

Coin Based EV Charging Station

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in
ELECTRONICS AND COMMUNICATION ENGINEERING
BY

ASHWANTH PUSHPAKIRAN P A (REG.NO :38130249)
M D NAVEEN (REG.NO : 38130150)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING SCHOOL OF ELECTRICAL AND ELECTRONICS
ENGINEERING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MAY-2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Ashwanth Pushpakiran (Reg no:38130249), M. D Naveen (Reg no:38130150) who carried out the project entitled "Coin Based EV Charging Station" Under our supervision from December 21 to April 22.

Internal Guide

(Dr. P Chitra)

External Guide

(Dr.R.PANDIAN)

Head of the Department

Dr. T.RAVI M.E., Ph.D.

Submitted for Viva voce Examination held on 6.5.2022

Internal Examiner

External Examiner

DECLARATION

Ashwanth Pupakiran P A and M D NAVEEN , hereby declare that the Project Report entitled “Coin Based EV Charging Station” done by us under the guidance of Dr.P.CHITRA at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in ELECTRONICS AND COMMUNICATIONENGINEERING.

DATE:

SIGNATURE OF THE CANDIDATE

PLACE:

ACKNOWLEDGEMENT

We are pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr .N.M.NANDHITHA, M.E., Ph.D.**, Prof & Dean of school of Electrical and Electronics Engineering and **Dr .T.RAVI, M.E., Ph.D.**, Heads of the Department, Department of Electronics and communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **DR .P.CHITRA, M.E., Ph.D.**, for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff and also panel members of the Department of Electronics and communication engineering who were helpful in many ways for the completion of the project.

ABSTRACT

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

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	8-13
	LIST OF FIGURES	
	LIST OF SYMBOLS	
	LIST OF ABBREVIATIONS	
	LIST OF TABLES	
1.	CHAPTER 1 : INTRODUCTION	14-19
	1.1 GENERAL	
	1.2 SCOPE OF THE PROJECT	
	1.3 EXISTING SYSTEM	
	1.3.1 EXISTING SYSTEM DISADVANTAGES	
	1.4 LITERATURE SURVEY	
	1.5 PROPOSED SYSTEM	
	1.5.1 PROPOSED SYSTEM ADVANTAGES	
2.	CHAPTER 2 : PROJECT DESCRIPTION	20-24
	2.1 GENERAL	
	2.2 BLOCK DIAGRAM	
	2.3 MODULES NAME	
	2.4 MODULE DESCRIPTION	
	2.4.1 COIN ACCEPTANCE WORKING	
	2.4.2 IDENTIFICATION	
	2.5 BLOCK DIAGRAM DESCRIPTION	

3.	CHAPTER 3 : HARDWARE AND SOFTWARE DESCRIPTION	26-136
	3.1HARDWARE DESCRIPTION	
	3.1.1 ARDUINO MEGA	
	3.1.2 POWER SUPPLY	
	3.1.3 LCD	
	3.1.4 KEYPAD	
	3.1.5 RELAY	
	3.1.6 RFID READER	
	3.1.7 RFID TAG	
	3.1.8 COIN ACCEPTER MODULE	
	3.1.9 GSM	
	3.1.10BUZZER	
	3.2 SOFTWARE DESCRIPTION	
	3.2.1 EMBEDDED C	
	3.2.2 ARDUINO SOFTWARE IDE	
4.	CHAPTER 4 : OUTPUT	137
	4.1 OUTPUT	
5.	CHAPTER 5 : CONCLUSION	138-140
	5.1 APPLICATIONS	
	5.2 FUTURE ENHANCEMENTS	

5.3 ADVANTAGES

5.4 CONCLUSION

5.5 REFERENCES

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO.
1	Block Diagram of the Hardware Project	21
2	COIN ACCEPTOR MODULE	22
3	ARDUINO MEGA	29
4	ARDUINO MEGA PIN DIAGRAM	38
5	LCD PIN DIAGRAM	50
6	LCD TYPE HD44780 PIN DIAGRAM	51
7	KEYPAD	566
8	RELAY	60
9	DIFFERENT TYPES OF RELAY	60
10	DC VS AC RELAY	62
11	ATTRACTION TYPE RELAY	63
12	INDUCTION TYPE RELAY	63
13	SOLID STATE RELAY	64
14	REED RELAY	65

15	RFID TAG	68
16	RFID READER	70
17	GSM NETWORKING ELEMENTS	87
18	NETWORK AREAS	87
19	LOCATION AREAS	88
20	MSC/VLR SERVICE AREA	88
21	PLMN NETWORK AREA	88
22	GSM MODEM	94
23	BUZZER	98
24	BASIC STRUCTURE OF EMBEDDED SYSTEM	111
25	EMBEDDED SYSTEM TOOLS AND PERIPHERALS	112

LIST OF SYMBOLS

NOTATION

S.NO	NAME	NOTATION	DESCRIPTION
1	\$	DOLLAR	US CURRENCY
2	I2C	INTER INTEGRATED CIRCUITS	I2C alternatively known as I2C or IIC, is a synchronous, multi- controller/multi-target
3	RX-TX	RECIEVE AND TRANSMIT	Recieve from this server and transmit to other server
4	°C	DEGREE CELCIUS	Denoting temperature
5	dB	DECIBEL	MEASURING UNITS
6	cm	CENTIMETER	Measuring Unit
7	KHz	KILO HERTZ	Unit of alternating Current

LIST OF ABBREVIATION

S.NO	ABBREVIATION	EXPANSION
1	EV	Electronic Vehicle
2	V2G	Vehicle to Grid
3	V	Voltage
4	LCD	Liquid Crystal Display
5	GSM	Global System for Mobile Communication
6	IDE	Integrated Development Environment
7	ASCII	American Standard Code for Information Interchange
8	NC	Normally Close
9	NO	Normally open
10	RFID	Radio Frequency Identification
11	AIDC	Automatic Identification and data capture
12	BAP	Battery Assisted Passive
13	RF	Radio Frequency
14	PRAT	Passive Reader Active Tag

15	ARPT	Active Reader Passive tag
16	SS	Switching System
17	HLR	Home Location register
18	MSC	Mobile Services Switching Center
19	VLR	Visitors Location Register
20	AUC	Authentication Center
21	EIR	Equipment Identity Register
22	BSS	Base Station System
23	OMC	Operation and Maintenance center
24	OSS	Operation and Support system
25	MXE	Message Center
26	MSN	Mobile Service Node
27	GMSC	Gateway Mobile Services Switching Center
28	GIWU	GSM Interworking Unit
29	DTR	Data Terminal ready

LIST OF TABLES

TABLE NO.	DESCRIPTION
1	Technical Specification of Arduino Mega
2	Arduino Mega 2560 Pin Mapping table
3	Pin Description of LCD Display
4	Character LCD type HD44780 Pin diagram
5	Frequency Bands
6	Development Mobile telephone Systems
7	GSM Milestones
8	Pin Specifications for GSM Modem

CHAPTER: 1 - INTRODUCTION

1.1 GENERAL

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

SCOPE OF THE PROJECT:

The aim of our project is a coin based EV charging station

1.2 EXISTING SYSTEM

- In this system, there is the only option to access the device is using card.
- In the process of charging time we can't disconnect the charger for the emergency purpose.
- . If we disconnect the charger we lost the particular amount of money.

1.2.1 EXISTING SYSTEM DISADVANTAGES

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

1.3 LITERATURE SURVEY

Title 1: Smart Electric Vehicle Charging System

Authors João C. Ferreira, Vítor Monteiro, João L. Afonso, Alberto Silva
Member, IEEE

Year: 2011

Description:

In this work is proposed the design of a system to create and handle Electric Vehicles (EV) charging procedures, based on intelligent process. Due to the electrical power distribution network limitation and absence of smart meter devices, Electric Vehicles charging should be performed in a balanced way, taking into account past experience, weather information based on data mining, and simulation approaches. In order to allow information exchange and to help user mobility, it was also created a mobile application to assist the EV driver on these processes. This proposed Smart Electric Vehicle Charging System uses Vehicle-to-Grid (V2G) technology, in order to connect Electric Vehicles and also renewable energy sources to Smart Grids (SG). This system also explores the new paradigm of Electrical Markets (EM), with deregulation of electricity production and use, in order to obtain the best conditions for commercializing electrical energy

Title2: Development of an iot system with smart charging current control for electric vehicle

Authors: ruban a.sousa, victor monteiro, joao c. Ferreira, andres a. Nogueiras melendez, joao l. Afonso, jose a. Afonso

Year: 2018

Description:

The project is named as "DIGITALLY SENSE HEART RATE AND BODY TEMPERATURE USING ARDUINO". With the development of technology, in this project we can digitally sensing body temperature and heart rate using Arduino. Mainly Arduino is used because it can sense the environment by receiving input from variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language". LM35 is used for the sense body temperature. Body temperature is a basic parameter for monitoring and diagnosing human health. Heart beat sensor was used for sensing heart rate. This device will allow one to measure their mean arterial pressure (MAP) in about one minute and the accurate body temperature will be displayed on the Android. The system can be used to measure physiological parameters, such as Heart rate (Systolic and Diastolic), Pulse rate.

Title 3: An Intelligent Charging Control Method for Electric Vehicle Charging System

Authors: Liu Zheng, Fan Shao-sheng

Year: 2016

Description:

In this paper, an intelligent double closed loop controller for electric vehicle (EV) charging system in smart grid (SG) is proposed, in which it can perform seamless changeover from constant current (CC) to constant voltage (CV) charging stage. Moreover, it can perform CC and CV charging control separately without adding in an external logic. Experimental results will be provided in full paper submission in order to verify the proposed controller in comparison to the conventional one.

Title 4: COIN BASED MOBILE BATTERY CHARGER WITH HIGH SECURITY

Authors: Aliae Squalli Houssaini, Hassan Qjidaa

Year: 2019

Description:

Recently, driver hypovigilance (drowsiness and fatigue) becomes one of the principal causes of traffic crashes, it can prompt many deaths, wounds and many economic losses. Therefore, the use of a system that takes into account the driver's level of vigilance can play an important role in preventing accidents

and saving human lives. In this work, we propose a non-intrusive driver hypovigilance detection system in real-time. This system makes it possible to detect drowsiness by the identification of MicroSleep corresponding to a sleepiness of more than 2 seconds through the analysis of eye-closure, and to identify fatigue by the analysis of the movement of the mouth to detect yawning. In case of drowsiness or fatigue, an alert is launched to make the driver vigilant and thus definitely avoid road accidents, decrease the percentage of murders and injuries caused by driver hypovigilance, then save many human lives. Experiments were conducted in real time to evaluate the proposed approach.

Title 5: Automatic Gadget Charger using Coin Detection

Authors: Gunjan Chhabra, Sunil Kumar and Pankaj Badoni

Year: 2015

Description:

In this communication era, mobile -telephone industry has grown profoundly. The urban population uses the latest mobile-phone technology while the rural population buys second hand ones, mostly with degraded battery, that require frequent charging. This battery-problem becomes a giant when user doesnt have a standard charger or an electricity connection. In this paper, researchers intent is to propose a public coin based mobile battery charging system. By using image processing techniques, value of the coin has been detected for a limited time, it will charge the device accordingly. A suitable microcontroller is programmed for all the controlling applications. The source for charging is obtained either from a direct power grid or by solar

energy.

1.4

PROPOSED SYSTEM

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

1.4.1

PROPOSED SYSTEM ADVANTAGES

- ✓ Option to use card and carless.
- ✓ No Loss Money.
- ✓ Return policy for the unused charging time

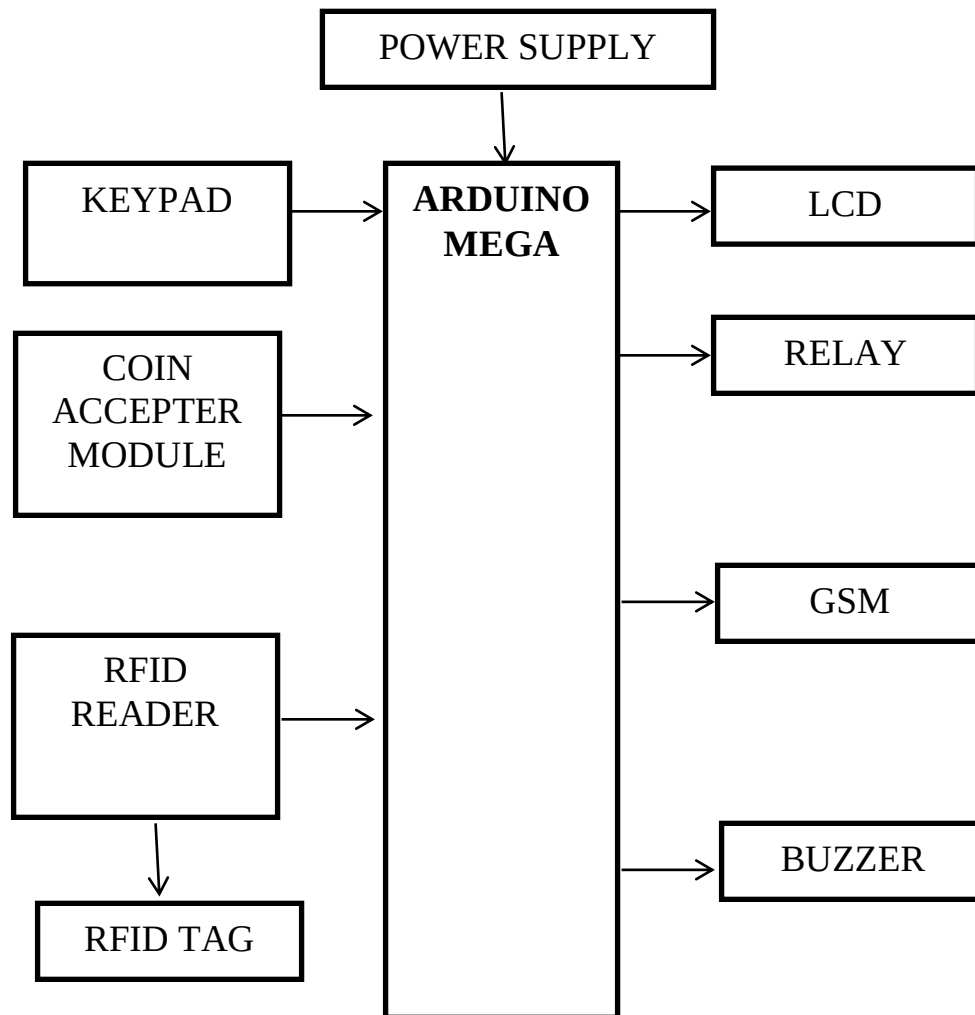
CHAPTER: 2

PROJECT DESCRIPTION

2.1 GENERAL

The coin based mobile charging system charges the mobile phones when the coin is inserted. This system is used by shop owners, rural people and can be implemented in the public places like railway stations, bus stand to provide mobile charging facility. So the coin acceptor recognizes valid coins and then signals the arduino for further action. If a valid coin is found, it signals the arduino and then arduino starts the mobile charging mechanism providing a 5V supply through a power supply section to the mobile phone. The arduino starts a reverse countdown timer to display the charging time for that mobile phone. Further the user adds another coin, the arduino adds to the currently remaining time and once again decrements the countdown. This system can be used for smart mobile charging at public places. This coin based mobile charging system will supply the enough amount of charge to the mobile phone and is available on demand in public places.

