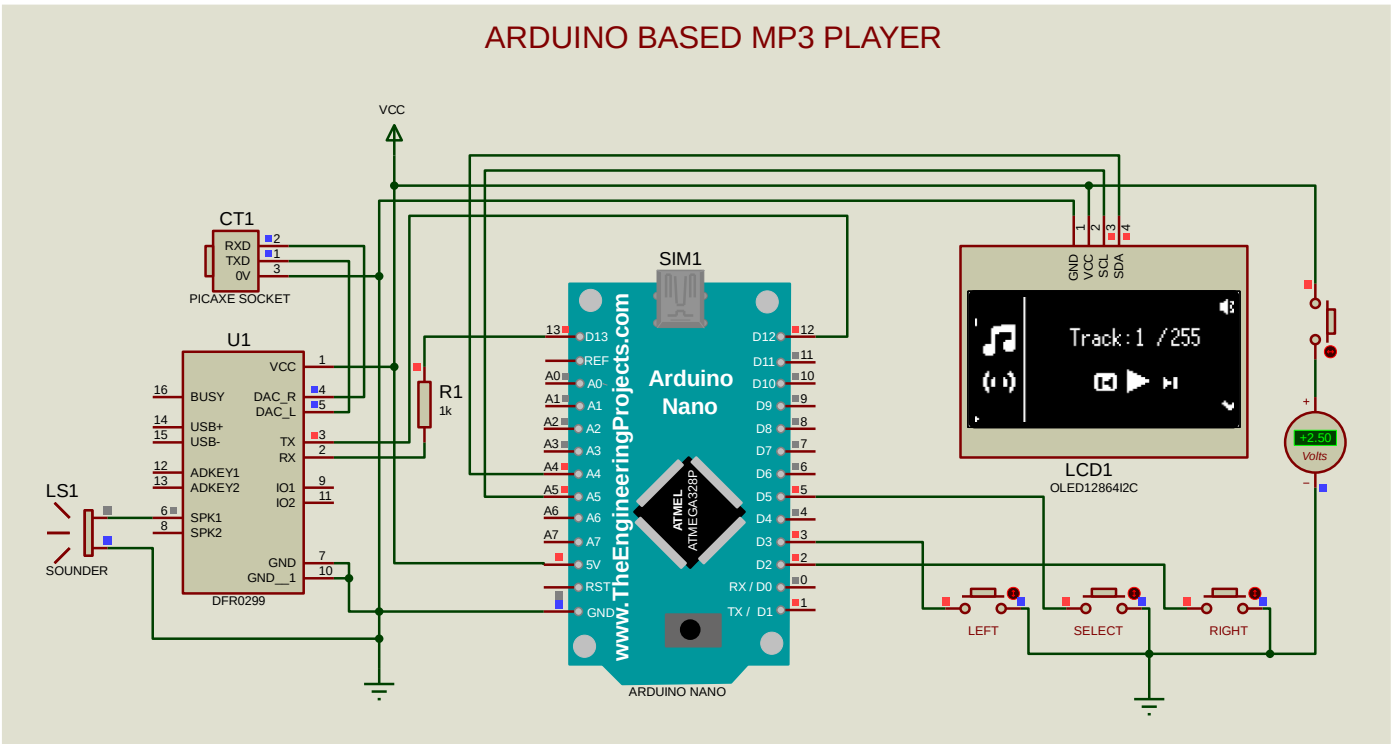
2. Changing the volume level



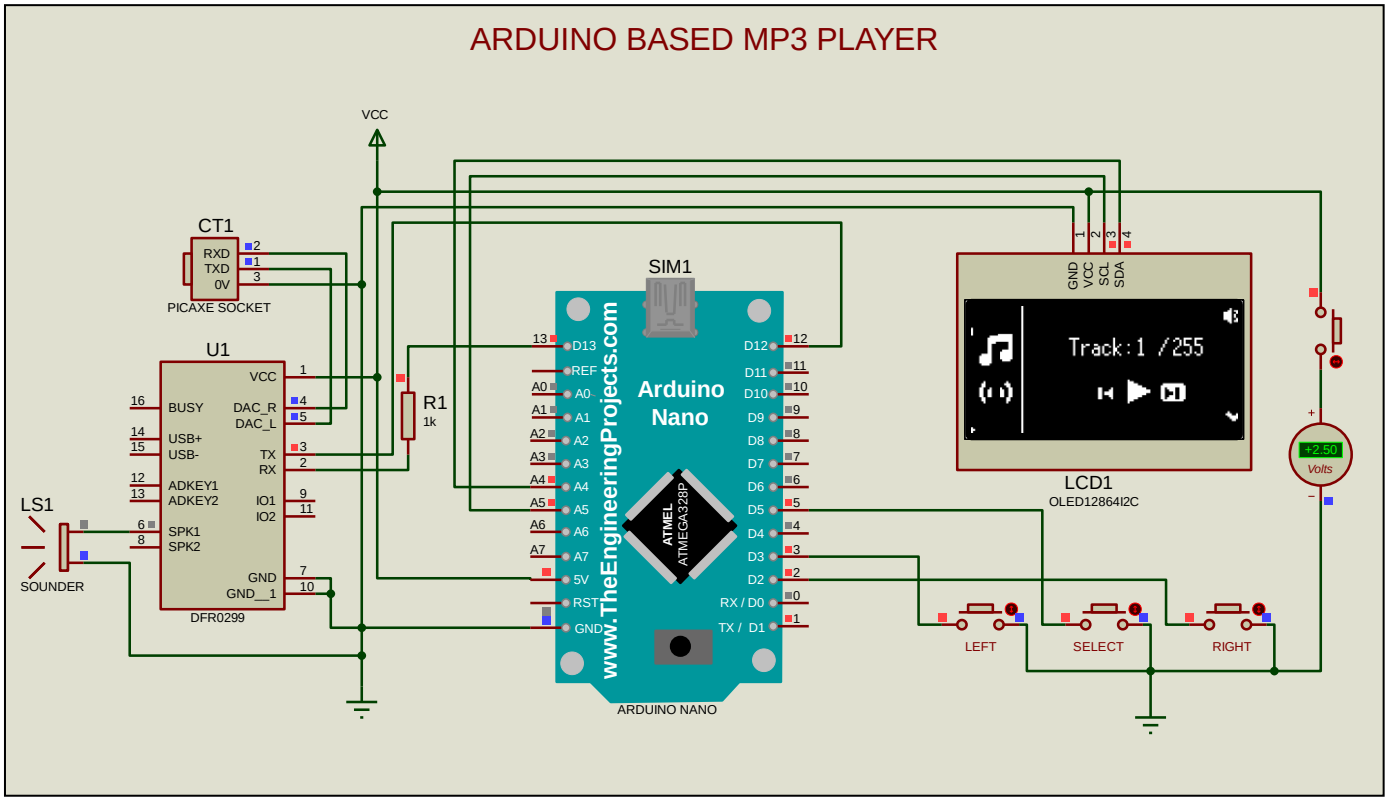
3. Changing the EQ Modes



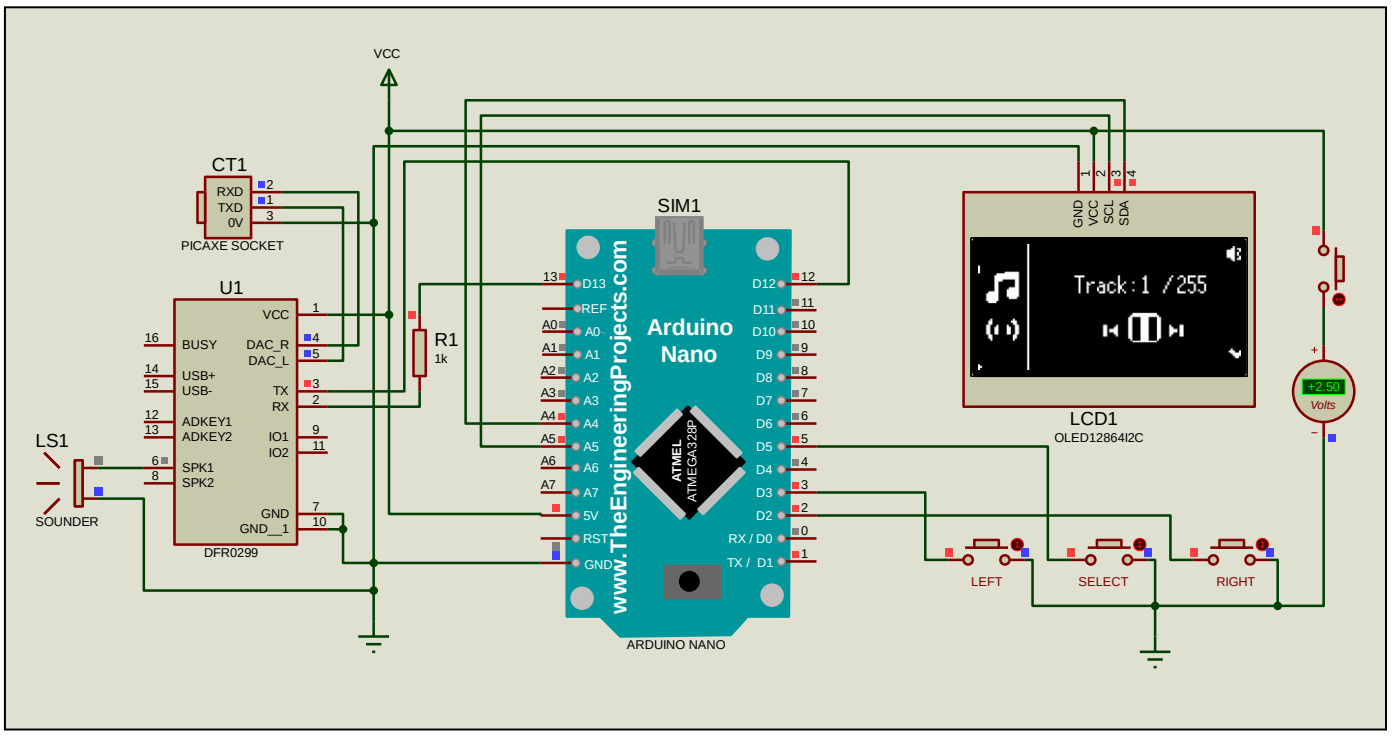
4. Switching to Prev. Track



5. Switching to Next Track

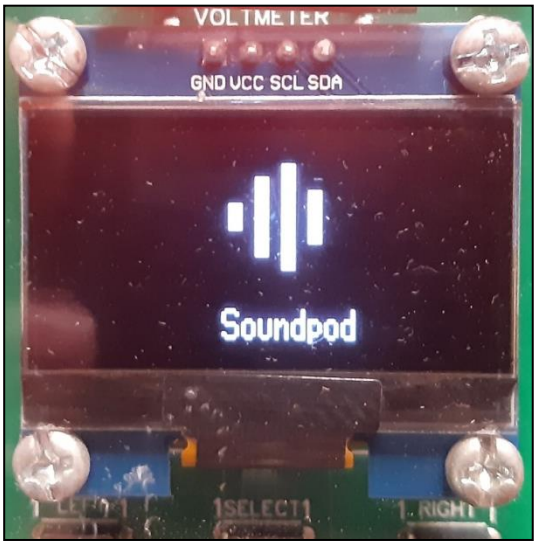


6. Play / Pause the Track

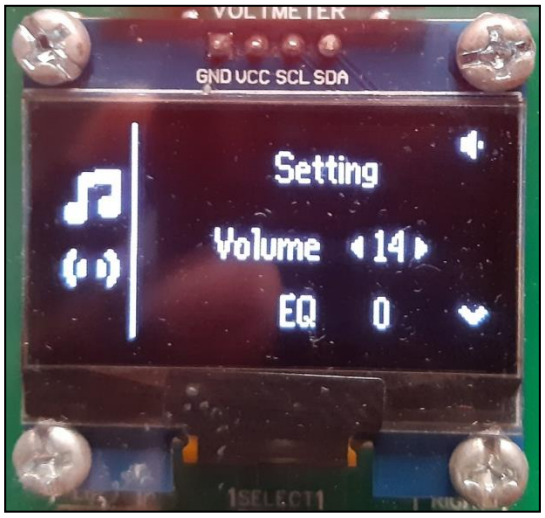


5.3.2 Result in Hardware Test

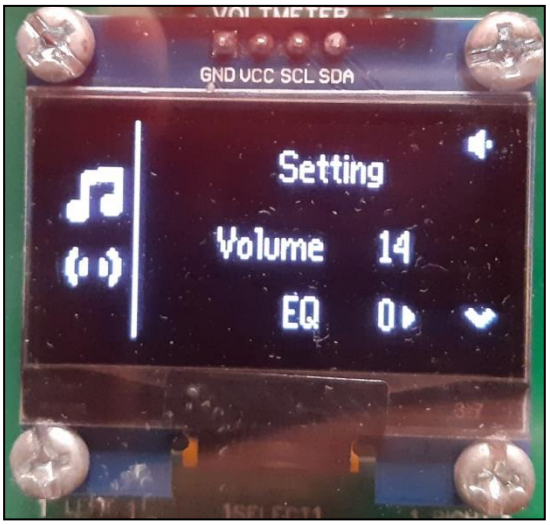
1. OLED is working / Booting….



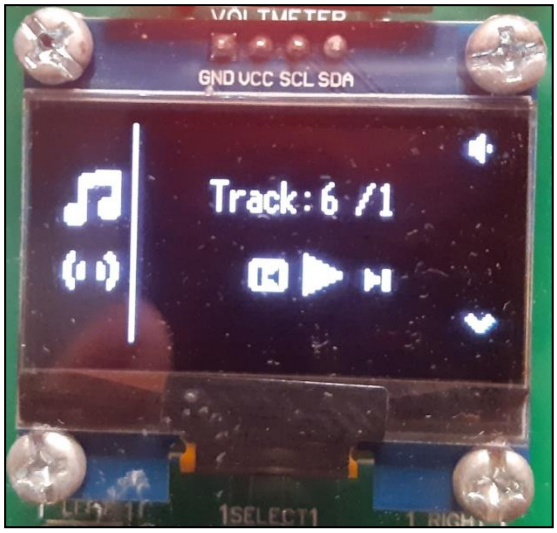
2. Changing the Volume level



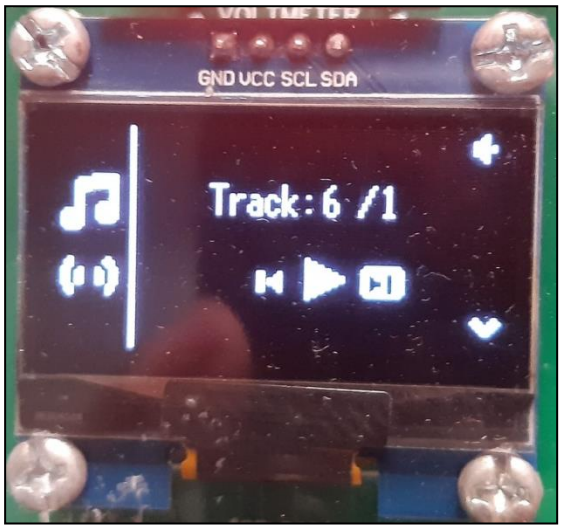
3. Changing the EQ Modes



4. Switching to Previous Track



5. Switching to Next Track



6. Play / Pause the Track

Hence, we have observed that our project is working fine along with this headphone jack is also working. So, we can say that our project is working in a desired manner.

Chapter 5 : Applications

This chapter explains the industrial application of each and every component of the project as well as the total project.

5.1 Control System

A control system is a system, which provides the desired response by controlling the output. The following figure shows the simple block diagram of a control system.

Here, the control system is represented by a single block. Since, the output is controlled by varying input, the control system got this name. We will vary this input with some mechanism. In the next section on open loop and closed loop control systems, we will study in detail about the blocks inside the control system and how to vary this input in order to get the desired response.

Examples – Traffic lights control system, washing machine.

Traffic lights control system is an example of control system. Here, a sequence of input signal is applied to this control system and the output is one of the three lights that will be on for some duration of time. During this time, the other two lights will be off. Based on the traffic study at a particular junction, the on and off times of the lights can be determined. Accordingly, the input signal controls the output. So, the traffic lights control system operates on time basis.

Classification of Control Systems: Based on some parameters, we can classify the control systems into the following ways.

Continuous time and Discrete-time Control Systems

- Control Systems can be classified as continuous time control systems and discrete time control systems based on the type of the signal used.

- In continuous time control systems, all the signals are continuous in time. But, in discrete time control systems, there exists one or more discrete time signals.

SISO and MIMO Control Systems

- Control Systems can be classified as SISO control systems and MIMO control systems based on the number of inputs and outputs present.