1.2 BLOCK DIAGRAM

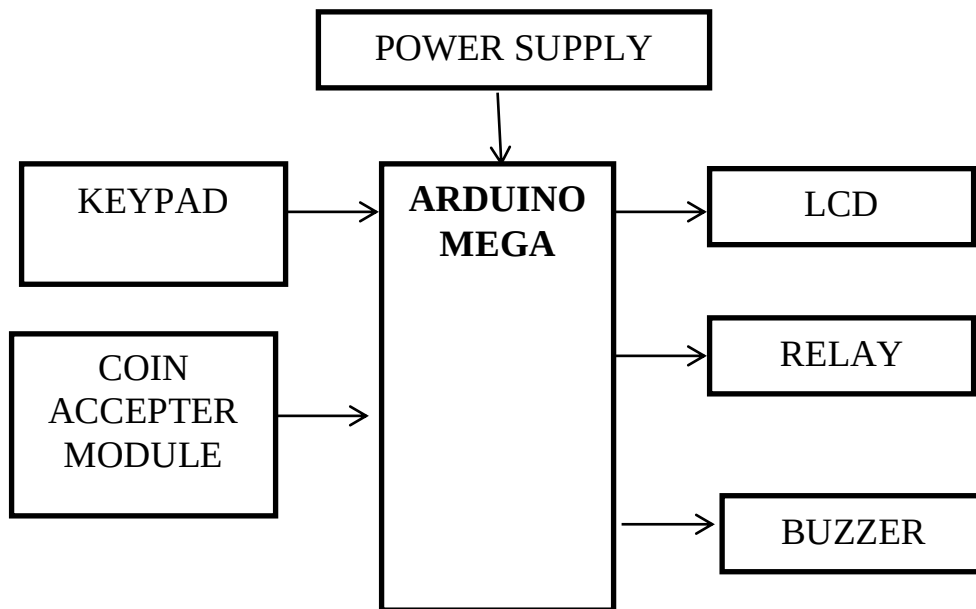


MODULES NAME

- ❖ COIN ACCEPTER MODULE WORKING
- ❖ IDENTIFICATION

MODULE DESCRIPTION

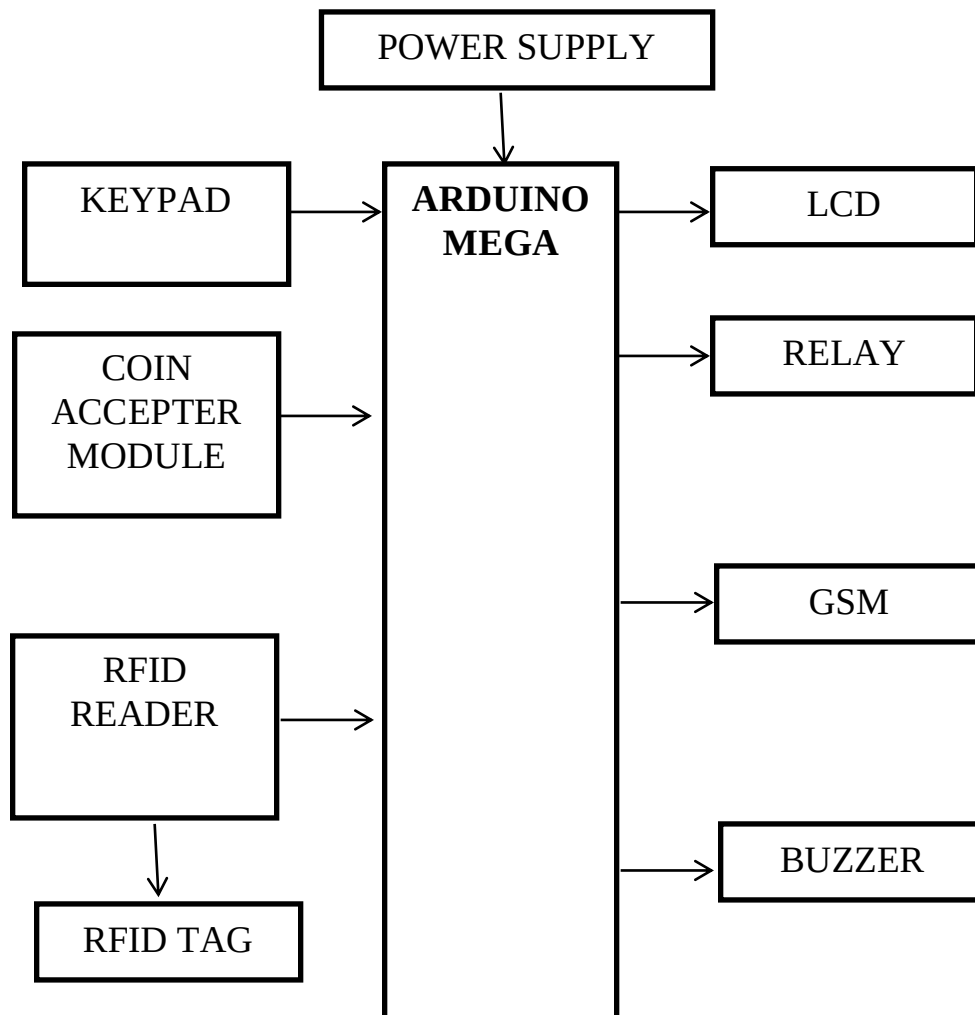
❖ COIN ACCEPTER MODULE WORKING



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

❖ IDENTIFICATION

Using rfid to identify next time you charging, the every time remaining amount will be save rfid card every time. So, we use that money alternate time for charging.



BLOCK DIAGRAM WORKING

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

CHAPTER: 3

HARDWARE AND SOFTWARE DESCRIPTION

3.1 HARDWARE DESCRIPTION:

3.1.1 ARDUINO MICRO-CONTROLLER

INTRODUCTION TO ARDUINO:

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

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

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. All

Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

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.

ARDUINO MEGA:

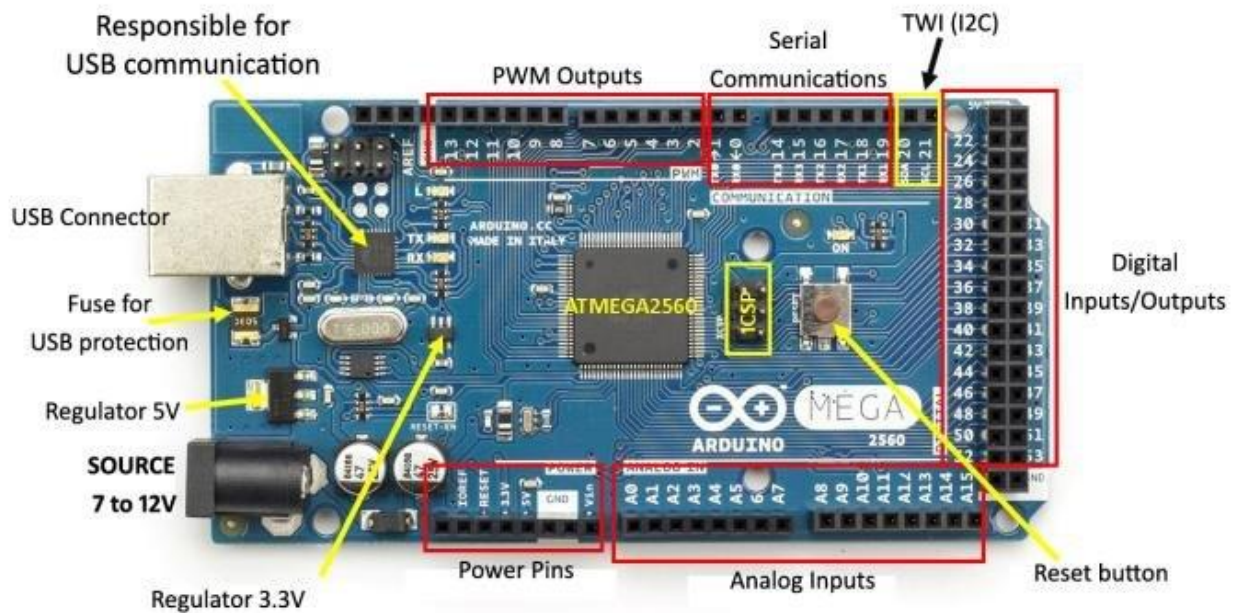
The MEGA 2560 is designed for more complex projects. With 54 digital I/O pins, 16 analog inputs and a larger space for your sketch it is the recommended board for 3D printers and robotics projects. This gives your projects plenty of room and opportunities.



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

TECHNICAL SPECIFICATIONS:

Microcontroller	<u>ATmega2560</u>
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by boot loader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g



HARDWARE:

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

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released

have avoided the project name by using various names ending in -duino. An early Arduino board with an RS-232 serial interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are at the top, the 6 analog input pins at the lower right, and the power connector at the lower left.

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

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

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

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

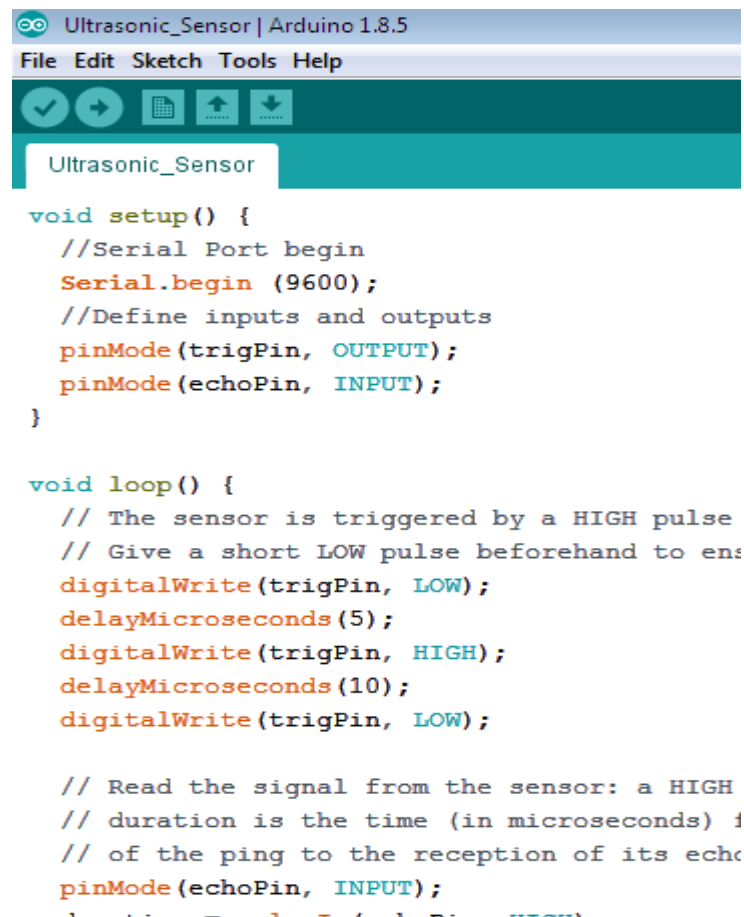
Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically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

PROGRAMMING:

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

The ATmega2560 on the Mega 2560 comes preprogrammed with a [boot loader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using [Arduino ISP](#) or similar; see these [instructions](#) for details.

A screenshot of the Arduino IDE interface. The title bar reads "Ultrasonic_Sensor | Arduino 1.8.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a grid, an upload arrow, and a download arrow. A tab labeled "Ultrasonic_Sensor" is active. The main text area contains the following C++ code:

```
void setup() {  
  //Serial Port begin  
  Serial.begin (9600);  
  //Define inputs and outputs  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);  
}  
  
void loop() {  
  // The sensor is triggered by a HIGH pulse  
  // Give a short LOW pulse beforehand to ens  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(5);  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  
  // Read the signal from the sensor: a HIGH  
  // duration is the time (in microseconds) f  
  // of the ping to the reception of its ech  
  pinMode(echoPin, INPUT);  
  // duration = digitalRead(echoPin) * 54  
}
```

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

- **On Rev1 boards:** connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- **On Rev2 or later boards:** there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU boot loader). See [this user-contributed tutorial](#) for more information.

WARNINGS:

The Mega 2560 has a resettable poly fuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

POWER:

The Mega 2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **Vin.** The input voltage to the board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins

bypasses the regulator, and can damage your board. We don't advise it.

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

MEMORY:

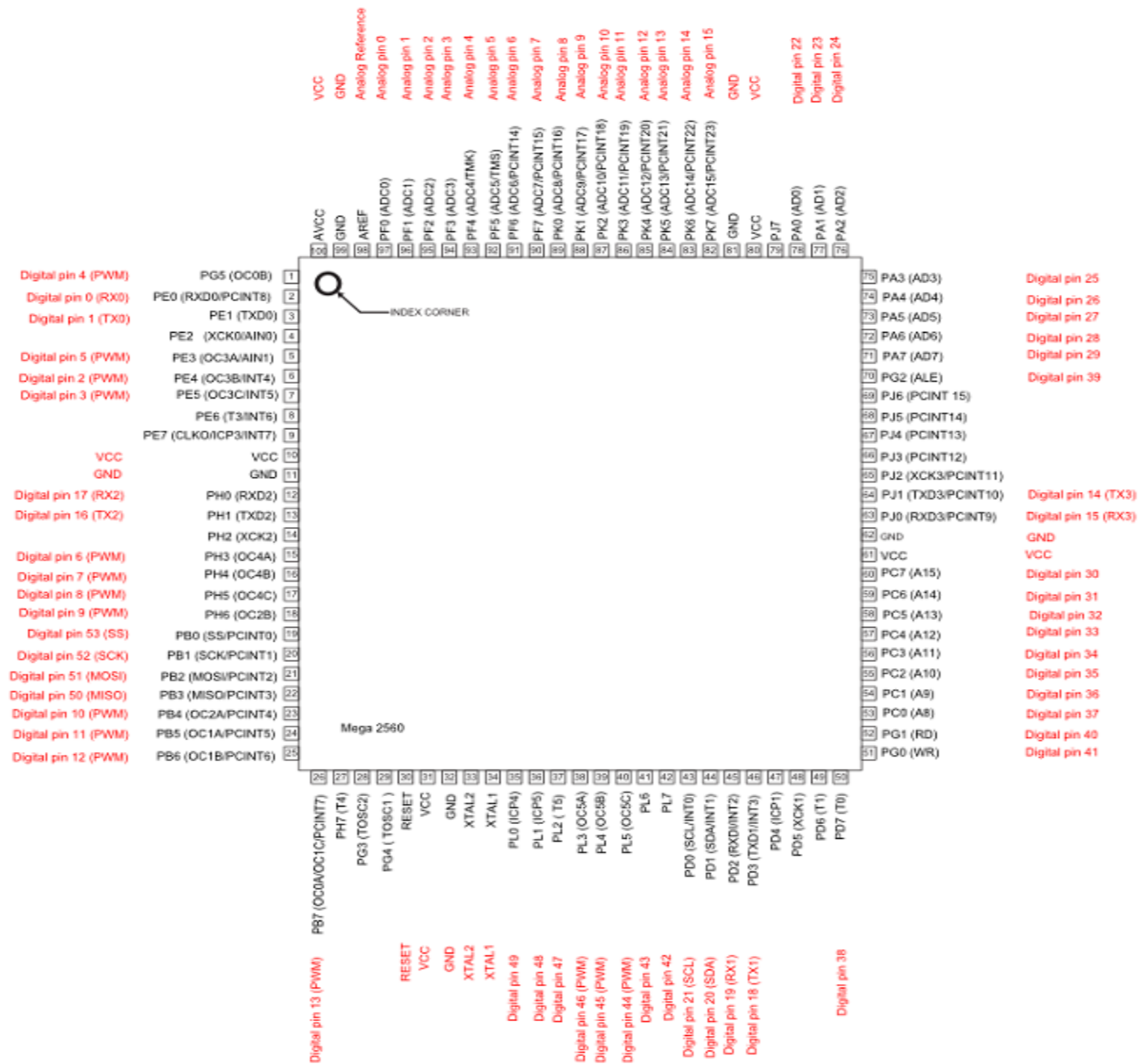
The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

INPUT AND OUTPUT:

ATmega2560-Arduino Pin Mapping.

Below is the pin mapping for the Atmega2560. The chip used in Arduino 2560. There are pin mappings to Atmega8 and Atmega 168/328 as well.

Arduino Mega 2560 PIN diagram



Arduino Mega 2560 PIN mapping table

Pin Number	Pin Name	Mapped Pin Name
1	PG5 (OC0B)	Digital pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)

3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLK0/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8 (PWM)
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	

29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36

55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC

81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

Each of the 54 digital pins on the Mega can be used as an input or output, using **pinMode()**, **digitalWrite()**, and **digitalRead()** functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the

value that must not be exceeded to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

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

- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **TWI:** 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the old Duemilanove or Diecimila Arduino boards.

The Mega 2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

COMMUNICATION:

The Mega 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2 (ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer

(Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega 2560's digital pins.

The Mega 2560 also supports TWI and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library.

PHYSICAL CHARACTERISTICS AND SHIELD COMPATIBILITY:

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

The Mega 2560 is designed to be compatible with most shields designed for the Uno and the older Diecimila or Duemilanove Arduino boards. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent

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

AUTOMATIC (SOFTWARE) RESET:

Rather than requiring a physical press of the reset button before an upload, the Mega 2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega 2560 board is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the ATmega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when

it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

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

REVISIONS:

The Mega 2560 does not use the FTDI USB-to-serial driver chip used in past designs. Instead, it features the ATmega16U2 (ATmega8U2 in the revision 1 and revision 2 Arduino boards) programmed as a USB-to-serial converter. Revision 2 of the Mega 2560 board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into [DFU mode](#).