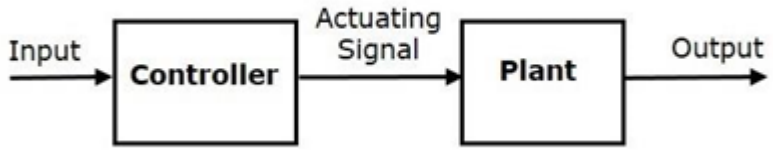
- SISO (Single Input and Single Output) control systems have one input and one output. Whereas, MIMO (Multiple Inputs and Multiple Outputs) control systems have more than one input and more than one output.

Open Loop and Closed Loop Control Systems

Control Systems can be classified as open loop control systems and closed loop control systems based on the feedback path.

In **open loop control systems**, output is not fed-back to the input. So, the control action is independent of the desired output.

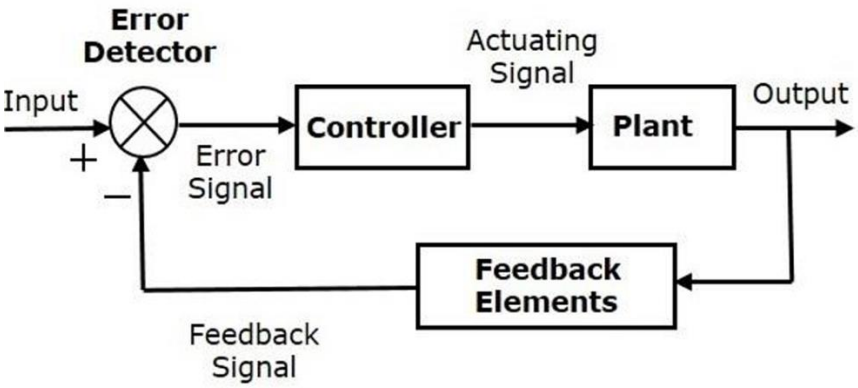
The following figure shows the block diagram of the open loop control system.



Here, an input is applied to a controller and it produces an actuating signal or controlling signal. This signal is given as an input to a plant or process which is to be controlled. So, the plant produces an output, which is controlled. The traffic lights control system which we discussed earlier is an example of an open loop control system.

In **closed loop control systems**, output is fed back to the input. So, the control action is dependent on the desired output.

The following figure shows the block diagram of negative feedback closed loop control system.



The error detector produces an error signal, which is the difference between the input and the feedback signal. This feedback signal is obtained from the block (feedback elements) by considering the output of the overall system as an input to this block. Instead of the direct input, the error signal is applied as an input to a controller.

So, the controller produces an actuating signal which controls the plant. In this combination, the output of the control system is adjusted automatically till we get the desired response. Hence, the closed loop control systems are also called the automatic control systems. Traffic lights control system having sensor at the input is an example of a closed loop control system. The differences between the open loop and the closed loop control systems are mentioned in the following table.

Open Loop Control Systems	Closed Loop Control Systems
Control action is independent of the desired output.	Control action is dependent of the desired output.
Feedback path is not present.	Feedback path is present.
These are also called as **non-feedback control systems**.	These are also called as **feedback control systems**.
Easy to design.	Difficult to design.
These are economical.	These are costlier.
Inaccurate.	Accurate.

5.2 Microcontroller

A **microcontroller** is a small and low-cost microcomputer, which is designed to perform the specific tasks of embedded systems like displaying microwave's information, receiving remote signals, etc.

The general microcontroller consists of the processor, the memory (RAM, ROM, EPROM), Serial ports, peripherals (timers, counters), etc.

Types of Microcontrollers: Microcontrollers are divided into various categories based on memory, architecture, bits and instruction sets. Following is the list of their types –

Bit - Based on bit configuration, the microcontroller is further divided into three categories.

- **8-bit microcontroller** − This type of microcontroller is used to execute arithmetic and logical operations like addition, subtraction, multiplication division, etc. For example, Intel 8031 and 8051 are 8 bits microcontroller.

- **16-bit microcontroller** − This type of microcontroller is used to perform arithmetic and logical operations where higher accuracy and performance is required. For example, Intel 8096 is a 16-bit microcontroller.

- **32-bit microcontroller** − This type of microcontroller is generally used in automatically controlled appliances like automatic operational machines, medical appliances, etc.

Memory: Based on the memory configuration, the microcontroller is further divided into two categories.

- **External memory microcontroller** − This type of microcontroller is designed in such a way that they do not have a program memory on the chip. Hence, it is named as external memory microcontroller. For example: Intel 8031 microcontroller.
- **Embedded memory microcontroller** − This type of microcontroller is designed in such a way that the microcontroller has all programs and data memory, counters and timers, interrupts, I/O ports are embedded on the chip. For example: Intel 8051 microcontroller.

Instruction Set: Based on the instruction set configuration, the microcontroller is further divided into two categories.

- CISC − CISC stands for complex instruction set computer. It allows the user to insert a single instruction as an alternative to many simple instructions.
- RISC − RISC stands for Reduced Instruction Set Computers. It reduces the operational time by shortening the clock cycle per instruction.

Applications of Microcontrollers: Microcontrollers are widely used in various different devices such as –

- Light sensing and controlling devices like LED.
- Temperature sensing and controlling devices like microwave oven, chimneys.
- Fire detection and safety devices like Fire alarm.
- Measuring devices like Volt Meter.

5.2.1 Arduino Nano

Some projects that can be created using the Arduino Nano include:

- QR code scanner
- DIY Arduino pedometer
- Digital taxi fare meter
- Arduino wireless weather station
- Smart blind stick
- RFID door lock
- Water level controller

The Arduino Nano is a microcontroller board with 32 pins, including 8 analog pins, that can be used for variety of applications. Some of its application include:

- Robotics
- Control system
- Instrumentation
- Automation
- Embedded system
- Engineering student project
- Medical instruments
- Industrial automation
- Android applications
- GSM based project
- Home automation and defense system

5.3 DF Player Mini module

Overview

DFPlayer mini MP3 player is a small and low cost MP3 module player with a simplified output directly to the speaker. The module can be used as a standalone module with attached battery, speaker and push buttons or used in combination with an Arduino UNO or any other with RX/TX capabilities.
It perfectly integrates hard decoding module which supports common audio formats such as MP3, WAV and WMA.
Besides, it also supports TF card with FAT16, FAT32 file system. Through a simple serial port, users can play the designated music without any other tedious underlying operations.

Application of DF Player Mini MP3 player:

- Car navigation voice broadcast
- Road transport inspectors, toll stations voice prompts
- Railway station, bus safety inspection voice prompts
- Electricity, communications, financial business hall voice prompts
- Vehicle into and out of the channel verify that the voice prompts
- The public security border control channel voice prompts
- Multi-channel voice alarm or equipment operating guide voice
- The electric tourist car safe driving voice notices
- Electromechanical equipment failure alarm
- Fire alarm voice prompts
- The automatic broadcast equipment, regular broadcast

5.4 OLED Display

Applications of OLED Display Technology

- TVs
- Cellphone screens
- Computer screens
- Keyboards
- Lights
- Portable device displays

1. OLED Televisions

- **Sony application:** Sony released XEL-1 in the year February, 2009. The first OLED TV sold in all stores had high resolutions and these specifications: 11” screen and 3mm thin. The approximate weight of this TV was 1.9 kg, along with a 178 degrees wide range of viewing angle.
- **LG applications:** In the year 2010, LG had produced new OLED television with a 15 inch screen, 15EL9500, and announced an OLED 3D television with these specifications: 31” screen and 78cm in the year March, 2011.

- **Mitsubishi applications:** Lumiotec is the first company in the world that has been developing and selling mass produced OLED lighting panels with immense brightness and long lifetime since January 2011. Luiotec is the joint venture of Mitsubishi heavy industries.

2. Keyboards: In Optimus Maximus Keyboard type of keyboard keys are linked to display notes, applications, numerals, etc., through programming to perform a series of functions.

3. Lighting: OLEDs are used for flexible and bendable lighting, wallpaper and also for transparent lighting.

Thus, the OLED system gives exceptional display compared to other display systems. Due to its robust design, these systems come in several portable devices like cell phones, DVD players, digital video cameras, etc. And, this is the weight and space saving technology. Finally, the applications of OLEDs are being continuously expanding, and – as a matter of fact – this is definitely going to be the best display technology in the future.