Revision 3 of the Arduino board and the current Genuino Mega 2560 have the following improved features:

- **1.0 pinout:** SDA and SCL pins - near to the AREF pin - and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the board that uses ATSAM3X8E, that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.

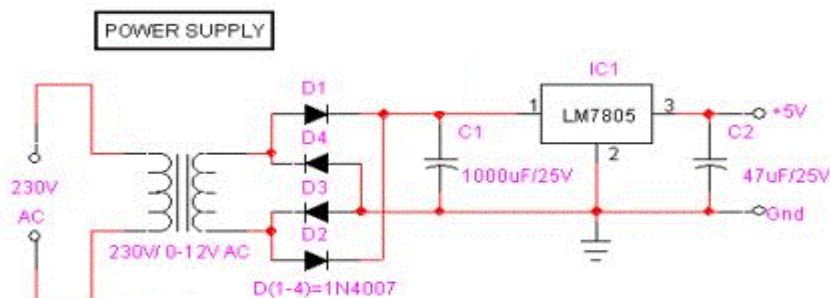
- Atmega 16U2 replace the 8U2.

APPLICATIONS:

- Arduboy, a handheld game console based on Arduino.
- Arduinome, a MIDI controller device that mimics the Monome.
- Ardupilot, drone software and hardware.
- ArduSat, a cubesat based on Arduino.
- C-STEM Studio, a platform for hands-on integrated learning of computing, science, technology, engineering, and mathematics (C-STEM) with robotics.
- Data loggers for scientific research.
- OBDDuino, a trip computer that uses the on-board diagnostics interface found in most modern cars.

3.1.2_Power Supply:

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



The power supply section is the important one. It should deliver constant output regulated power supply for successful working of the project. A 0-12V/1 mA transformer is used for this purpose. The primary of this transformer is connected in to main supply through on/off switch& fuse for protecting from

overload and short circuit protection. The secondary is connected to the diodes to convert 12V AC to 12V DC voltage. And filtered by the capacitors, Which is further regulated to +5v, by using IC 7805.

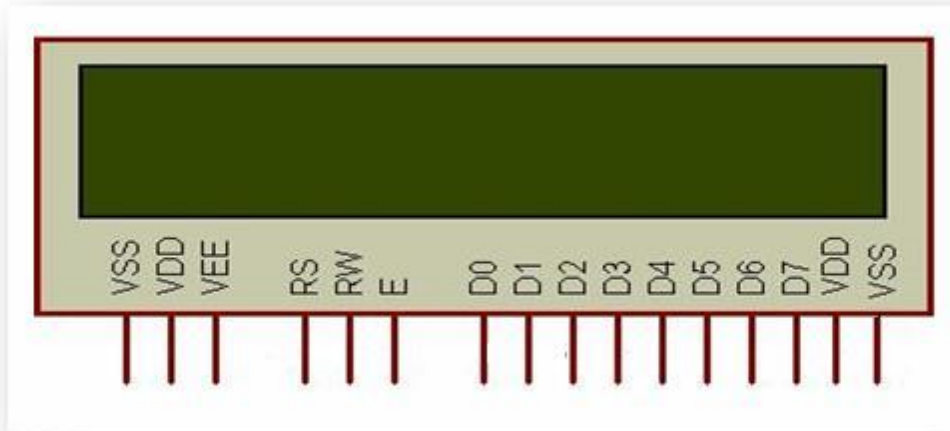
3.1.3 LCD (Liquid Crystal Display)

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

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

Pin Diagram:



Pin Description:

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	V _{cc}
3	Contrast adjustment; through a variable resistor	V _{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7

15	Backlight V _{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Introduction

The most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580. In this tutorial, we will discuss about character based LCDs, their interfacing with various microcontrollers, various interfaces (8-bit/4-bit), programming, special stuff and tricks you can do with these simple looking LCDs which can give a new look to your application.

Pin Description

The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters, whereas LCDs supporting more than 80 characters make use of 2 HD44780 controllers.

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections). Pin description is shown in the table below.

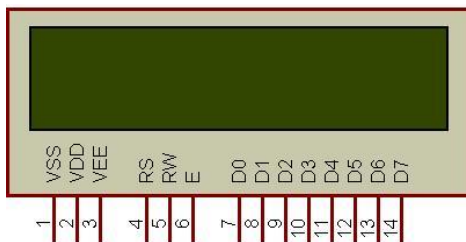


Figure 1: Character LCD type HD44780 Pin diagram

Pin No.	Name	Description
Pin no. 1	D7	Data bus line 7 (MSB)
Pin no. 2	D6	Data bus line 6
Pin no. 3	D5	Data bus line 5

Pin no. 4	D4	Data bus line 4
Pin no. 5	D3	Data bus line 3
Pin no. 6	D2	Data bus line 2
Pin no. 7	D1	Data bus line 1
Pin no. 8	D0	Data bus line 0 (LSB)
Pin no. 9	EN1	Enable signal for row 0 and 1 (1 st controller)
Pin no. 10	R/W	0 = Write to LCD module 1 = Read from LCD module
Pin no. 11	RS	0 = Instruction input 1 = Data input
Pin no. 12	VEE	Contrast adjust
Pin no. 13	VSS	Power supply (GND)
Pin no. 14	VCC	Power supply (+5V)
Pin no. 15	EN2	Enable signal for row 2 and 3 (2 nd controller)
Pin no. 16	NC	Not Connected

Table 2: Character LCD pins with 2 Controller

Usually these days you will find single controller LCD modules are used more in the market. So in the tutorial we will discuss more about the single controller LCD, the operation and everything else is same for the double controller too. Lets take a look at the basic information which is there in every LCD.

BF - Busy Flag:

Busy Flag is an status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing. To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

Instruction Register (IR) and Data Register (DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. when send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD. We will discuss more on LCD instruction set further in this tutorial.

Commands and Instruction set

Only the instruction register (IR) and the data register (DR) of the LCD can be controlled by the MCU. Before starting the internal operation of the LCD, control information is temporarily stored into these registers to allow interfacing with various MCUs, which operate at different speeds, or various peripheral control devices. The internal operation of the LCD is determined by signals sent from the MCU. These signals, which include register selection signal (RS), read/write signal (R/W), and the data bus (DB0 to DB7), make up the LCD instructions (Table 3). There are four categories of instructions that:

- Designate LCD functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM
- Perform miscellaneous functions

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

Below is a brief list of useful commands which are used frequently while working on the LCD.

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1

16	Set DDRAM address or cursor position on display	0x80+add* 128+add*
17	Set CGRAM address or set pointer to CGRAM location	0x40+add** 64+add**

Table 4: Frequently used commands and instructions for LCD

* DDRAM address given in LCD basics section see Figure 2,3,4

** CGRAM address from 0x00 to 0x3F, 0x00 to 0x07 for char1 and so on.

LCD LIQUID CRYSTAL DISPLAYS:

The LCD standard requires 3 control lines and 8 I/O lines for the data bus.

• 8 data pins D7:D0

Bi-directional data/command pins.

Alphanumeric characters are sent in ASCII format.

• RS: Register Select

RS = 0 -> Command Register is selected

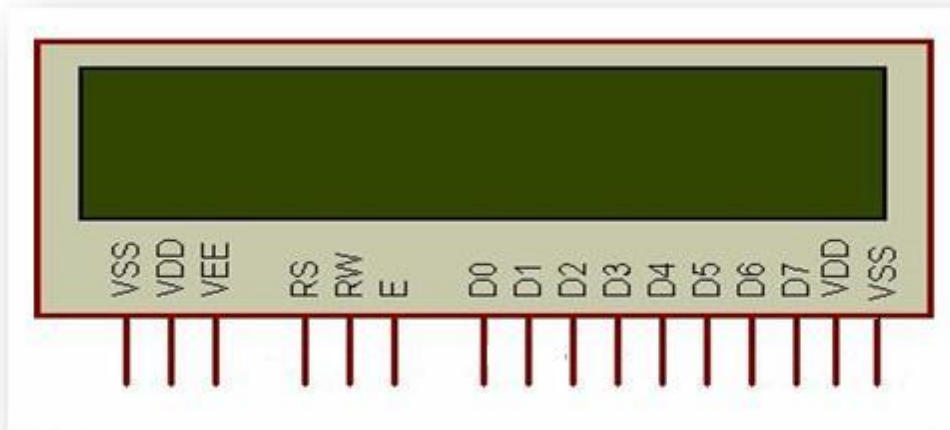
RS = 1 -> Data Register is selected

• R/W: Read or Write

0 -> Write, 1 -> Read

• E: Enable (Latch data)

The 8 data lines are connected to PORT 1 of 8051 microcontroller. The three control lines (RS, RW and EN) are connected to PORT 3.5,3.6 and 3.7 respectively.

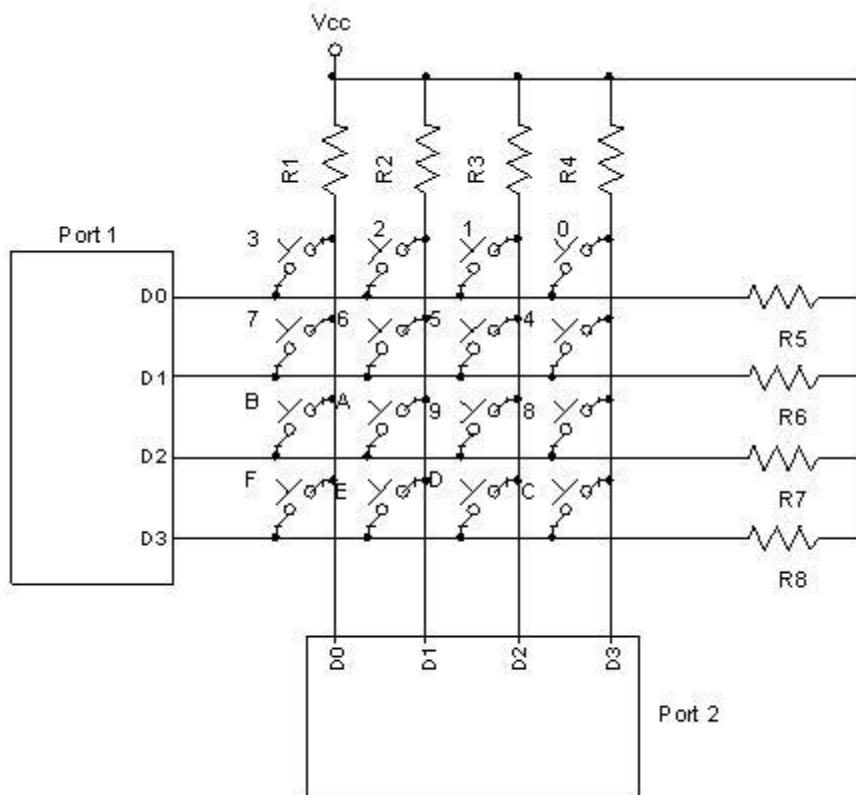


3.1.4 Keypad Interfacing with the Microcontrollers:

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



Figure shows a 4*4 matrix connected to two ports. The rows are connected to an output port and the columns are connected to an input port. If no key has been pressed, reading the input port will yield 1s for all columns since they are all connected to high (V_{cc}). If all the rows are grounded and a key is pressed, one of the columns will have 0 since the key pressed provides the path to ground. It is the function of the microcontroller to scan the keyboard continuously to detect and identify the key pressed.



Grounding rows and reading columns

To detect a pressed key, the microcontroller grounds all rows by providing 0 to the output latch, and then it reads the columns. If the data read from the columns is $D3-D0=1111$, no key has been pressed and the process continues until a key press is detected. However, if one of the column bits has a zero, this means that a key press has occurred. For example, if $D3-D0=1101$,

this means that a key in the D1 column has been pressed. After a key press is detected, the microcontroller will go through the process of identifying the key. Starting with the top row, the microcontroller grounds it by providing a low to row D0 only; then it reads the columns. If the data read is all 1s, no key in that row is activated and the process is moved to the next row. It grounds the next row, reads the columns, and checks for any zero. This process continues until the row is identified. After identification of the row in which the key has been pressed, the next task is to find out which column the pressed key belongs to. This should be easy since the microcontroller knows at any time which row and column are being accessed.

In the program, it is assumed that P1 and P2 are initialized as output and input, respectively. Program13.1 goes through the following four major stages:

- 1) To make sure that the preceding key has been released, 0s are output to all rows at once, and the columns are read and checked repeatedly until all the columns are high. When all columns are found to be high, the program waits for a short amount of time before it goes to the next stage of waiting for a key to be pressed.
- 2) To see if any key is pressed, the columns are scanned over and over in an infinite loop until one of them has a 0 on it. Remember that the output latches connected to rows still have their initial zeros (provided in stage 1), making them grounded. After the key press detection, it waits 20ms for the bounce and then scans the columns again. This serves two functions: (a) it ensures that the first key press detection was not an erroneous one due to spike noise, and (b) the 20ms delay prevents the same key press from being interpreted as a multiple key press. If after the 20-ms delay the key is still

pressed, it goes to the next stage to detect which row it belongs to; otherwise, it goes back into the loop to detect a real key press

3) To detect which row the key press belongs to, it grounds one row at a time, reading the columns each time. If it finds that all columns are high, this means that the key press cannot belong to that row; therefore, it grounds the next row and continues until it finds the row the key press belongs to. Upon finding the row that the key press belongs to, it sets up the starting address for the look-up table holding the scan codes (or the ASCII value) for that row and goes to the next stage to identify the key.

4) To identify the key press, it rotates the column bits, one bit at a time, into the carry flag and checks to see if it is low. Upon finding the zero, it pulls out the ASCII code for that key from the look-up table; Otherwise, it increments the pointer to point to the next element of the look-up table.

While the key press detection is standard for all keyboards, the process for determining which key is pressed varies. The look-up table method shown in program can be modified to work with any matrix up to 8*8.

3.1.5 RELAY:

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



RELAY

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

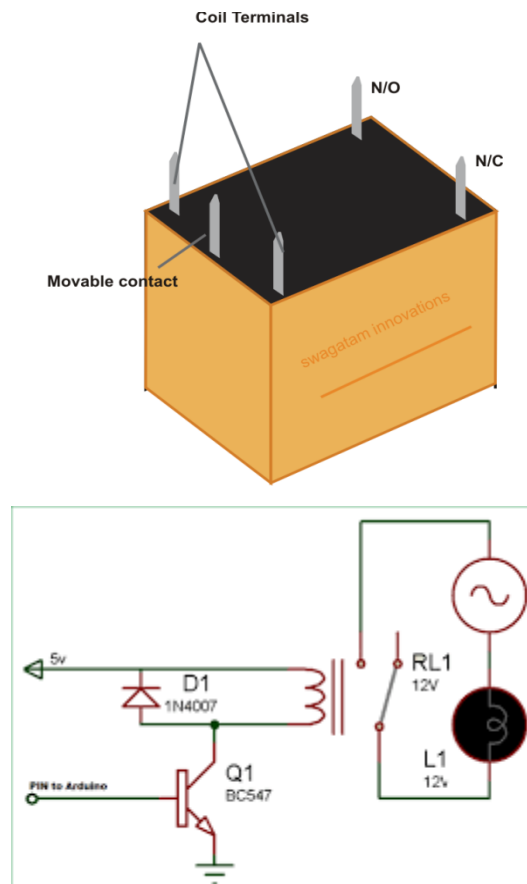
Protective relays continuously monitor these parameters: voltage, current, and power; and if these parameters violate from set limits they generate alarm or isolate that particular circuit. These types of relays are used to protect equipment like motors, generators, and transformers, and so on.



DIFFERENT TYPES OF RELAYS

Reclosing relays are used to connect various components and

devices within the system network, such as synchronizing process, and to restore the various devices soon after any electrical fault vanishes, and then to connect transformers and feeders to line network. Regulating relays are the switches that contacts such that voltage boosts up as in the case of tap changing transformers.



RELAY CIRCUIT

Auxiliary contacts are used in circuit breakers and other protective equipment for contact multiplication. Monitoring relays monitors the system conditions such as direction of power and accordingly generates the alarm. These are also called directional relays.