5.5 Boost Converter

Applications of Boost Converters

Boost converters find widespread use in various industries and applications, thanks to their ability to efficiently step up voltage levels. Some common applications include:

- **Power Supplies:** Boost converters are used in power supplies for electronic devices, such as laptops and smartphones, to generate the required voltage levels for different components within the device.
- **Electric Vehicles:** In electric vehicles, boost converters are employed to increase the battery voltage to a higher level, suitable for driving the electric motor and other onboard systems.
- **Solar Energy Systems:** Boost converters are used in solar power systems to maximize power output by matching the panel's output voltage to the battery bank's optimal charging voltage.
- **LED Lighting:** LED lights require a constant current for proper operation. Boost converters are often utilized to step up the voltage in LED driver circuits, ensuring stable and efficient performance.
- **Portable Electronics:** Devices like portable chargers, power banks, and GPS units rely on boost converters to maintain a stable output voltage while drawing power from a lower-voltage battery source.

5.6 Lithium Battery

Lithium Polymer Batteries are widely used for wearable medical devices, Hardware Detector prototypes, Smart And High-tech Design, Electronic Development, etc.
A lithium polymer battery, or more correctly lithium-ion polymer battery (abbreviated as LiPo, LIP, Li-poly, lithium-poly, and others), is a rechargeable battery of lithium-ion technology using a polymer electrolyte instead of a liquid electrolyte. High conductivity semisolid (gel) polymers form this electrolyte. These batteries provide higher specific energy than other lithium battery types and are used in applications where weight is a critical feature, such as mobile devices, radio-controlled aircraft, and some electric vehicles.

Applications ofLithium Battery:

Lithium polymer batteries are used in a wide range of applications due to their high energy density, lightweight, and compact size. Some common applications of LiPo batteries include:

1. **Consumer electronics:** Lithium polymer batteries are commonly used in smartphones, tablets, laptops, digital cameras, and other portable electronic devices due to their high energy density and long battery life.
2. **Drones and remote-controlled vehicles:** Lithium polymer batteries are used to power drones, quadcopters, and other remote-controlled vehicles due to their lightweight and high power output.
3. **Medical devices:** Lithium polymer batteries are used in medical devices such as implantable devices, portable monitoring equipment, and diagnostic tools due to their small size, high energy density, and long battery life.
4. **Electric vehicles:** Lithium polymer batteries are used in electric vehicles such as electric cars, bikes, and scooters due to their high energy density and ability to deliver high power output.
5. **Aerospace and defense:** Lithium polymer batteries are used in aerospace and defense applications such as satellites, spacecraft, and missiles due to their lightweight and high energy density.
6. **Power banks and backup power:** Lithium polymer batteries are used in power banks and backup power systems to provide a portable and reliable source of power for charging electronic devices or providing backup power during power outages.

Overall, Lithium polymer batteries are versatile and widely used due to their high energy density, lightweight, and compact size, making them suitable for a wide range of applications where portability and energy efficiency are important.

Wearable medical devices, Hardware Detector prototypes, Smart and High-tech Design, Electronic Development,

- **Wearable medical devices – Lithium Polymer Batteries**
  Lithium Polymer Batteries are widely used for Automatic pipettes, blood pressure devices, blood sugar indicators, body fat monitors, dosing systems, insulin pumps, thermometers, veterinary medicines…

- **Hardware Detector prototypes – Lithium Polymer Batteries**
  Lithium Polymer Batteries used for Electric vehicles, Radio controlled equipment and aircraft, 3-Cell Lithium polymer batteries for RC models, LiPo batteries are now almost ubiquitous when used to power radio-controlled aircraft, radio-controlled cars, and large scale model trains, where the advantages of lower weight and increased capacity.

- **Smart And High-tech Design – Lithium Polymer Batteries**
  eCall solution, vehicle tracking, Bluetooth headset, cellular phones, cordless phones, handy terminal, mobile phones, modems, pagers, personal digital assistants(PDA), personal mobile radius(PMR), satellite phones, smartphones, walkie-talkie…

- **Electronic Development – Lithium Polymer Batteries**
  LiPo batteries are pervasive in Personal electronics, such as mobile devices, power banks, very thin laptop computers, portable media players, wireless controllers for video game consoles, wireless PC peripherals, electronic cigarettes, and other applications where small form factors are sought and the high energy density outweighs cost considerations.
  Their power to size and weight ratio is seen as a major benefit in many industries requiring critical power back up including data centers where space is often at a premium. The longer cycle life, useable energy (Depth of discharge), and thermal runaway are also seen as a benefit for using Li-po batteries,
  Lithium polymer batteries packs also see widespread use in airsoft, where their higher discharge currents and better energy density compared to more traditional NiMH batteries have a very noticeable performance gain (higher rate of fire). The high discharge currents do damage the switch contacts due to arcing (causing the contacts to oxidize and often deposit carbon), so it is advised to either use a solid-state MOSFET switch or clean the trigger contacts regularly.

- **Smart And High-tech Design – Lithium Polymer Batteries**
  eCall solution, vehicle tracking, Bluetooth headset, cellular phones, cordless phones, handy terminal, mobile phones, modems, pagers, personal digital assistants(PDA), personal mobile radios(PMR), satellite phones, smartphones, walkie-talkie…
  Lithium Polymer Batteries used for Electric vehicles, Radio controlled equipment and aircraft, 3-Cell Lithium polymer batteries for RC models, LiPo batteries are now almost ubiquitous when used to power radio-controlled aircraft, radio-controlled cars, and largescale model trains, where the advantages of lower weight and increased capacity.
  Automatic pipette, blood pressure devices, blood sugar indicators, body fat monitors, dosing system, insulin pump, thermometers, veterinary medicines…

5.7 Total system Application

'MP3' is the abbreviation for 'MPEG Audio Layer III' ('MPEG' stands for 'Motion Pictures Expert Group'). It's a compressed digital audio file. A player stores these files on a memory chip so that you can play it back at your leisure.

Features and benefits of an MP3 player:

- Contains a miniature hard disk drive, which can store huge quantities of data.
- MP3s usually contain music, but the format is also used for general audio files such as audiobooks and podcasts.
- MP3s can be listened to with earphones or headphones, or played through speakers via an external amplifier – for example, a 'dock' with speakers or through a stereo system.
- MP3 players allow the user to play music, podcasts and so on anywhere.

- Music is easily stored and managed on the MP3 player, including skipping tracks or, on some models, arranging playlists.

History of the MP3 player:

- The MP3 project began in 1977 in Germany, and the patent for its use was finally granted in 1989.
- The first MP3 player was produced in Korea in 1998.
- Apple produced the iPod in 2001. It has since become the most popular MP3 player in the world, although it actually converts MP3 files into its own file format.

The different types of MP3 player:

- Memory flash players: small, easy to use and relatively inexpensive, ideal for joggers but have a limited storage capacity. Batteries are replaceable.
- Micro hard-drive players: smaller than hard drive players but more portable.
- Hard-drive players: much larger capacity, holding thousands of tracks, but their rechargeable batteries can't be replaced.
- Multimedia players: larger and heavier, but allow the user to watch movies; usually have a large colour screen.

Choosing an MP3 player

- Some of the most popular brands of MP3 player are iPod, Philips, Zen and Archos.
- Many of these are available in the micro format – for instance, iPod mini and Zen Micro.
- An interesting alternative is the Eco-Media Player Revolution. This has a wind-up facility – a one-minute wind of the handle gives 45 minutes of play time.

Future Scope

Though I have put the features those should be possible at maximum limit in a MP3 Player. Yet If I talk about the upgradations of this project then there are some scopes those one can upgrade. Those scopes has been described below –

- We can attach online radio + mp3 player + Clock facility and try to make an all in one system. But there is problem. To access online radio facility we need wifi connectivity. And for that we have to use either ESP32 board or ESP 8266 board. That means we have to come out of Arduino.
- But if we want to go with the Arduino only then using Arduino mega we can build a mp3 player + clock system together.
- Only using esp32 we can make web browser + touch controlled radio station.
- By using Arduino only we can add facility of voice control means we can control the music using our human voice. Using human voice we can control volume, previous track & Next track.
- By using ESP8266 module we can build an online mp3 player though which we will be able to control and use online music platforms like Spotify.

These were the top 5 major upgraded versions those we can make. Along with this we can also perform some minor works on our existing project. Those are –

- We can attack a speaker with amplifier facility so that one can listen to music without any earphone or directly through speaker.
- We can upgrade audio quality by attaching audio filters because the audio that is currently playing is quite distorted.
- Now instead of using 3.5 mm headphone jack we can attach Bluetooth audio transmitter so that we can complete with modern Bluetooth earphone. Means we can listen to music using the Bluetooth earphones from this MP3 player.