DIFFERENT TYPES OF RELAYS:

Depending on the operating principle and structural features relays are of different types such as electromagnetic relays, thermal relays, power

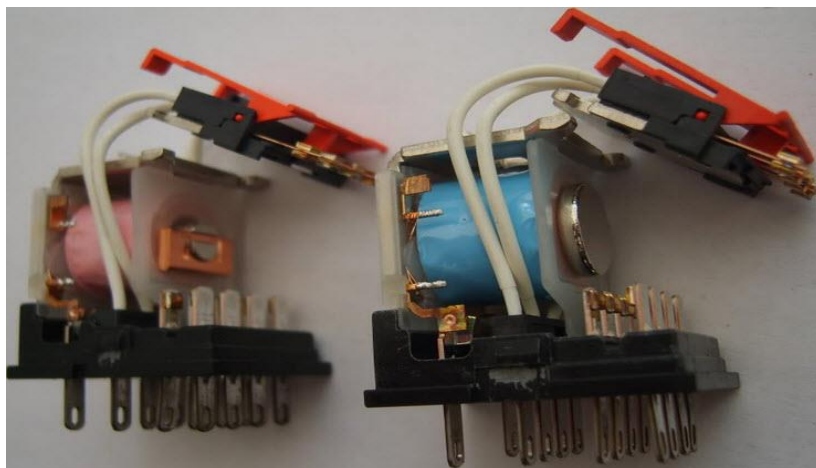
varied relays, multi-dimensional relays, and so on, with varied ratings, sizes and applications.

1. ELECTROMAGNETIC RELAYS:

These relays are constructed with electrical, mechanical and magnetic components, and have operating coil and mechanical contacts. Therefore, when the coil gets activated by a supply system, these mechanical contacts gets opened or closed. The type of supply can be AC or DC.

DC vs AC Relays

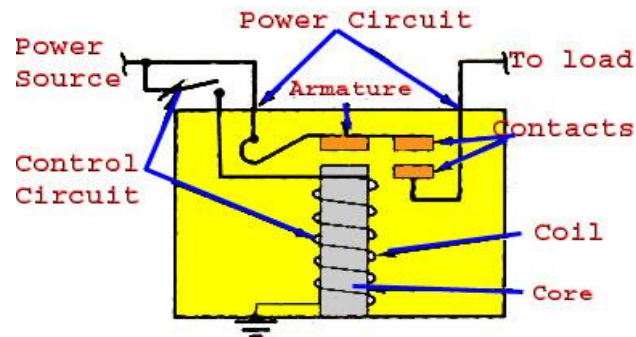
Both AC and DC relays work on the same principle as electromagnetic induction, but the construction is somewhat differentiated and also depends on the application for which these relays are selected. DC relays are employed with a freewheeling diode to de-energize the coil, and the AC relays uses laminated cores to prevent eddy current losses.



DC VS AC RELAYS

The very interesting aspect of an AC is that for every half cycle, the direction of the current supply changes; therefore, for every cycle the coil loses its magnetism since the zero current in every half cycle makes the relay continuously make and break the circuit. So, to prevent this – additionally one shaded coil or another electronic circuit is placed in the AC relay to provide magnetism in the zero current position.

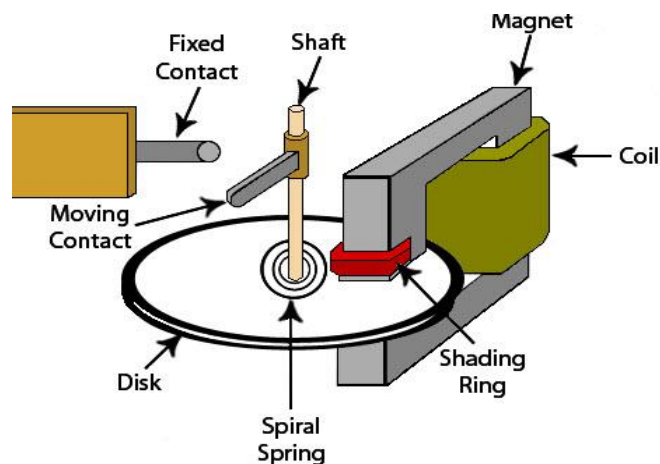
Attraction Type Electromagnetic Relays



ATTRACTION TYPE RELAYS

These relays can work with both AC and DC supply and attract a metal bar or a piece of metal when power is supplied to the coil. This can be a plunger being drawn towards the solenoid or an armature being attracted towards the poles of an electromagnet as shown in the figure. These relays don't have any time delays so these are used for instantaneous operation.

Induction Type Relays These are used as protective relays in AC systems alone and are usable with DC systems. The actuating force for contacts movement is developed by a moving conductor that may be a disc or a cup, through the interaction of electromagnetic fluxes due to fault currents.



INDUCTION TYPE RELAYS

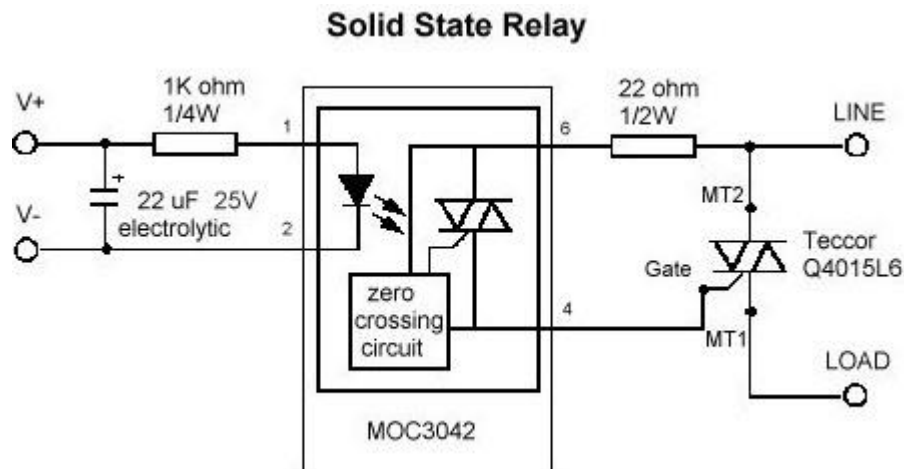
These are of several types like shaded pole, watt-hour and induction cup structures and are mostly used as directional relays in power-system protection and also for high-speed switching operation applications

Magnetic Latching Relays

These relays use permanent magnet or parts with a high remittance to remain the armature at the same point as the coil is electrified when the coil power source is taken away.

2. SOLID STATE RELAYS:

Solid State uses solid state components to perform the switching operation without moving any parts. Since the control energy required is much lower compared with the output power to be controlled by this relay that results the power gain higher when compared to the electromagnetic relays. These are of different types: reed relay coupled SSR, transformer coupled SSR, photo-coupled SSR, and so on.



SOLID STATE RELAYS

The above figure shows a photo coupled SSR where the control signal is applied by LED and it is detected by a photo-sensitive semiconductor device. The output from this photo detector is used to trigger the gate of TRIAC or SCR that switches the load.

3. HYBRID RELAY:

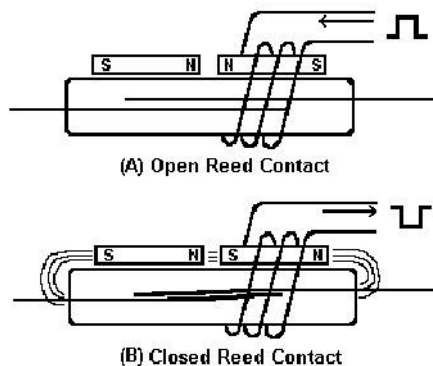
These relays are composed of electromagnetic relays and electronic components. Usually, the input part contains the electronic circuitry

that performs rectification and the other control functions, and the output part include electromagnetic relay.

4. THERMAL RELAY:

These relays are based on the effects of heat, which means – the rise in the ambient temperature from the limit, directs the contacts to switch from one position to other. These are mainly used in motor protection and consist of bimetallic elements like temperature sensors as well as control elements. Thermal overload relays are the best examples of these relays.

5. REED RELAY:



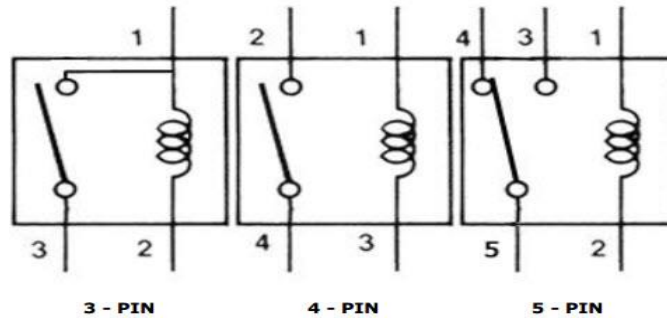
REED RELAY

Reed Relays consist of a pair of magnetic strips (also called as reed) that is sealed within a glass tube. This reed acts as both an armature and a contact blade. The magnetic field applied to the coil is wrapped around this tube that makes these reeds move so that switching operation is performed.

Based on dimensions, relays are differentiated as micro miniature, subminiature and miniature relays. Also, based on the construction, these relays are classified as hermetic, sealed and open type relays. Furthermore, depending on the load operating range, relays are of micro, low, intermediate and high power types.

Relays are also available with different pin configurations like 3

pin, 4 pin and 5 pin relays. The ways in which these relays are operated is shown in the below figure. Switching contacts can be SPST, SPDT, DPST and DPDT types. Some of the relays are normally open (NO) type and the other are normally closed (NC) types.



These are some of the different types of relays that are employed in most of the electronic as well as electrical circuits. The information about the different types of relays serves readers' purpose and we hope that they will find this basic information very useful.

3.1.6 RFID (RADIO-FREQUENCY IDENTIFICATION)

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

RFID is similar to barcoding in that data from a tag or label are captured by a device that stores the data in a database. RFID, however, has several advantages over systems that use barcode asset tracking software. The most notable is that RFID tag data can be read outside the line-of-sight, whereas barcodes must be aligned with an optical scanner. If you are considering implementing an RFID solution, take the next step and contact the RFID experts at AB&R® (American Barcode and RFID).

Active tags have a local power source (such as a battery) and may operate hundreds of meters from the RFID reader. Unlike a barcode, the tags

don't need to be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method of automatic identification and data capture (AIDC).

RFID tags are used in many industries. For example, an RFID tag attached to an automobile during production can be used to track its progress through the assembly line; RFID-tagged pharmaceuticals can be tracked through warehouses; and implanting RFID microchips in livestock and pets enables positive identification of animals.

Since RFID tags can be attached to cash, clothing, and possessions, or implanted in animals and people, the possibility of reading personally-linked information without consent has raised serious privacy concerns. These concerns resulted in standard specifications development addressing privacy and security issues.

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

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

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

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

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

TAGS:



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

RFID tags can be passive, active or battery-assisted passive. An active tag has an on-board battery and periodically transmits its ID signal. A battery-assisted passive (BAP) has a small battery on board and is activated when in the presence of an RFID reader. A passive tag is cheaper and smaller because it has no battery; instead, the tag uses the radio energy transmitted by the reader. However, to operate a passive tag, it must be illuminated with a power level roughly a thousand times stronger than for signal transmission. That makes a difference in interference and in exposure to radiation.

Tags may either be read-only, having a factory-assigned serial number that is used as a key into a database, or may be read/write, where object-specific data can be written into the tag by the system user. Field programmable tags may be writing-once, read-multiple; "blank" tags may be written with an electronic product code by the user.

RFID tags contain at least three parts: an integrated circuit that stores and processes information and that modulates and demodulates radio-frequency (RF) signals; a means of collecting DC power from the incident reader signal; and an antenna for receiving and transmitting the signal. The tag information is stored in a non-volatile memory. The RFID tag includes either fixed or programmable logic for processing the transmission and sensor data, respectively.

An RFID reader transmits an encoded radio signal to interrogate the tag. The RFID tag receives the message and then responds with its identification and other information. This may be only a unique tag serial number, or may be product-related information such as a stock number, lot or batch number, production date, or other specific information. Since tags have individual serial numbers, the RFID system design can discriminate among

several tags that might be within the range of the RFID reader and read them simultaneously.

READERS:



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

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

An Active Reader Active Tag (ARAT) system uses active tags awoken with an interrogator signal from the active reader. A variation of this system could also use a Battery-Assisted Passive (BAP) tag which acts like a passive tag but has a small battery to power the tag's return reporting signal.

Fixed readers are set up to create a specific interrogation zone which can be tightly controlled. This allows a highly defined reading area for

when tags go in and out of the interrogation zone. Mobile readers may be handheld or mounted on carts or vehicles.

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

FREQUENCY BANDS:

BAND	REGULATIONS	RANGE	DATA SPEED	REMARKS
120– 150 kHz (LF)	Unregulated	10 cm	Low	Animal identification, factory data collection
13.56 MHz (HF)	ISM bandworldwide	10 cm–1 m	Low to moderate	Smart cards (ISO/IEC 15693, ISO/IEC

				14443 A, B). ISO-non-compliant memory cards (MifareClassic, iCLASS, Legic, Felica ...). ISO-compatible microprocessor cards (Desfire EV1, Seos)
433 MHz (UHF)	Short range devices	1–100 m	Moderate	Defense applications, with active tags
865– 868 MHz (Europe) 902– 928 MHz (North America) UHF	ISM band	1–12 m	Moderate to high	EAN, various standards; used by railroads
2450– 5800 MHz (microwave)	ISM band	1–2 m	High	802.11 WLAN, Bluetooth standards
3.1–10 GHz	Ultra wide band	Up to	High	Requires semi-

(microwave)		200 m		active or active tags
-------------	--	-------	--	-----------------------

SIGNALING:

Signaling between the reader and the tag is done in several different incompatible ways, depending on the frequency band used by the tag. Tags operating on LF and HF bands are, in terms of radio wavelength, very close to the reader antenna because they are only a small percentage of a wavelength away. In this near field region, the tag is closely coupled electrically with the transmitter in the reader. The tag can modulate the field produced by the reader by changing the electrical loading the tag represents. By switching between lower and higher relative loads, the tag produces a change that the reader can detect. At UHF and higher frequencies, the tag is more than one radio wavelength away from the reader, requiring a different approach. The tag can backscatter a signal. Active tags may contain functionally separated transmitters and receivers, and the tag need not respond on a frequency related to the reader's interrogation signal.

An Electronic Product Code (EPC) is one common type of data stored in a tag. When written into the tag by an RFID printer, the tag contains a 96-bit string of data. The first eight bits are a header which identifies the version of the protocol. The next 28 bits identify the organization that manages the data for this tag; the organization number is assigned by the EPC Global consortium. The next 24 bits are an object class, identifying the kind of product; the last 36 bits are a unique serial number for a particular tag. These last two fields are set by the organization that issued the tag. Rather like a URL, the total

electronic product code number can be used as a key into a global database to uniquely identify a particular product.

Often more than one tag will respond to a tag reader, for example, many individual products with tags may be shipped in a common box or on a common pallet. Collision detection is important to allow reading of data. Two different types of protocols are used to "singulate" a particular tag, allowing its data to be read in the midst of many similar tags. In a slotted Aloha system, the reader broadcasts an initialization command and a parameter that the tags individually use to pseudo-randomly delay their responses. When using an "adaptive binary tree" protocol, the reader sends an initialization symbol and then transmits one bit of ID data at a time; only tags with matching bits respond, and eventually only one tag matches the complete ID string.

USES:

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

RFID offers advantages over manual systems or use of bar codes. The tag can be read if passed near a reader, even if it is covered by the object or not visible. The tag can be read inside a case, carton, box or other container, and unlike barcodes, RFID tags can be read hundreds at a time. Bar codes can only be read one at a time using current devices.

In 2011, the cost of passive tags started at US\$0.09 each; special tags, meant to be mounted on metal or withstand gamma sterilization, can go up to US\$5. Active tags for tracking containers, medical assets, or monitoring environmental conditions in data centers start at US\$50 and can go up over

US\$100 each. Battery-Assisted Passive (BAP) tags are in the US\$3–10 range and also have sensor capability like temperature and humidity.

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

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

3.1.8 Summary of the Coin Detector and Counter

- The system of the design allows voltage differences in the output signal from the receiving coil to be measured as the primary or sole basis for coin identification. Because of the novel geometry of the coils, such voltage differences are indicative of the conductance of the whole coin; not just the conductivity of the alloy composition used to fabricate the coin.
- In a preferred embodiment, each coil of the design is wound on a substantially rectangular base, such that the width of each coil is slightly less than the diameter of a dime, and the length of each coil is slightly greater than the diameter of a half-dollar.
- The dime and half-dollar are selected because they are the smallest and the largest coin in the assortment to be tested. When other coin sets are to be counted, the preferred coil width is

substantially equal to the diameter of the smallest coin, and the coil length is substantially equal to the diameter of the largest.