Conclusions

8.1 Overall Conclusion

From the above scenarios we have summarised few points –

- ☑ All the components are performing well without producing any error.
- ☑ All the hardware components have been interfaced with main controller.
- ☑ The programming has been done by using Arduino IDE successfully.
- ☑ The Circuit has been designed by using Simulator (Proteus 8 Professional) Successfully.
- ☑ The Physical Circuit diagram has been also made.
- ☑ The errors has been studied and recovery successfully.
- ☑ PCB has been made and assembled successfully.
- ☑ The Electronics products have been assembled successfully.
- ☑ Now the product can be launched as an industrial product work.

Analysing the above points we can conclude that we have full filled our objectives that we fixed for our project. Additionally we have made a transparent enclosure to give it a cool look and we have also use a wooden packing box for safe delivery of our product as a commercial industrial prototype.

8.2 Environmental study

Nowadays, engineering projects must take into account the environmental impact.

At the end of the life of each device, those must be deposited in containers of waste electrical and electronic equipment for recycling.

The use of batteries as a power source, for different components is harmful to health and the environment, if the batteries are not recycled correctly. These batteries contain heavy metals which are very harmful to health. The layer which protects them is readily decomposed and if these batteries are located into an inadequate location such as on the ground, the waste is filtered and consequently, it may pollute the water. Moreover, it recommends using rechargeable batteries or direct connection to power source, allowing a long period of duration, and prevents continuous replacement of batteries.

The material of the Structure / Frame / Body also plays a main role to the environment pollution. For this reason the body or the structure of the project has been made by the transparent acrylic sheet i.e. a type of a degradable plastic material which is very eco-friendly.

All the components used in the development of this project meet with the RoHS Directive which aims to restrict certain dangerous substances commonly used in electronic and electronic component.

References

Throughout this project the references those have been used are given below –

- Arduino Nano Board - https://www.rs-online.com/designspark/what-is-arduino-nano-a-getting-started-guide
- MT3608 Step Up Module - https://components101.com/modules/mt3608-2a-dc-dc-step-up-power-module
- DF Player Mini Module - https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299
- 1.3 inch OLED - https://www.oled-info.com/oled-introduction
- DC Voltmeter - https://byjus.com/jee/voltmeter/
- LIPO Battery - https://www.linkedin.com/pulse/lithium-polymer-battery-working-characteristics-vijay-tharad-plj6c?utm_source=share&utm_medium=member_android&utm_campaign=share_via
- TP4056 type C Charging Module - https://www.dnatechindia.com/TP4056-type-c-Li-ion-lithium-Battery-Charging-Module.html
- Tactile Push button micro switch - https://www.e-switch.com/blog/what-are-pushbutton-switches-vs-tactile-switches/
- 1K Resistor - https://byjus.com/physics/resistor/
- SPDT Slide switch - https://www.elprocus.com/slide-switch/
- Breadboard - https://www.javatpoint.com/breadboard
- 3.5mm Audio Jack - https://www.scienceabc.com/innovation/3-5mm-jack-electronic-devices-become-norm.html
- PCB - https://resources.altium.com/p/what-is-a-pcb
- 3.5 mm wired headphone - https://components101.com/connectors/35mm-audio-jack
- Transparent Acrylic sheet - https://www.acmeplastics.com/what-is-acrylic-plexiglass
- Wooden Box - https://www.amazon.in/dp/B078NHH2JT?psc=1&ref=ppx_pop_dt_b_product_details
- Arduino IDE - https://www.arduino.cc/en/software
- Proteus 8 Professional - https://www.labcenter.com/
- Fritzing Software - https://maker.pro/custom/tutorial/an-introduction-to-circuit-design-with-fritzing
- EasyEDA PCB design software - https://www.elprocus.com/how-to-design-circuits-online-free-with-easyeda/
- Commercial MP3 Player - https://www.digitalunite.com/technology-guides/music-audio/listening-devices/what-mp3-player#:~:text=MP3%20players%20allow%20the%20user,on%20some%20models%2C%20arranging%20playlists

Appendix

10.1 Appendix A : Code

[Folder : Soundpod]

```
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <DFRobotDFPlayerMini.h>
#include <U8g2lib.h>
#include <Wire.h>
#include <EEPROM.h>
#include "OneButton.h"

// 65 - Audio Settings Icon
// 66 - Flash screen Spectrum
// 67 - Battery empty
// 68 - Battery full
// 69 - left adjustment button
// 70 - right adjustment button
// 71 - back button
// 72 - pause button
// 73 - play button
// 74 - previous button
// 75 - Next button
// 76 - Music Icon
// 77 - volume full
// 78 - volume half
// 79 - volume low

extern const uint8_t soundpod_icon_pack_font[350] U8G2_FONT_SECTION("soundpod_icon_pack_font") =
  "\17\0\4\4\5\6\3\3\7\34 \0\0\16\0\16\0\0\0\0\0\1AA&p\341\241\22\64\214\310"
  "!BE\210\11\24\206\210 !D\4\11!\42H\10\21ABD\10\25\42TL\320 \0BZ\34"
  "$\301\17\202<H\362 \311\203$\17\202\20y\20\204\310\203 D\36\4!\362 \10QH\210BB"
  "\24\22\42\204\210\42D\24!\242\10\21E\210(BD\21\42\212\20\321\220\20\205\204($D!!\12"
  "\11\221\7A\210<\10B\344A\20\42\17\222<H\362 \311\203$\10\0C\17\310`\221p\42T\210"
  "PS\205\70\1\0D\14\310`\221p\342\304\3\24'\0E\12\303x\221\22\2\205\220\0F\13\303p"
  "\221\20D\4\212 \0G\16\250\240\221\21(\304\220\21fH\211\1H\12\214\261\241@\10\377G\4I"
  "\24\213\261\241\20t`\261CI\36<\10r\250\330\300\240\0J\14\307`\221 h\14\11k\10\5K"
  "\15\307`\221\20\210\214\11\23d\6\11L\35\20\42\241yj\21G\304\4\11\24$P\220@A\242\12"
  "\11\62Q\210W$\314\222\5M\24\10!\221\23L\204\10\42&B\224\10QD\12A\201\0N\16\6"
  ")\221\23H\10\25\25\211\12\2\0O\13\4\61\221\23\344\301\20\61\1\0\0\0\4\377\377\0";

const uint8_t leftButtonPin = 2;
const uint8_t selectionButtonPin = 5;
const uint8_t rightButtonPin = 3;

volatile uint8_t sMenuSelection = 2;
volatile uint8_t selection = 1;
```

```
volatile bool updateScreen = true;

// Variables
uint8_t filecounts;                 // total number of files in current folder
uint8_t foldercounts;               // total number of folders on sd-card

uint8_t volume = 20;                // current volume (0 .. 30)
uint8_t folder = 1;                 // current sd-card folder
uint8_t file   = 1;                 // curent file in current folder
uint8_t eq = 0;

uint8_t batteryLevel = 0;
uint8_t mins = 0;
unsigned long lastBatteryRead = 0;

boolean playing = false;
boolean inSideMenuSelection = true;

OneButton PreviousBTN(leftButtonPin, true);
OneButton PlayBTN(selectionButtonPin, true);
OneButton NextBTN(rightButtonPin, true);

SoftwareSerial customSoftwareSerial(12,13);
DFRobotDFPlayerMini myDFPlayer;

U8G2_SH1106_128X64_NONAME_1_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE);

void setup(void) {

 //being hardware serial for debugging
 Serial.begin(9600);

// Initializing u8g2 library
 u8g2.begin();
 u8g2.firstPage();
 do{

   flashPage();

 }while(u8g2.nextPage());

 PreviousBTN.attachClick(preivousButtonClicked);
 PlayBTN.attachClick(playButtonClicked);
 NextBTN.attachClick(nextButtonClicked);

 PreviousBTN.setDebounceMs(50);
 PlayBTN.setDebounceMs(50);
 NextBTN.setDebounceMs(50);
```

```
 //Initializing software serial
 customSoftwareSerial.begin(9600);

 if (!myDFPlayer.begin(customSoftwareSerial)) {  //Use softwareSerial to communicate with mp3.
  Serial.println(F("Please insert the SD card!"));
 }
 volume = EEPROM.read(0);
 if(volume > 30)
 volume = 30;

 eq     = EEPROM.read(1);
 if(eq > 5)
 eq = 5;

 file   = EEPROM.read(2);
 if(file >= 255)
 file = 1;

 delay(1000);
 myDFPlayer.volume(volume);  //Set volume value. From 0 to 30
 delay(500);
 foldercounts = myDFPlayer.readFolderCounts();
 startFolderPlay();
//
}

void loop() {

 //Check constantly for button inputs
 PreviousBTN.tick();
 PlayBTN.tick();
 NextBTN.tick();
 //Update the display as required
 updateDisplay();
 updateDFplayer();

}
```