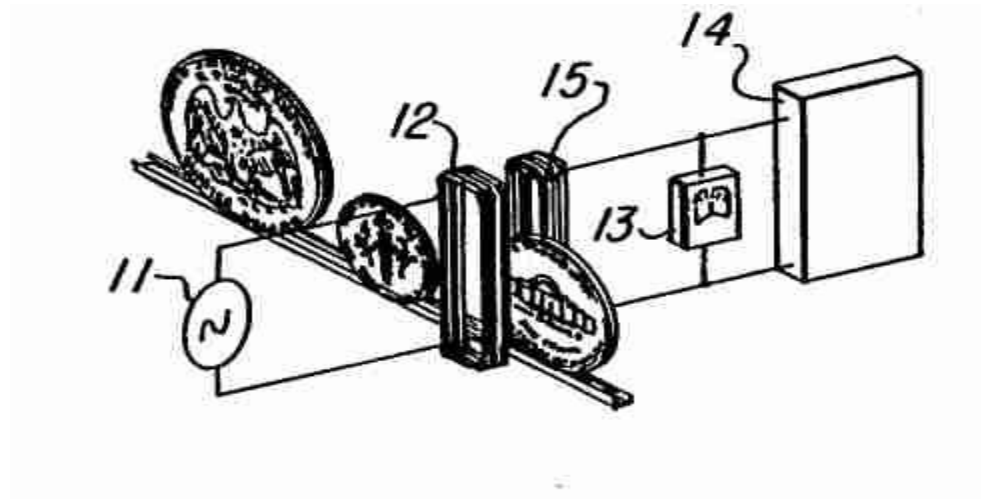
- The transmitter coil is placed on one side of the coin path, and the receiver coil is placed on the opposite side. Each coil is arranged such that the windings lie substantially in a plane parallel to the coin path, and the length of each coil is perpendicular to the coin path.
- Thus, each coin passes between the coils in a direction that traverses the field from one side of the coils to the other side.

When the center of a dime reaches the center of the field it momentarily fills substantially the entire width of the field, and therefore the field "sees" only the dime at that instant, and not any significant portion of the coin ahead of it or behind it in the coin path.

- Each larger coin also interacts with the field across its entire diameter only at the moment its center passes the center of the field, because the length of the coils is substantially equal to or slightly greater than the diameter of the largest coin.
- Circular coils are not suitable because a dime-sized coil pair could never effectively test a larger coin, and a half-dollar-sized coil pair would frequently interact with more than one coin at a time.
- The system of the design reliably identifies and counts each coin as it passes through the field, regardless of the rate of motion of the coins, which pass through the field in a steady stream, or in an intermittent stream, such as in a fare box, for example. No

mechanism is required to control the position or speed of coins as they pass through.

- This greatly simplifies the coin handling apparatus required for feeding coin through the field. The only requirement is that no two coins be allowed to overlap within the space between the coils.

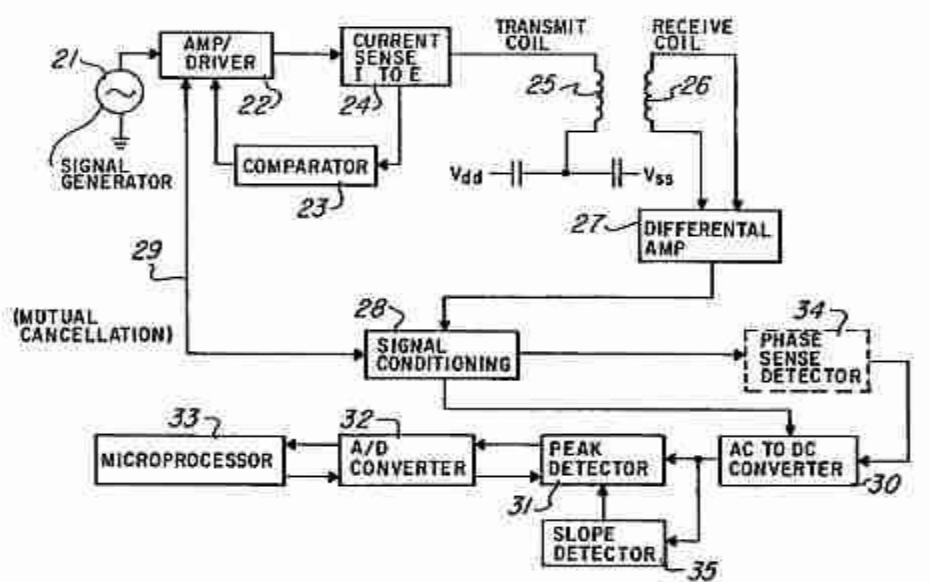


Description of the Coin Detector and Counter

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

- Each coil is wound on an oblong base, which may be either oval-shaped or rectangular. The dimensions of each coil are selected such that the diameter of the smallest coin to be counted, a dime for example, is slightly greater than the width of each coil, and the diameter of the largest coin, a half dollar for example, is slightly less than the length of each coil.
- The coils are coaxial and aligned such that all corresponding sides are parallel to each other.

An assortment of coins to be counted is passed one at a time between the coils. No control of coin speed or position is required except that the full diameter of each coin must pass between the coils.



3.1.9 GLOBAL SYSTEM FOR MOBILE COMMUNICATION (GSM)

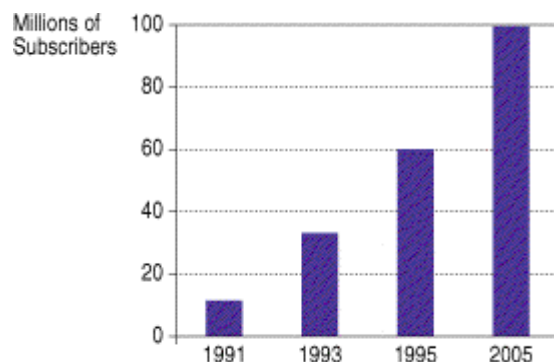
DEFINITION:

Global system for mobile communication (GSM) is a globally accepted standard for digital cellular communication. GSM is the name of a standardization group established in 1982 to create a common European

mobile telephone standard that would formulate specifications for a pan-European mobile cellular radio system operating at 900 MHz. It is estimated that many countries outside of Europe will join the GSM partnership.

INTRODUCTION: THE EVOLUTION OF MOBILE TELEPHONE SYSTEMS:

Cellular is one of the fastest growing and most demanding telecommunications applications. Today, it represents a continuously increasing percentage of all new telephone subscriptions around the world. Currently there are more than 45 million cellular subscribers worldwide, and nearly 50 percent of those subscribers are located in the United States. It is forecasted that cellular systems using a digital technology will become the universal method of telecommunications. By the year 2005, forecasters predict that there will be more than 100 million cellular subscribers worldwide. It has even been estimated that some countries may have more mobile phones than fixed phones by the year 2000.



The concept of cellular service is the use of low-power transmitters where frequencies can be reused within a geographic area. The idea

of cell-based mobile radio service was formulated in the United States at Bell Labs in the early 1970s. However, the Nordic countries were the first to introduce cellular services for commercial use with the introduction of the Nordic Mobile Telephone (NMT) in 1981.

Cellular systems began in the United States with the release of the advanced mobile phone service (AMPS) system in 1983. The AMPS standard was adopted by Asia, Latin America, and Oceanic countries, creating the largest potential market in the world for cellular.

In the early 1980s, most mobile telephone systems were analog rather than digital, like today's newer systems. One challenge facing analog systems was the inability to handle the growing capacity needs in a cost-efficient manner. As a result, digital technology was welcomed. The advantages of digital systems over analog systems include ease of signaling, lower levels of interference, integration of transmission and switching, and increased ability to meet capacity demands. The chart shows worldwide development of mobile telephonesystems.

THE DEVELOPMENT OF MOBILE TELEPHONE SYSTEMS

Year	Mobile System
1981	Nordic Mobile Telephone (NMT) 450
1983	American Mobile Phone System (AMPS)
1985	Total Access Communication System (TACS)
1986	Nordic Mobile Telephony (NMT) 900
1991	American Digital Cellular (ADC)
1991	Global System for Mobile Communication (GSM)
1992	Digital Cellular System (DCS) 1800
1994	Personal Digital Cellular (PDC)
1995	PCS 1900—Canada
1996	PCS—United States

GSM

Throughout the evolution of cellular telecommunications, various systems have been developed without the benefit of standardized specifications. This presented many problems directly related to compatibility, especially with the development of digital radio technology. The GSM standard is intended to address these problems.

From 1982 to 1985 discussions were held to decide between building an analog or digital system. After multiple field tests, a digital system was adopted for GSM. The next task was to decide between a narrow or broadband solution. In May 1987, the narrowband time division multiple access (TDMA) solution was chosen. A summary of GSM milestones is given in Table.

GSM MILESTONES:

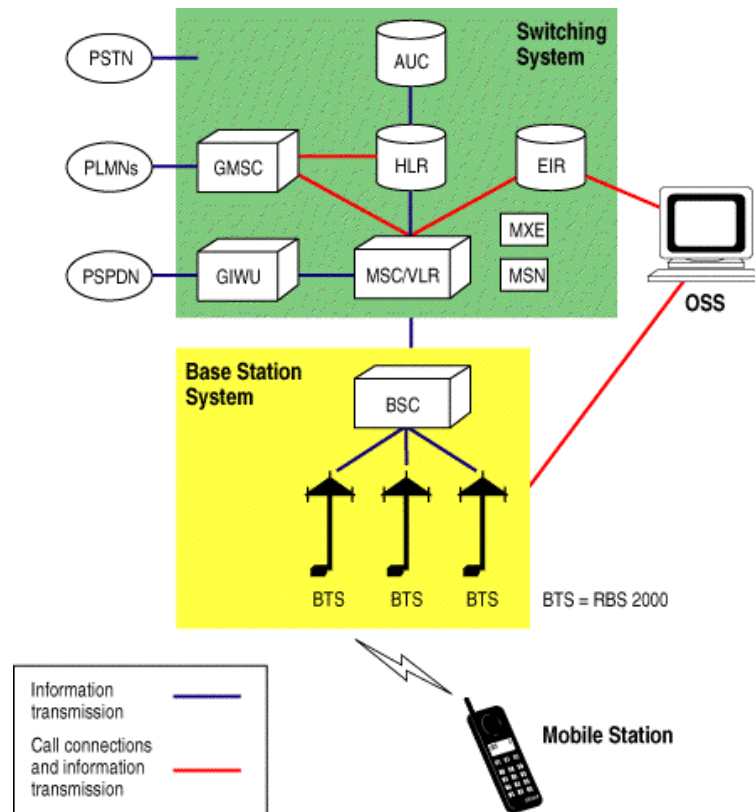
Year	Milestone
1982	GSM formed
1986	field test
1987	TDMA chosen as access method
1988	memorandum of understanding signed

1989	validation of GSM system
1990	preoperation system
1991	commercial system start-up
1992	coverage of larger cities/airports
1993	coverage of main roads
1995	coverage of rural areas

THE GSM NETWORK:

GSM provides recommendations, not requirements. The GSM specifications define the functions and interface requirements in detail but do not address the hardware. The reason for this is to limit the designers as little as possible but still to make it possible for the operators to buy equipment from different suppliers. The GSM network is divided into three major systems: the switching system (SS), the base station system (BSS), and the operation and support system (OSS). The basic GSM network elements are shown in below.

GSM NETWORK ELEMENTS:



THE SWITCHING SYSTEM:

The switching system (SS) is responsible for performing call processing and subscriber-related functions. The switching system includes the following functional units:

- **Home location register (HLR)**—The HLR is a database used for storage and management of subscriptions. The HLR is considered the most important database, as it stores permanent data about subscribers, including a subscriber's service profile, location information, and activity status. When an individual buys a subscription from one of the PCS operators, he or she is registered in the HLR of that operator.
- **Mobile services switching center (MSC)**—The MSC performs the

telephony switching functions of the system. It controls calls to and from other telephone and data systems. It also performs such functions as toll ticketing, network interfacing, common channel signaling, and others.

- **Visitor location register (VLR)**—The VLR is a database that contains temporary information about subscribers that is needed by the MSC in order to service visiting subscribers. The VLR is always integrated with the MSC. When a mobile station roams into a new MSC area, the VLR connected to that MSC will request data about the mobile station from the HLR. Later, if the mobile station makes a call, the VLR will have the information needed for call setup without having to interrogate the HLR each time.
- **Authentication center (AUC)**—A unit called the AUC provides authentication and encryption parameters that verify the user's identity and ensure the confidentiality of each call. The AUC protects network operators from different types of fraud found in today's cellular world.
- **Equipment identity register (EIR)**—The EIR is a database that contains information about the identity of mobile equipment that prevents calls from stolen, unauthorized, or defective mobile stations. The AUC and EIR are implemented as stand-alone nodes or as a combined AUC/EIR node.

THE BASE STATION SYSTEM (BSS):

All radio-related functions are performed in the BSS, which consists of base station controllers (BSCs) and the base transceiver stations (BTSs).

- **BSC**—The BSC provides all the control functions and physical links between the MSC and BTS. It is a high-capacity switch that provides functions such as handover, cell configuration data, and control of radio frequency (RF) power levels in base transceiver stations. A number of BSCs are served by an MSC.
- **BTS**—the BTS handles the radio interface to the mobile station. The BTS is the radio equipment (transceivers and antennas) needed to service each cell in the network. A group of BTSs are controlled by a BSC.

THE OPERATION AND SUPPORT SYSTEM:

The operations and maintenance center (OMC) is connected to all equipment in the switching system and to the BSC. The implementation of OMC is called the operation and support system (OSS). The OSS is the functional entity from which the network operator monitors and controls the system. The purpose of OSS is to offer the customer cost-effective support for centralized, regional, and local operational and maintenance activities that are required for a GSM network. An important function of OSS is to provide a network overview and support the maintenance activities of different operation and maintenance organizations.

ADDITIONAL FUNCTIONAL ELEMENTS:

- **Message center (MXE)**—The MXE is a node that provides integrated voice, fax, and data messaging. Specifically, the MXE handles short message service, cell broadcast, voice mail, fax mail, e-mail, and notification.
- **Mobile service node (MSN)**—The MSN is the node that handles the mobile intelligent network (IN) services.
- **Gateway mobile services switching center (GMSC)**—A gateway is

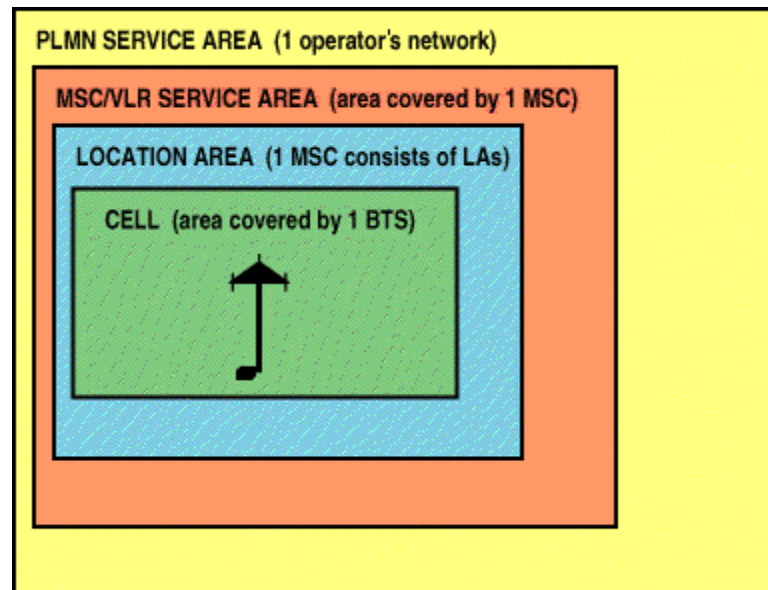
a node used to interconnect two networks. The gateway is often implemented in an MSC. The MSC is then referred to as the GMSC.

- **GSM interworking unit (GIWU)**—The GIWU consists of both hardware and software that provides an interface to various networks for data communications. Through the GIWU, users can alternate between speech and data during the same call. The GIWU hardware equipment is physically located at the MSC/VLR.

GSM NETWORK AREAS:

The GSM network is made up of geographic areas. These areas include cells, location areas (LAs), MSC/VLR service areas, and public land mobile network (PLMN) areas.

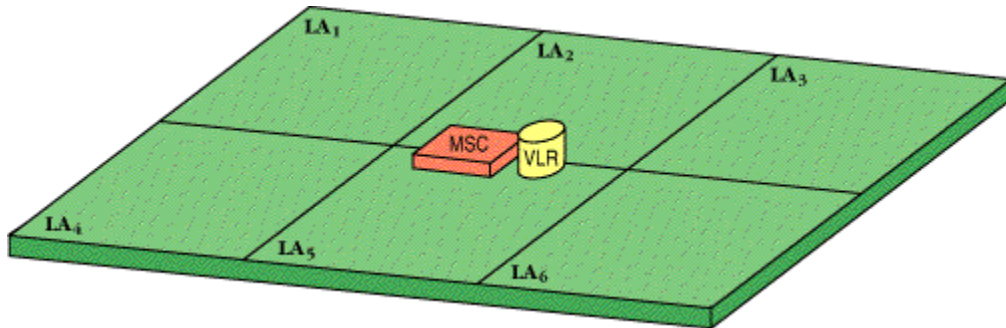
NETWORK AREAS:



The cell is the area given radio coverage by one base transceiver station. The GSM network identifies each cell via the cell global identity (CGI) number assigned to each cell. The location area is a group of cells. It is the area in which the subscriber is paged. Each LA is served by one or more base station

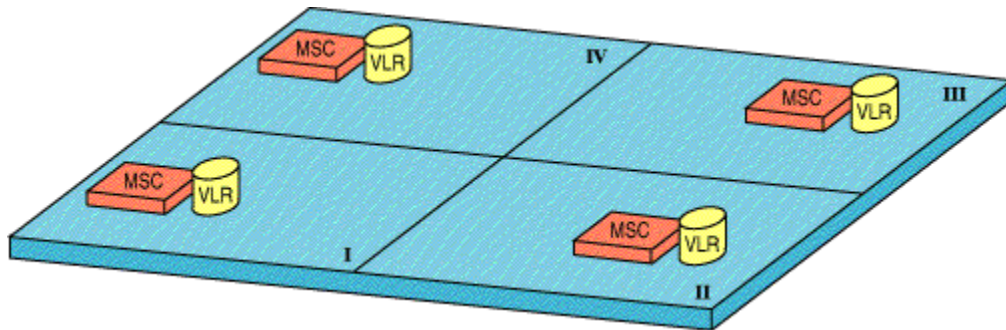
controllers, yet only by a single MSC. Each LA is assigned a location area identity (LAI) number.

LOCATION AREAS:



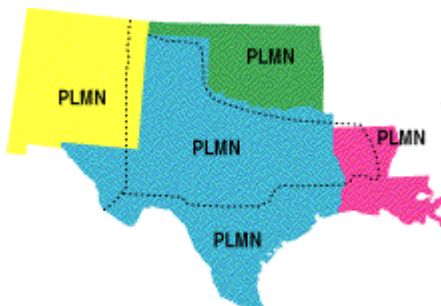
An MSC/VLR service area represents the part of the GSM network that is covered by one MSC and which is reachable, as it is registered in the VLR of the MSC.

MSC/VLR SERVICE AREAS:



The PLMN service area is an area served by one network operator.

PLMN NETWORK AREAS:



GSM SPECIFICATIONS:

Before looking at the GSM specifications, it is important to understand the following basic terms:

- **Bandwidth**—the range of a channel's limits; the broader the bandwidth, the faster data can be sent
- **Bits per second (bps)**—a single on-off pulse of data; eight bits are equivalent to one byte
- **Frequency**—the number of cycles per unit of time; frequency is measured in hertz (Hz)
- **Kilo (k)**—kilo is the designation for 1,000; the abbreviation kbps represents 1,000 bits per second
- **Megahertz (MHz)**—1,000,000 hertz (cycles per second)
- **Milliseconds (ms)**—one-thousandth of a second
- **Watt (W)**—a measure of power of a transmitter.
- Specifications for different personal communication services (PCS) systems vary among the different PCS networks. Listed below is a description of the specifications and characteristics for GSM.
- **Frequency band:** The frequency range specified for GSM is 1,850 to 1,990 MHz (mobile station to base station).
- **Duplex distance:** The duplex distance is 80 MHz. Duplex distance is the distance between the uplink and downlink frequencies. A channel has two frequencies, 80 MHz apart.
- **Channel separation:** The separation between adjacent carrier frequencies. In GSM, this is 200 kHz.
- **Modulation:** Modulation is the process of sending a signal by

changing the characteristics of a carrier frequency. This is done in GSM via Gaussian minimum shift keying (GMSK).

- **Transmission rate:** GSM is a digital system with an over-the-air bit rate of 270 kbps.
- **Access method**—GSM utilizes the time division multiple access (TDMA) concept. TDMA is a technique in which several different calls may share the same carrier. Each call is assigned a particular time slot.
- **Speech coder**—GSM uses linear predictive coding (LPC). The purpose of LPC is to reduce the bit rate. The LPC provides parameters for a filter that mimics the vocal tract. The signal passes through this filter, leaving behind a residual signal. Speech is encoded at 13 kbps.

GSM SUBSCRIBER SERVICES:

There are two basic types of services offered through GSM: telephony (also referred to as teleservices) and data (also referred to as bearer services).

Telephony services are mainly voice services that provide subscribers with the complete capability (including necessary terminal equipment) to communicate with other subscribers. Data services provide the capacity necessary to transmit appropriate data signals between two access points creating an interface to the network. In addition to normal telephony and emergency calling, the following subscriber services are supported by GSM:

- **Dual-tone multifrequency (DTMF)**—DTMF is a tone signaling scheme often used for various control purposes via the telephone network, such as remote control of an answering machine. GSM supports full-originating DTMF.
- **Facsimile group III**—GSM supports CCITT Group 3 facsimile. As

standard fax machines are designed to be connected to a telephone using analog signals, a special fax converter connected to the exchange is used in the GSM system. This enables a GSM-connected fax to communicate with any analog fax in the network.

- **Short message services**—A convenient facility of the GSM network is the short message service. A message consisting of a maximum of 160 alphanumeric characters can be sent to or from a mobile station. This service can be viewed as an advanced form of alphanumeric paging with a number of advantages. If the subscriber's mobile unit is powered off or has left the coverage area, the message is stored and offered back to the subscriber when the mobile is powered on or has reentered the coverage area of the network. This function ensures that the message will be received.
- **Cell broadcast**—A variation of the short message service is the cell broadcast facility. A message of a maximum of 93 characters can be broadcast to all mobile subscribers in a certain geographic area. Typical applications include traffic congestion warnings and reports on accidents.
- **Voice mail**—This service is actually an answering machine within the network, which is controlled by the subscriber. Calls can be forwarded to the subscriber's voice-mail box and the subscriber checks for messages via a personal security code.
- **Fax mail**—With this service, the subscriber can receive fax messages at any fax machine. The messages are stored in a service center from which they can be retrieved by the subscriber via a personal security code to the

desired fax number.

SUPPLEMENTARY SERVICES:

GSM supports a comprehensive set of supplementary services that can complement and support both telephony and data services. Supplementary services are defined by GSM and are characterized as revenue-generating features. A partial listing of supplementary services follows.

- **call forwarding**—This service gives the subscriber the ability to forward incoming calls to another number if the called mobile unit is not reachable, if it is busy, if there is no reply, or if call forwarding is allowed unconditionally.
- **barring of outgoing calls**—This service makes it possible for a mobile subscriber to prevent all outgoing calls.
- **barring of incoming calls**—This function allows the subscriber to prevent incoming calls. The following two conditions for incoming call barring exist: barring of all incoming calls and barring of incoming calls when roaming outside the home PLMN.
- **advice of charge (AoC)**—The AoC service provides the mobile subscriber with an estimate of the call charges. There are two types of AoC information: one that provides the subscriber with an estimate of the bill and one that can be used for immediate charging purposes. AoC for data calls is provided on the basis of time measurements.

- **call hold**—This service enables the subscriber to interrupt an ongoing call and then subsequently reestablish the call. The call hold service is only applicable to normal telephony.
- **call waiting**—This service enables the mobile subscriber to be notified of an incoming call during a conversation. The subscriber can answer, reject, or ignore the incoming call. Call waiting is applicable to all GSM telecommunications services using a circuit-switched connection.
- **multiparty service**—The multiparty service enables a mobile subscriber to establish a multiparty conversation—that is, a simultaneous conversation between three and six subscribers. This service is only applicable to normal telephony.
- **calling line identification presentation/restriction**—These services supply the called party with the integrated services digital network (ISDN) number of the calling party. The restriction service enables the calling party to restrict the presentation. The restriction overrides the presentation.
- **closed user groups (CUGs)**—CUGs are generally comparable to a PBX. They are a group of subscribers who are capable of only calling themselves and certain numbers.

SIM 900 GSM/GPRS MODULE:



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

FEATURES:

- Dual-Band GSM/GPRS 900/ 1800 MHz.
- RS232 interface for direct communication with computer or MCU kit.
- Configurable baud rate.
- Power controlled using 29302WU IC.
- ESD Compliance.
- Enable with MIC and SPEAKER socket.

- With slid in SIM card tray.
- With Stub antenna and SMA connector.
- Input Voltage: 12V DC.

POWER MODES:

Power down mode:

- SIM900A is set power down mode by “AT+CPOWD=0”.
- There are two methods for the module to enter into low current consumption status.

Minimum Functionality Mode:

- Minimum functionality mode reduces the functionality of the module to a minimum and thus minimizes the current consumption to the lowest level.
- If SIM900A has been set to minimum functionality by “AT+CFUN=0”.
- If SIM900A has been set to full functionality by “AT+CFUN=1”.
- If SIM900A is set “AT+CFUN=4” to disable both the above functionality.

Sleep mode:

We can control SIM900A module to enter or exit the SLEEP mode in customer applications through DTR signal. When DTR is in high level and there is no on air and hardware interrupt (such as GPIO interrupt or data on serial port), SIM900A will enter SLEEP mode automatically. In this mode, SIM900A can still receive paging or SMS from network but the serial port is not accessible.

Wake up SIM900A from sleep mode:

- Enable DTR pin to wake up SIM900A. If DTR pin is pulled down to a low level.

- This signal will wake up SIM900A from power saving mode. The serial port will be active after DTR changed to low level for about 50ms.
- Receiving a voice or data call from network to wake up SIM900A.
- Receiving a SMS from network to wake up SIM900A.

PIN SPECIFICATIONS:

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

WORKING:

Unlike mobile phones, a GSM modem doesn't have a keypad and display to interact with. It just accepts certain commands through a serial interface and acknowledges for those. These commands are called as AT commands. There are a list of AT commands to instruct the modem to perform its functions. Every command starts with "AT". That's why they are called as AT commands. AT stands for attention. In our simple project, the program waits for the mobile number to be entered through the keyboard. When a ten digit

mobile number is provided, the program instructs the modem to send the text message using a sequence of AT commands

TESTING YOUR GSM MODEM:

- *The GSM modem can be tested by connecting it with a PC. The modem is equipped with a RS232 cable. Just use a Serial to USB converter and connect it with the PC.*
- *Now you can proceed with sending the commands to the modem using any serial communication program like Hyperterminal, minicom etc. Ensure the serial parameters are configured to 8N1 and the baudrate is set to 9600bps.*
- *For each command you send the modem acknowledges with a message. Example: Just try sending "AT" to the modem. It sends back a result code "OK" which states that the modem is responding. If it's not working fine, it sends "ERROR".*

AT COMMANDS:

- *Set the SIM900 to text mode: AT+CMGF=1\r.*
- *Send SMS to a number: AT+CMGS=PHONE_NUMBER (in international format).*
- *Read the first SMS from the inbox: AT+CMGR=1\r.*
- *Read the second SMS from the inbox: AT+CMGR=2\r.*
- *Read all SMS from the inbox: AT+CMGR=ALL\r.*
- *Call to a number: ATDP+PHONE_NUMBER (in international format).*
- *Hang up a call: ATH.*
- *Receive an incoming call: ATA.*

3.1.10 BUZZER:

A buzzer or beeper is an [audio](#) signaling device, which may be [mechanical](#), [electromechanical](#), or [piezoelectric](#) (piezo for short). Typical

uses of buzzers and beepers include [alarm devices](#), [timers](#), and confirmation of user input such as a mouse click or keystroke.



1. MAGNETIC TRANSDUCER:

Magnetic transducers contain a magnetic circuit consisting of a iron core with a wound coil and a yoke plate, a permanent magnet and a vibrating diaphragm with a movable iron piece. The diaphragm is slightly pulled towards the top of the core by the magnet's magnetic field. When a positive AC signal is applied, the current flowing through the excitation coil produces a fluctuating magnetic field, which causes the diaphragm to vibrate up and down, thus vibrating air. Resonance amplifies vibration through resonator consisting of sound hole(s) and cavity and produces a loud sound.

2. MAGNETIC BUZZER (SOUNDER):

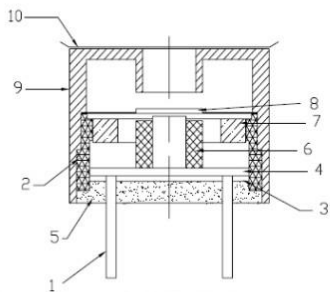
Buzzers like the TMB-series are magnetic audible signal devices with built-in oscillating circuits. The construction combines an oscillation circuit unit with a detection coil, a drive coil and a magnetic transducer. Transistors, resistors, diodes and other small devices act as circuit devices for driving sound generators. With the application of voltage, current flows to the

drive coil on primary side and to the detection coil on the secondary side. The amplification circuit, including the transistor and the feedback circuit, causes vibration. The oscillation current excites the coil and the unit generates an AC magnetic field corresponding to an oscillation frequency. This AC magnetic field magnetizes the yoke comprising the magnetic circuit. The oscillation from the intermittent magnetization prompts the vibration diaphragm to vibrate up and down, generating buzzer sounds through the resonator.

RECOMMENDED DRIVING CIRCUIT FOR MAGNETIC TRANSDUCER INTRODUCTION OF MAGNETIC BUZZER (TRANSDUCER).



Structure

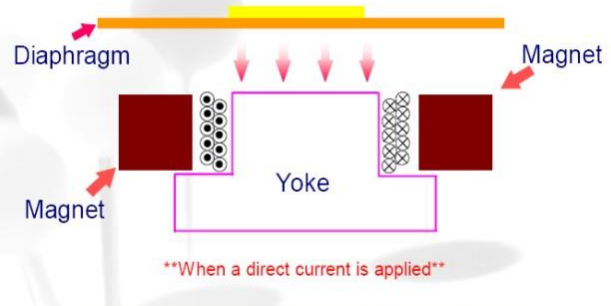


1. PIN
2. Ring
3. PCB
4. Frame
5. EPOXY
6. Coil
7. Magnet
8. Diaphragm
9. Case
10. Stick

We always innovate the best audio technology for you



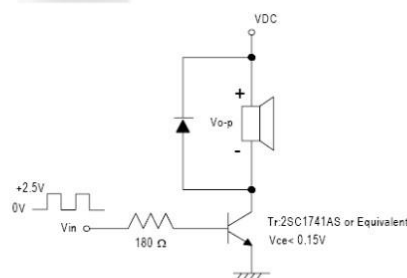
Theory



We always innovate the best audio technology for you



Testing Circuit



We always innovate the best audio technology for you

SPECIFICATIONS:

Rated Voltage: A magnetic buzzer is driven by 1/2 square waves (V o-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum Sound Pressure Level (SPL) under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

Direct Current Resistance: The direct current resistance is measured by ammeter directly.

Sound Output: The sound output is measured by decibel meter. Applying rated voltage and 1/2 square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temperature. : Keep working well between -30°C and +70°C.

How to choose:

Driving methods: AX series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Depend on V o-p (1/2 square waves)

Fixed methods: From the highest cost to the lowest- DIP, wires/ connector, SMD.

Soldering methods: AS series is soldered by hand, the frequency is lower because of the holes on the bottom. On the other hand, we suggest AC series for the reflow soldering, the reliability is better.

How to choose a buzzer :

There are many different kinds of buzzer to choose, first we need to know a few parameters, such as voltage, current, drive method, dimension, mounting type, and the most important thing is how much SPL and frequency we want.

Operating voltage: Normally, the operating voltage for a [magnetic buzzer](#) is from 1.5V to 24V, for a [piezo buzzer](#) is from 3V to 220V. However, in order to get enough SPL, we suggest giving at least 9V to drive a piezo buzzer.

Consumption current: According to the different voltage, the consumption current of a magnetic buzzer is from dozens to hundreds of mill amperes; oppositely, the piezo type saves much more electricity, only needs a few mill amperes, and consumes three times current when the buzzer start to work.

Driving method: Both magnetic and piezo buzzer have self drive type to choose. Because of the internal set drive circuit, the self drive buzzer can emit sound as long as connecting with the direct current. Due to the different work principle, the magnetic buzzer need to be driven by 1/2 square waves, and the piezo buzzer need square waves to get better sound output.