[Folder : Input]

```
void preivousButtonClicked()
{

 updateScreen = true;
 if(inSideMenuSelection && sMenuSelection < 2)
 {
  sMenuSelection++;
 }
```

```
 else if(selection < 4 && sMenuSelection == 1)
 {
   selection++;
 }
 else if(selection == 1 && volume < 30 &&sMenuSelection == 2 && !inSideMenuSelection)
 {
   volume++;
 }
 else if(selection == 2 && eq < 5 && sMenuSelection == 2 && !inSideMenuSelection)
 {
   eq++;
 }

}

void nextButtonClicked()
{
 updateScreen = true;
 if(inSideMenuSelection && sMenuSelection > 1)
 {
  sMenuSelection--;
 }
 else if(selection > 1 && sMenuSelection == 1)
 {
  selection--;
 }
 else if(selection == 1 && volume > 0 &&sMenuSelection == 2)
 {
   volume--;
 }
 else if(selection == 2 && eq > 0 && sMenuSelection == 2)
 {
   eq--;
 }
}

void playButtonClicked()
{
 //Selection button
 if(inSideMenuSelection)
 {
    inSideMenuSelection = false;
    updateScreen = true;
    delay(100);
 }
 else if(!inSideMenuSelection && sMenuSelection == 1)
 {
     if(selection == 1)
     {
```

```
    if(file > 1)
    {
     //previous audio
     myDFPlayer.previous();
     file--;
     if(!playing)
       playing = true;
     EEPROM.write(2, file);
    }
   }
   else if(selection == 2)
   {
    //pause / play
    if(playing)
    {
     myDFPlayer.pause();
    }
    else
    {
     myDFPlayer.start();
    }
    playing = !playing;
   }
   else if(selection == 3)
   {
     //next audio
     file++;
      myDFPlayer.next();
     if(!playing)
      playing = true;
      EEPROM.write(2, file);
   }
   else if(selection == 4)
   {
    //back to side menu
    selection = 1;
    inSideMenuSelection = true;
   }
   updateScreen = true;
   delay(200);
}else if(!inSideMenuSelection && sMenuSelection == 2)
{
  if(selection == 1)
  {
     selection = 2;
     myDFPlayer.volume(volume);
     EEPROM.write(0, volume);
  }
  else if(selection == 2)
```

```
    {
      selection = 4;
      myDFPlayer.EQ(eq);
      EEPROM.write(1, eq);
    }
    else if(selection == 4)
    {
       //back to side menu
       selection = 1;
       inSideMenuSelection = true;
    }
    updateScreen = true;
    delay(200);
  }

}
```

[Folder : Audio]

```
// starts to play the actual file in the actual folder
void startFolderPlay() {
  filecounts = myDFPlayer.readFileCountsInFolder(folder);
  myDFPlayer.playFolder(folder, file);
  playing = false;
}

void updateDFplayer()
{
   // check player status
  if (myDFPlayer.available()) {
    uint8_t type = myDFPlayer.readType();
    int value = myDFPlayer.read();

    switch (type) {
      case DFPlayerPlayFinished:
        if (file < filecounts) {
          file++;
          myDFPlayer.playFolder(folder, file);
          EEPROM.write(2, file);
          updateScreen = true;
        }
        break;
      default:
        break;
    }
  }


}
```

[Folder : Display]

```
void updateDisplay() {
 //Updating the dispaly
 if (updateScreen) {
  u8g2.firstPage();
  do {
   int ch = (sMenuSelection);
   switch (ch) {
    case 1:
     player();
     break;
    case 2:
     settings();
     break;
    default:
     Serial.println(F("Default Screen"));
   }
   updateScreen = false;
  } while (u8g2.nextPage());
 }
}

void settings() {
 sideMenu();
 topMenu();
 u8g2.setFont(u8g2_font_glasstown_nbp_tf);

 u8g2.setCursor(65, 17);
 u8g2.print(F("Setting"));

 u8g2.setFontMode(0);

 u8g2.setCursor(47, 40);
 u8g2.print(F("Volume"));
 if (selection == 1 && !inSideMenuSelection) {
  u8g2.setFont(soundpod_icon_pack_font);
  if (volume > 0)
   u8g2.drawGlyph(85, 40, 69); // drawing left ajustment button next to the volume
  if (volume < 30) {
   if (volume < 10)
    u8g2.drawGlyph(100, 40, 70); // drawing right ajustment button next to the volume for single digit volume
   else
    u8g2.drawGlyph(105, 40, 70);  // drawing right ajustment button next to the volume for double digit volume
  }
```