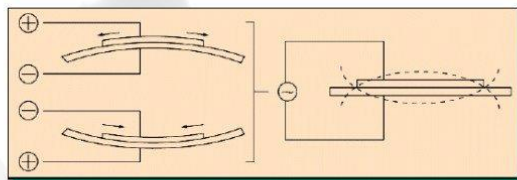
Dimension: The dimension of the buzzer affects its SPL and the frequency, the dimension of the magnetic buzzer is from 7 mm to 25 mm; the piezo buzzer is from 12 mm to 50 mm, or even bigger.

Connecting way: Dip type, Wire type, [SMD](#) type, and screwed type for big piezo buzzer are usually seen.

Sound Pressure Level (SPL): Buzzer is usually tested the SPL at the distance of 10 cm, if distance double, the SPL will decay about 6 dB; oppositely, the SPL will increase 6 dB when the distance is shortened by one time. The SPL of the magnetic buzzer can reach to around 85 dB/ 10 cm; the piezo buzzer can be designed to emit very loud sound, for example, the common siren, are mostly made of piezo buzzer.



Theory

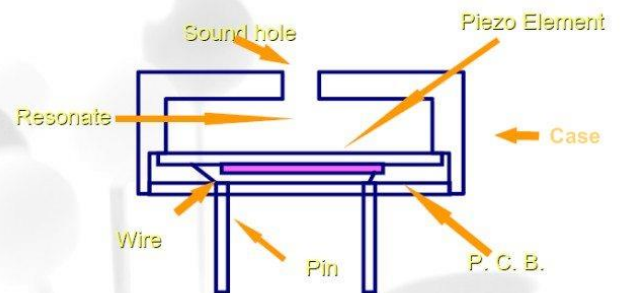


When we give voltage to the ceramic wafer, it contracts and expands. The tension will drive the attached metal sheet emits the sound.

We always innovate the best audio technology for you



Structure



We always innovate the best audio technology for you

INTRODUCTION OF PIEZO BUZZER:

SPECIFICATIONS:

Rated Voltage: A piezo buzzer is driven by square waves (V p-p).

Operating Voltage: For normal operating. But it is not guaranteed to make the minimum SPL under the rated voltage.

Consumption Current: The current is stably consumed under the regular operation. However, it normally takes three times of current at the moment of starting to work.

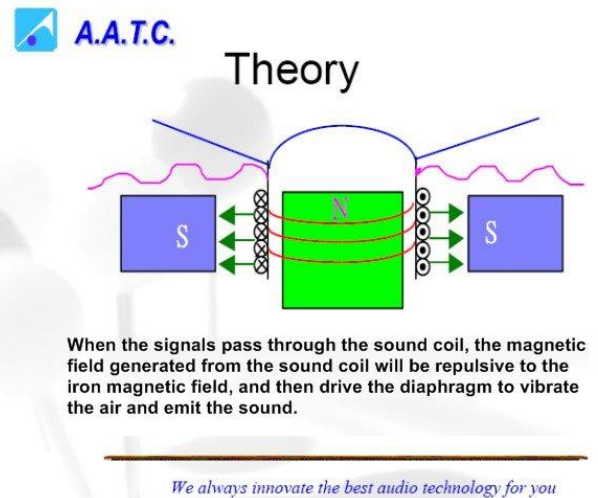
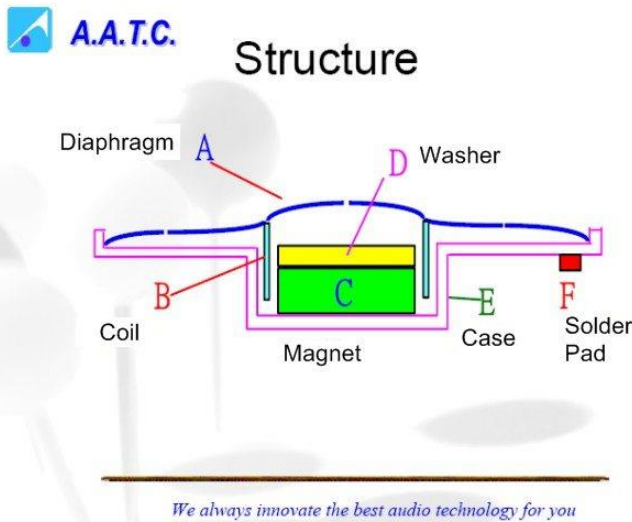
Capacitance: A piezo buzzer can make higher SPL with higher capacitance, but it consumes more electricity.

Sound Output: The sound output is measured by decibel meter. Applying rated voltage and square waves, and the distance of 10 cm.

Rated Frequency: A buzzer can make sound on any frequencies, but we

suggest that the highest and the most stable SPL comes from the rated frequency.

Operating Temp.: Keep working well between -30°C and $+70^{\circ}\text{C}$.



Driving methods: AZ-xxxxS-x series with built drive circuit will be the best choice when we cannot provide frequency signal to a buzzer, it only needs direct current. Besides, there are different tone nature for you to choose, such as continuous, fast pulse, and slow pulse.

Dimension: Dimension affects frequency, small size result in high frequency.

Voltage: Driven by square waves (V p-p), the higher voltage results in the higher SPL.

Pin Pitch: The numerous spec. for the piezo buzzers lead to the difficulty in finding a spec. in facsimile, therefore we suggest that you can firstly choose a spec. with the same pitch and similar frequency. Introduction of Micro Speaker How to choose:

The factors which affect the SPL: the square measure of diaphragm, the amplitude of vibration, magnetic field intensity, power, impedance, resonant chamber, the pattern and the thickness of diaphragm, and the holes.

Power vs. SPL: Suppose all the conditions are the same, increasing the power

dose not mean the SPL will increase as well. We need to revise the diaphragm and the sound coil to load the higher power, but it leads to lower SPL instead.

Dimension vs. SPL: A larger speaker can vibrate more air, therefore it provides higher SPL. In addition, the thicker speaker can give wider amplitude of vibration which also leads to higher SPL.

Acoustics: What we request most is how much SPL a micro speaker can output.

Matching: It will be better to provide the power slight higher than the rated power for the enlarged circuit.

Question for mechanism:

The volume of the resonant chamber: The general problem of the consuming products is that the resonant chambers are not big enough. We can only try to find space to enlarge the volume of the resonant chamber.

Sound Hole: Must be more than $1/8$ of the diaphragm's area at least.

Airtight: The front and back sound fields of the speaker should be separated to avoid neutralization.

Shock absorber: When a speaker works the vibration will also happen at the same time. In order to reduce interference, it will do good to have some material between speaker and case to absorb the shock.

Mounting: The speakers are usually fixed on the case. Firmly fixed is important especially for the iron housing or the large size to avoid separating in the drop test.

How to choose the speaker

Dimension: To the micro speaker, size has decisive influence on its volume. 5mm difference of diameter might result in double or half area of diaphragm, therefore the SPL is quite different. Besides, the thicker speaker has more space

to vibrate the air, and usually has bigger magnet, so it will be more powerful to push the air and emit louder sound.

Power: Mainly refer to how much power can a speaker bear, there is no direct relation to the SPL. The speaker with larger power needs to use thicker diaphragm and sound coil to bear larger power, which will lead to lower efficiency (SPL). Therefore, according to the mechanical design, try to select a larger speaker which matches the outputting power from the amplified circuit, then the best SPL would be emitted.

Impedance: Higher impedance can save more electricity, however, the SPL and the loaded power will go down. The reason is that we have to use thinner wire or to coil more, the front makes the power lower, and the after leads to heaviness and low efficiency.

The material of diaphragm: Most speakers (diameter less than 50mm) use mylar diaphragms, which are easily finished, cheaper and waterproof. However, mylar diaphragm is not good at heatproof and the sound is stiff.

The patterns of diaphragm: The speaker with concentric circles diaphragm is good for the speech sounds. Generally, the SPL is good at the frequency before 5-6 KHz, but will dramatically decrease after 6 KHz. On the other hand, the speaker with radiate diaphragm has average frequency response. Supposing other conditions are all the same, the SPL of radiate diaphragm will lower than the concentric circles one at the frequency before 6 KHz.

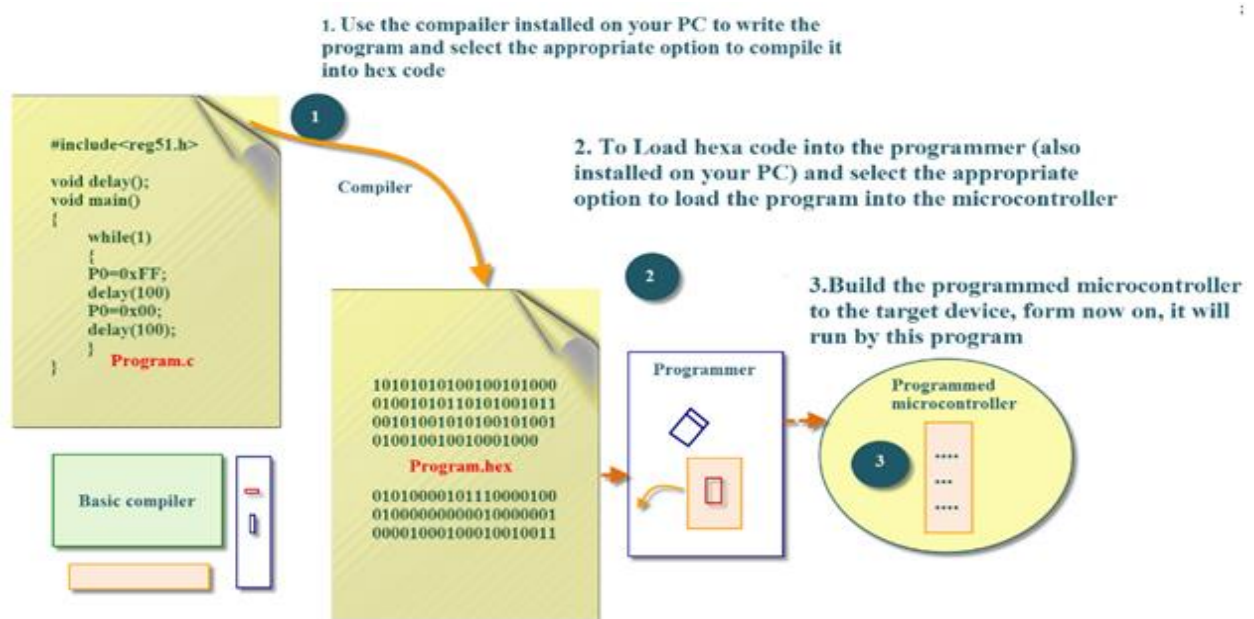
3. SOFTWARE REQUIREMENTS:

3.1.4 EMBEDDED C

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

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

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



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

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

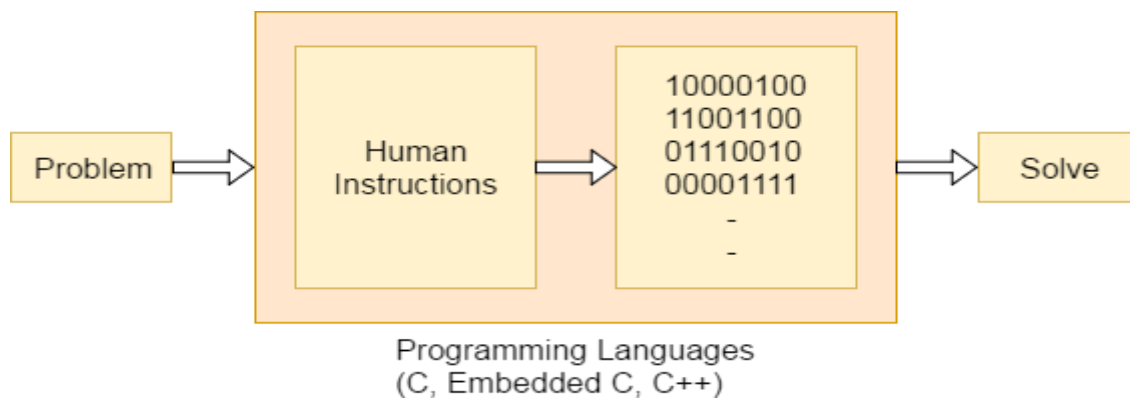
- Easy to understand

- High Reliability
- Portability
- Scalability

EMBEDDED SYSTEM PROGRAMMING:

Basic Declaration

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

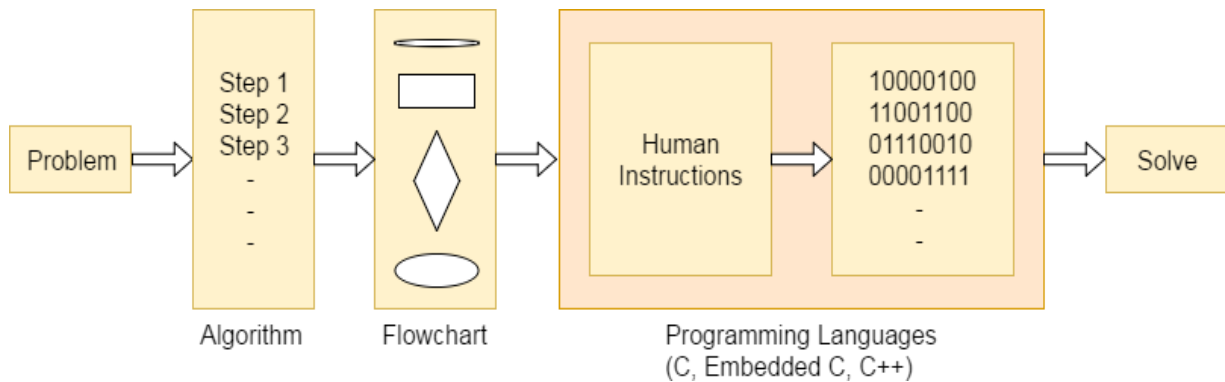


Function is a collection of statements that is used for performing a specific task and a collection of one or more functions is called a programming language. Every language is consisting of basic elements and grammatical rules. The C language programming is designed for function with variables, character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

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

BASIC EMBEDDED C PROGRAMMING STEPS:

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



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

EMBEDDED SYSTEMS:

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

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

SYSTEM:

System is a way of working, organizing or performing one or many tasks according to a fixed set of rules, program or plan.

It is an arrangement in which all the unit combined to perform a work together by following certain set of rules in real time computation. It can also be defined as a way of working, organizing or doing one or many tasks according to a fixed plan.

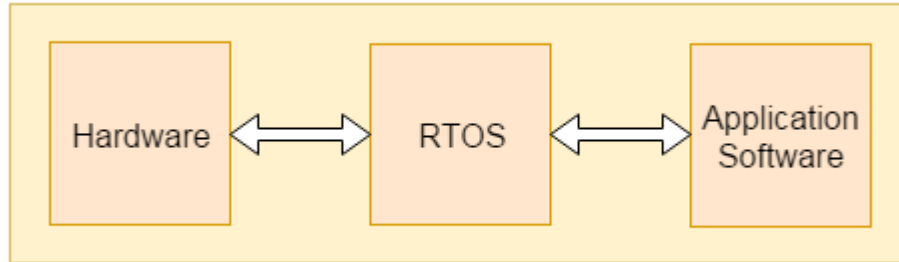
An Embedded System is a system that has software embedded into computer-hardware, which makes a system dedicated for a variety of application or specific part of an application or product or part of a larger system.

An embedded system can be a small independent system or a large combinational system. It is a microcontroller-based control system used to perform a specific task of operation.

An embedded system is a combination of three major components:

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

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



CHARACTERISTICS OF EMBEDDED SYSTEM:

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

Advantages:

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

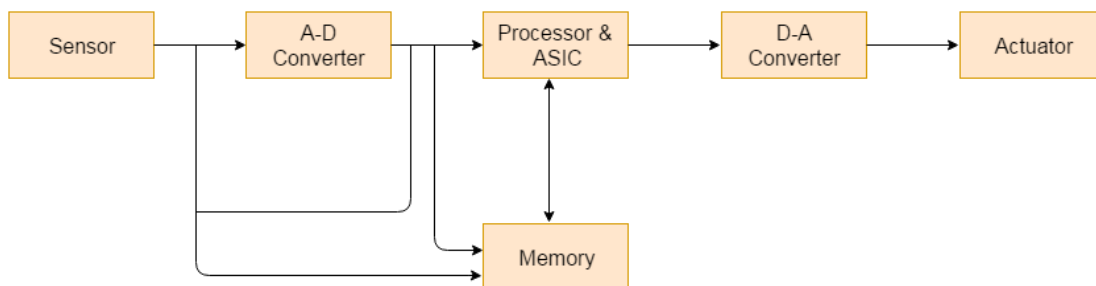
Disadvantages:

- Developing a system required more time. Due to functional complexity.
- Skilled engineers required because one mistake may result in destroying of complete project.