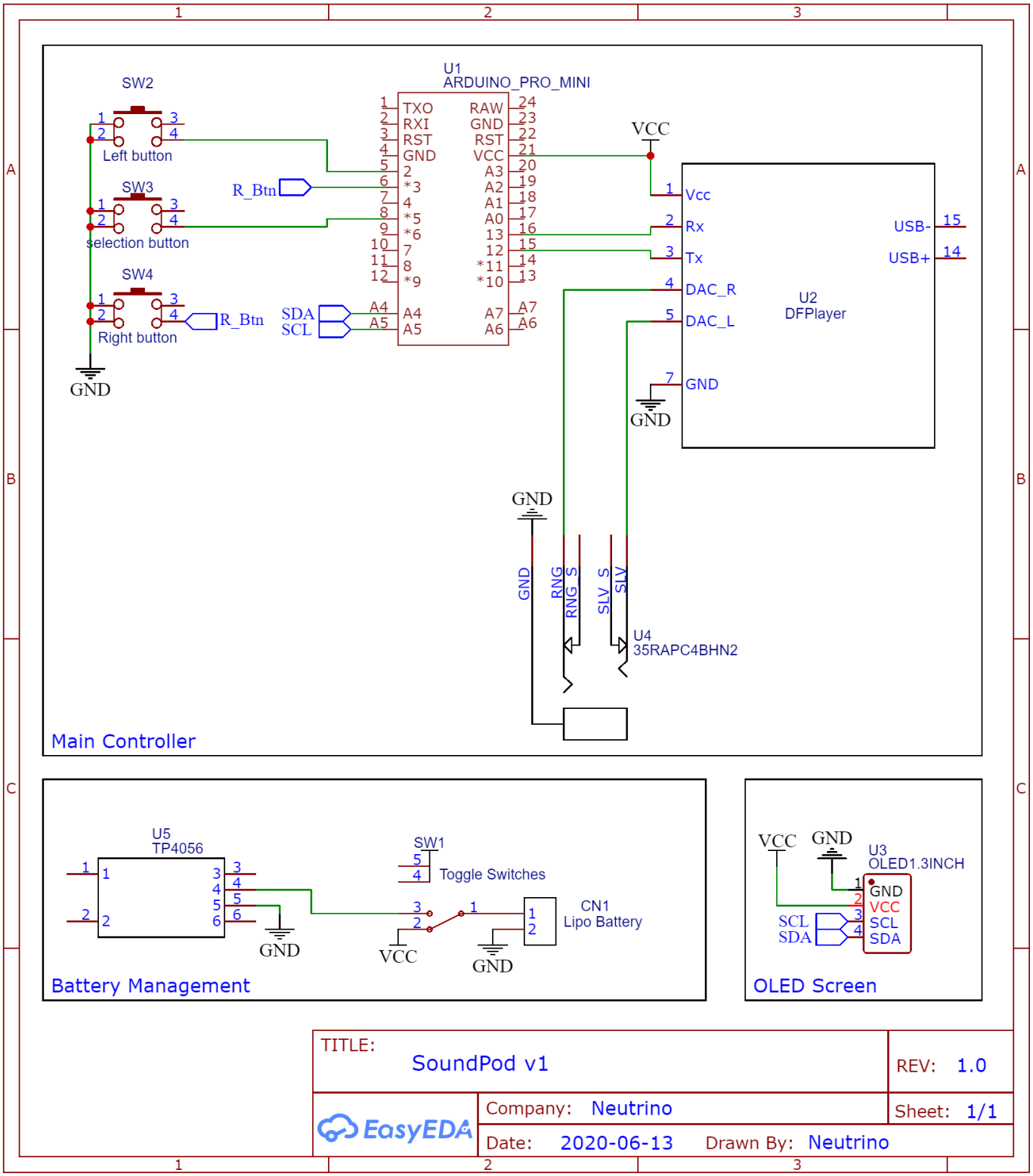
```
  }
  u8g2.setFont(u8g2_font_glasstown_nbp_tf);
  u8g2.setCursor(95, 40);
  u8g2.print(volume);
  u8g2.setDrawColor(1);

  u8g2.setCursor(67, 60);
  u8g2.print(F("EQ"));
  if (selection == 2 && !inSideMenuSelection) {
    u8g2.setFont(soundpod_icon_pack_font);
    if (eq > 0)
      u8g2.drawGlyph(85, 60, 69); // drawing left ajustment button next to the volume
    if (eq < 5)
      u8g2.drawGlyph(100, 60, 70); // drawing right ajustment button next to the volume
  }
  u8g2.setFont(u8g2_font_glasstown_nbp_tf);
  u8g2.setCursor(95, 60);
  u8g2.print(eq);
  u8g2.setDrawColor(1);

  if (selection == 4 && !inSideMenuSelection) {
    u8g2.setDrawColor(0);
  }
  u8g2.setFont(soundpod_icon_pack_font);
  u8g2.drawGlyph(120, 60, 71); // drawing back button to the bottom off the screen
  u8g2.setDrawColor(1);
}

void flashPage() {
  drawIcon(soundpod_icon_pack_font, u8g2.getDisplayWidth() / 2 - 12, u8g2.getDisplayHeight() - 22, 66);
//drawing spectrum for splash screen
  u8g2.setFont(u8g2_font_glasstown_nbp_tf);
  u8g2.setCursor(50, 63);
  u8g2.print(F("Soundpod"));
}

void topMenu() {
  u8g2.setFont(soundpod_icon_pack_font);

  if (volume >= 25)
    u8g2.drawGlyph(119, 9, 77);
  else if (volume > 10 && volume < 25)
    u8g2.drawGlyph(119, 9, 78);
  else if (volume <= 10)
    u8g2.drawGlyph(119, 9, 79);
  //
  //    u8g2.setFont(u8g2_font_glasstown_nbp_tf);
  //    u8g2.setCursor(25,9);
  //    u8g2.print(batteryLevel);
```

```
  //    u8g2.setCursor(35,9);
  //    u8g2.print("%");
}

void sideMenu() {
  const uint8_t menuListGlpy[2] = { 77, 64 };

  u8g2.setFontMode(0);

  //Audio player selection
  if (abs(sMenuSelection) == 1 && inSideMenuSelection) {
    u8g2.drawRBox(0, 12, 20, 21, 3);
    u8g2.setDrawColor(0);
  }

  //Audio player
  drawIcon(soundpod_icon_pack_font, 2, 30, 76);

  u8g2.setDrawColor(1);

  //setting Selection
  if (abs(sMenuSelection) == 2 && inSideMenuSelection) {
    u8g2.drawRBox(0, 33, 20, 19, 3);
    u8g2.setDrawColor(0);
  }
  //settings
  drawIcon(soundpod_icon_pack_font, 2, 52, 65);
  //    u8g2.setFont(menuList[1][1]);
  //    u8g2.drawGlyph(2,52,menuList[1][0]);

  u8g2.setDrawColor(1);
  u8g2.drawLine(22, 0, 22, 68);
}

void player() {
  sideMenu();
  topMenu();

  u8g2_uint_t midOriginX = 64;
  u8g2_uint_t midOriginY = 44;

  u8g2.setFontMode(0);
  u8g2.setCursor(45, 25);
  u8g2.setFont(u8g2_font_glasstown_nbp_tf);
  u8g2.print("Track : ");
  u8g2.setCursor(78, 25);
  u8g2.print(file);
  u8g2.setCursor(88, 25);
  u8g2.print('/');
```

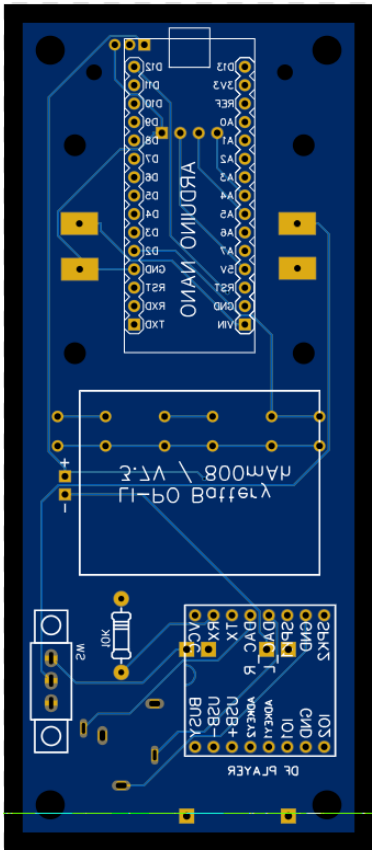
```
 u8g2.setCursor(95, 25);
 u8g2.print(filecounts);

 if (selection == 1 && !inSideMenuSelection) {
  u8g2.drawRBox(midOriginX - 7, midOriginY - 5, 11, 9, 2);
  u8g2.setDrawColor(0);
 }
 u8g2.setFont(soundpod_icon_pack_font);
 u8g2.drawGlyph(midOriginX - 5, midOriginY + 4, 74);
 u8g2.setDrawColor(1);

 if (selection == 2 && !inSideMenuSelection) {
  u8g2.drawRBox(midOriginX + 7.5, midOriginY - 8.5, 16.5, 15, 3);
  u8g2.setDrawColor(0);
 }
 u8g2.setFont(soundpod_icon_pack_font);
 if (playing) {
  u8g2.drawGlyph(midOriginX + 7.5, midOriginY + 7.5, 72);
 } else {
  u8g2.drawGlyph(midOriginX + 7.5, midOriginY + 7.5, 73);
 }
 u8g2.setDrawColor(1);

 if (selection == 3 && !inSideMenuSelection) {
  u8g2.drawRBox(midOriginX + 25, midOriginY - 5, 12, 9, 2);
  u8g2.setDrawColor(0);
 }
 u8g2.setFont(soundpod_icon_pack_font);
 u8g2.drawGlyph(midOriginX + 27, midOriginY + 4, 75);
 u8g2.setDrawColor(1);

 if (selection == 4 && !inSideMenuSelection) {
  u8g2.setDrawColor(0);
 }
 u8g2.setFont(soundpod_icon_pack_font);
 u8g2.drawGlyph(120, 60, 71);
 u8g2.setDrawColor(1);
}

void drawIcon(const uint8_t* iconName, u8g2_uint_t x, u8g2_uint_t y, uint16_t glyph) {
 u8g2.setFont(iconName);
 u8g2.drawGlyph(x, y, glyph);
}
```

10.2 Appendix B – PCB

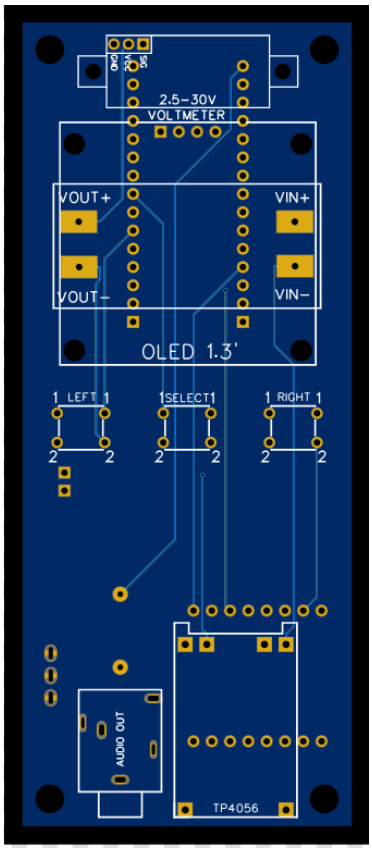
PCB Schematic (Easy EDA):



PCB View:



Bottom View



Top View

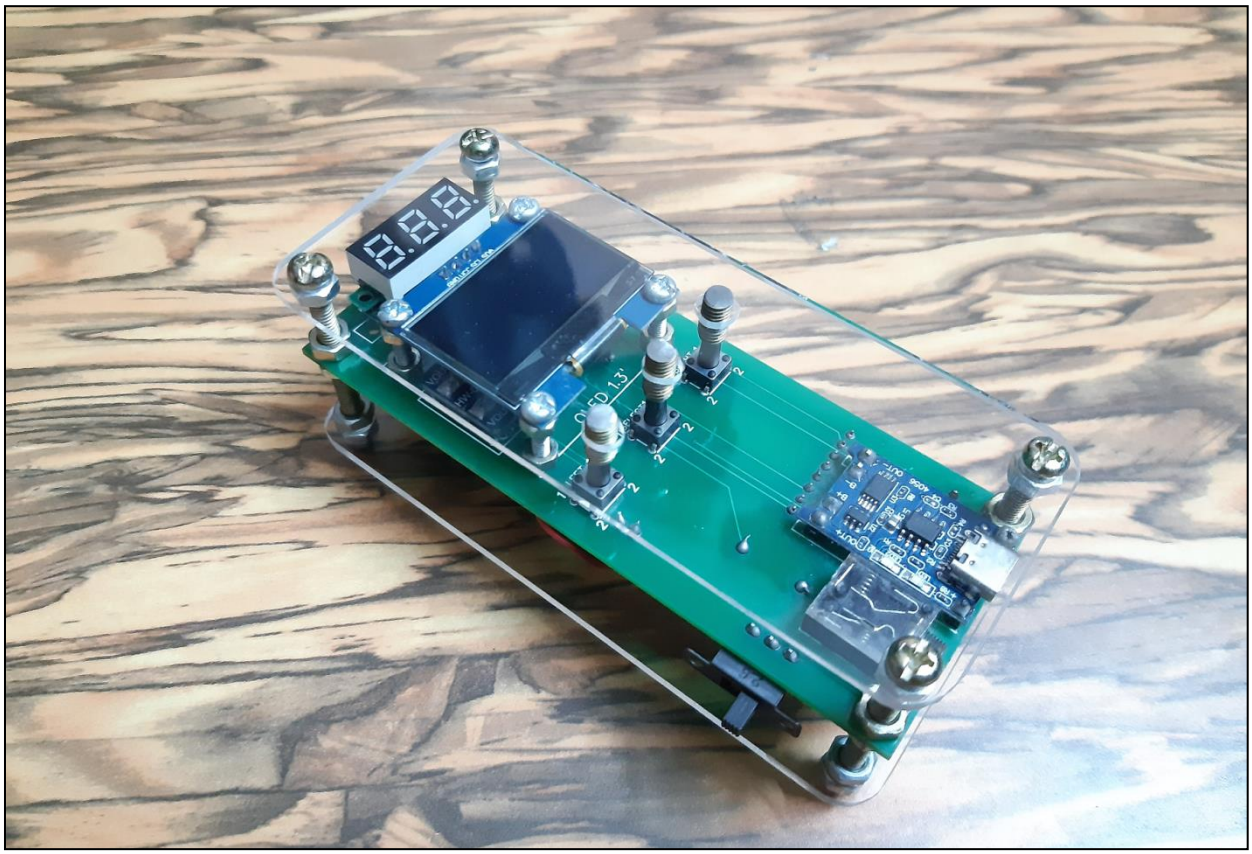
10.3 Appendix C : Budget

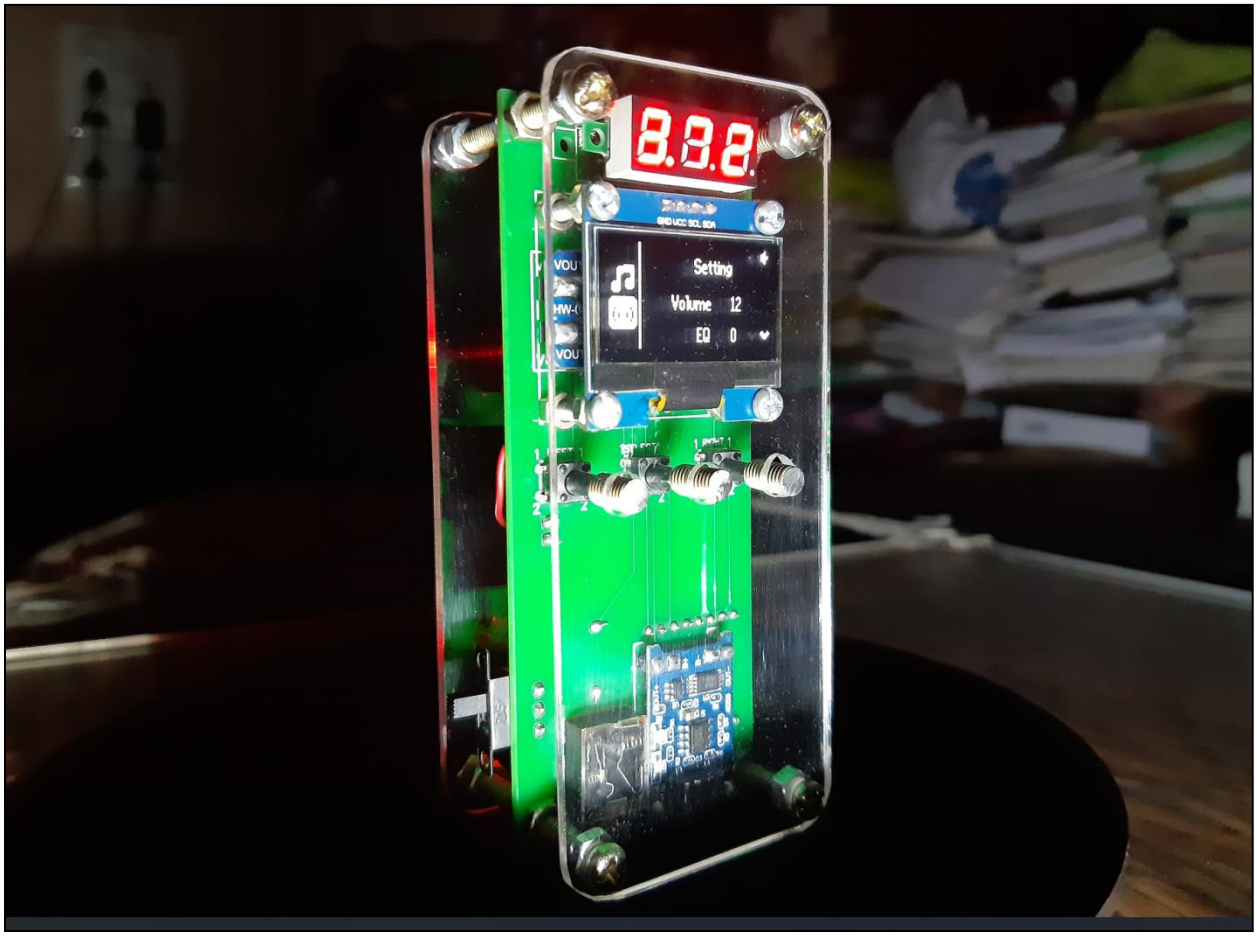
SL. No.	Component Names	Quantity	Prices	
			Predicted	Actual
1	Arduino Pro Mini (5v - 16Mhz)	x 1	400	200
2	FTDI USB to TTL Module	x 1	150	130
3	MT3608 / XL6009 Step Up Module	x 1	60	50
4	DF Player Mini MP3 player	x 1	200	100
5	1.3 inch Oled Display	x 1	400	230
6	DC Voltmeter	x 1	90	70
7	TP 4056 charging Module - Type C	x 1	50	25
8	3.7V Mini Lipo Battery	x 1	90	80
9	M to M Jumper Wire Set	x 1	40	40
10	Both side Vero Board (Medium)	x 1	60	100
11	Earphone	x 1	50	100
12	16 GB Memory TF Card	x 1	400	375
13	Acrylic Sheet Transparent	x 1	100	60
14	ECOMS PCB	x 1	1310	1310
15	Push Button - Micro Switch (Long Nose)	x 3	30	15
16	Wooden box for carrying	x 1	350	340
Total Price =			3780	3225

10.4 Appendix D – Work Plan

Work processes step by step	Months																			
	January				February				March				April				May			
	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4	Week 1	Week 2	Week 3	Week 4
Choosing Platform																				
Selection of Project																				
Background research																				
Designing in software																				
Purchasing components																				
Test & Study of the parts																				
Assembly in breadboard																				
Coding																				
Testing																				
Design PCB in EasyEDA																				
Ordering PCB																				
Assembly on PCB																				
Enclosure making																				
Final Testing																				
Preparing project report																				
Preparing presentation																				
Presenting the Prototype																				

10.5 Appendix E : Picture of Project Prototype





10.6 Appendix F : Picture of Project Prototype – Music Menu

10.7 Appendix G : Picture of Project Prototype – Settings / EQ + Volume Menu