Designing of an embedded system

BASIC STRUCTURE OF AN EMBEDDED SYSTEM:

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



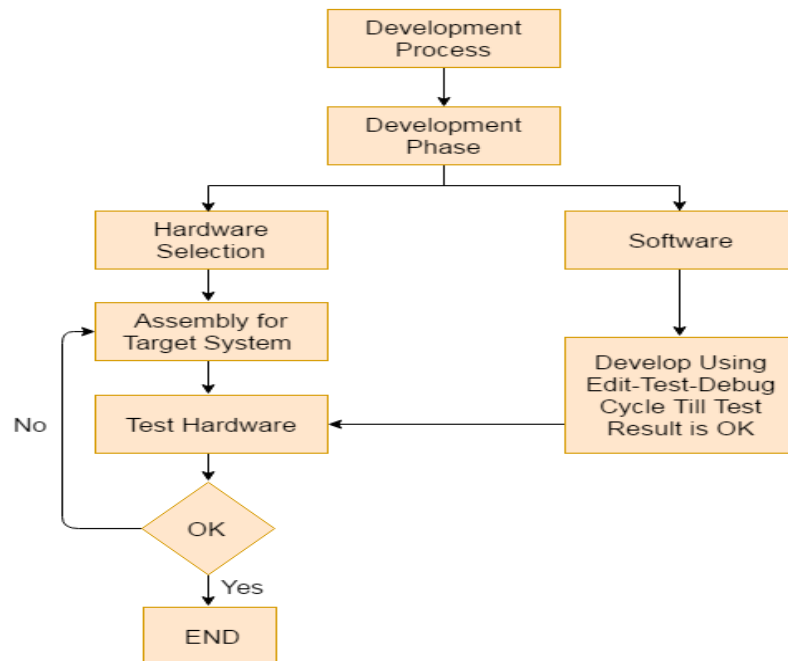
- **Sensor:** Sensor used for sensing the change in environment condition and it generate the electric signal on the basis of change in environment condition. Therefore it is also called as transducers for providing electric input signal on the basis of change in environment condition.
- **A-D Converter:**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

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

DESIGN STEPS REQUIRED FOR THE DEVELOPMENT OF EMBEDDED SYSTEM:

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

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



EMBEDDED SYSTEM TOOLS AND PERIPHERALS:

Compiler:

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

Let's see the operations performed by compiler are:

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

Cross-Compiler:

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

Decompiler:

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

Assembler:

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

DEBUGGING TOOLS IN AN EMBEDDED SYSTEM:

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

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

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

Simulators:

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

Functions of simulators:

Let's see the functions performed by simulator are:

- It defines the processing or processor device family with various version of target system.

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

MICROCONTROLLER STARTER KIT:

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

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

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

Emulators:

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

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

PERIPHERAL DEVICES IN EMBEDDED SYSTEMS:

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

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

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

CRITERIA FOR CHOOSING MICROCONTROLLER:

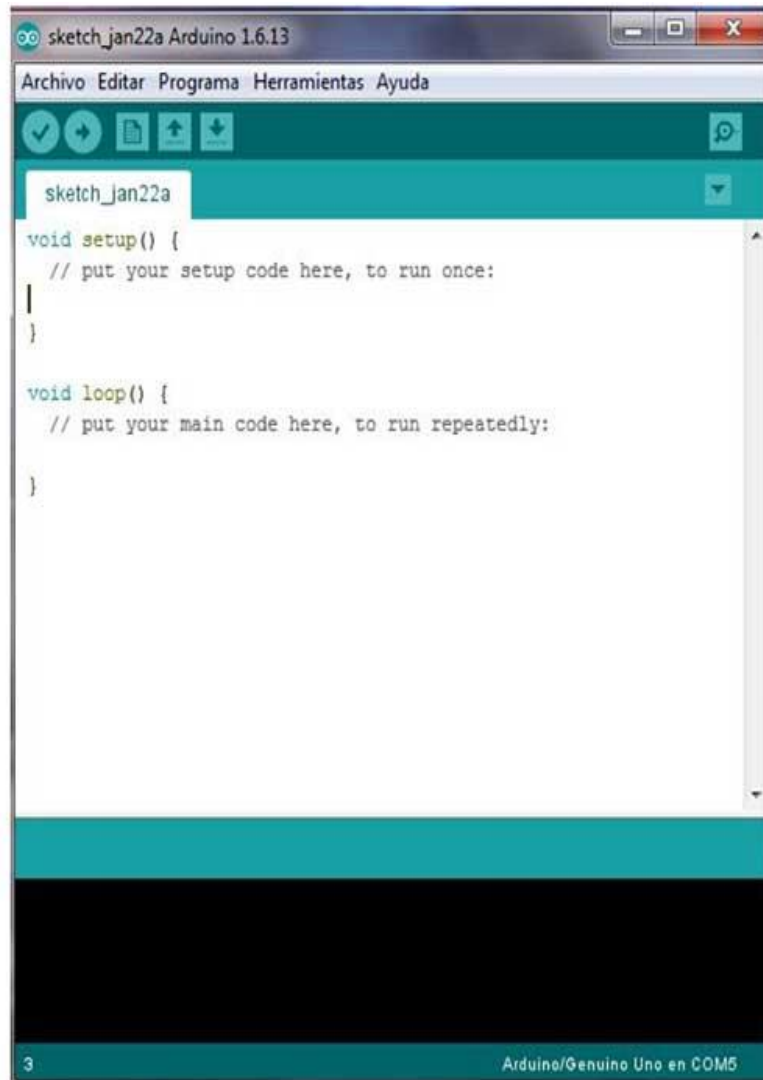
Choosing a microcontroller is essential process in designing of embedded system. While selecting a microcontroller, make sure that it meets the system need and it must be cost effective. We need to decide whether an 8-bit, 16-bit or 32-bit microcontroller is best suitable for the computing needs of a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

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

- **Speed:** The operational speed of the microcontroller or the highest speed microcontroller can support.
- **Packaging:** Packaging is important for improving the assembling, space and prototyping of an end-product.
- **RAM and ROM:** On the basis of operation of embedded system and memory need for storage data and programs the type of microcontroller required for designing system is decided.
- **Count of I/O pins:** The number of input and output devices connected with the system plays an essential role in choosing the type of microcontroller.
- **Cost per unit:** It is important in terms of final cost of the product in which the microcontroller is to be used.
- **Power consumption:** Power consumption plays an important role for maintaining the efficiency of an embedded system.

3.1.5 ARDUINO SOFTWARE IDE

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

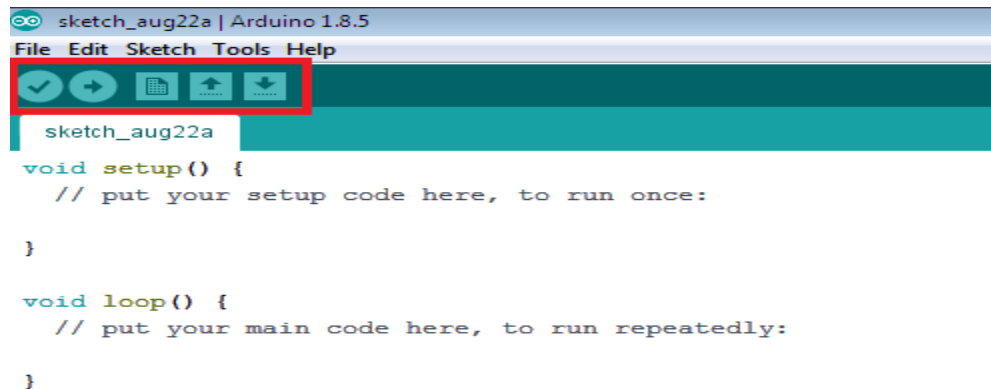


WRITING SKETCHES:

Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file

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

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

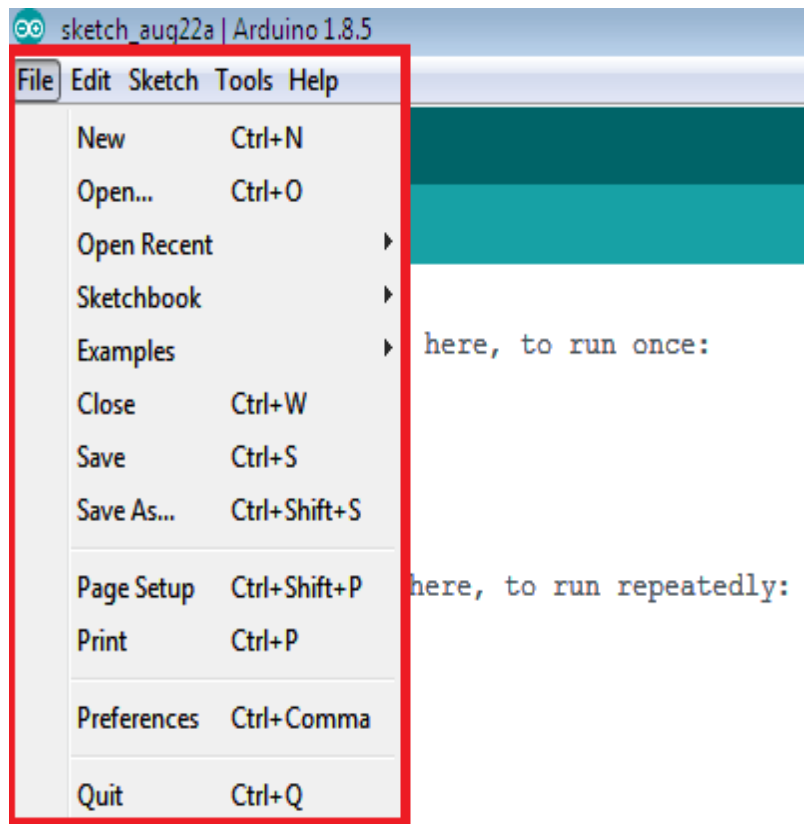


	Verify Checks your code for errors compiling it.
	Upload Compiles your code and uploads it to the configured board. See uploading below for details.

	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"
	<i>New</i> Creates a new sketch.
	<i>Open</i> 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.
	<i>Save</i> Saves your sketch.
	<i>Serial Monitor</i> Opens the <u>serial monitor</u> .

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

FILE:



➤ ***New***

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

➤ ***Open***

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

➤ ***OpenRecent***

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

➤ ***Sketchbook***

Shows the current sketches within the sketchbook folder structure;

clicking on any name opens the corresponding sketch in a new editor instance.

➤ ***Examples***

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

➤ ***Close***

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

➤ ***Save***

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

➤ ***Saveas***

Allows saving the current sketch with a different name.

➤ ***PageSetup***

It shows the Page Setup window for printing.

➤ ***Print***

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

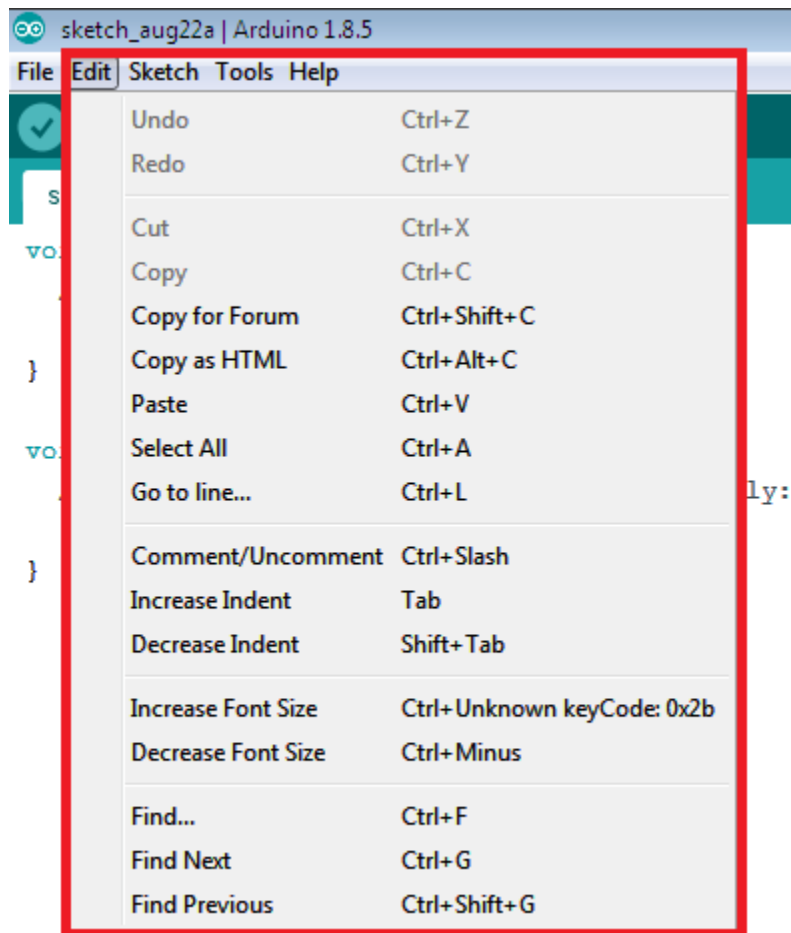
➤ ***Preferences***

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

➤ **Quit**

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

EDIT:



➤ **Undo/Redo**

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

➤ **Cut**

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

➤ ***Copy***

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

➤ ***CopyforForum***

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

➤ ***CopyasHTML***

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

➤ ***Paste***

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

➤ ***SelectAll***

Selects and highlights the whole content of the editor.

➤ ***Comment/Uncomment***

Puts or removes the // comment marker at the beginning of each selected line.

➤ ***Increase/DecreaseIndent***

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

➤ ***Find***

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

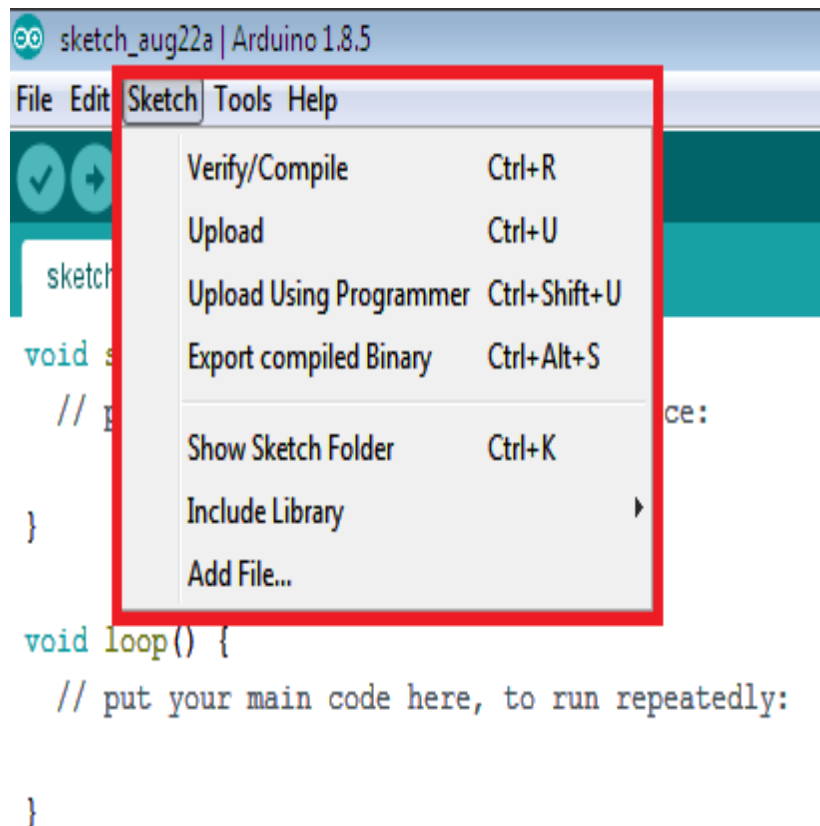
➤ ***FindNext***

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

➤ ***FindPrevious***

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

SKETCH:



➤ ***Verify/Compile***

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

➤ ***Upload***

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

➤ ***Upload Using Programmer***

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

➤ ***Export Compiled Binary***

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

➤ ***Show Sketch Folder***

Opens the current sketch folder.

➤ ***Include Library***

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

➤ ***Add File...***

Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking

on the small triangle icon below the serial monitor one on the right side of the toolbar.

TOOLS:

➤ ***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.

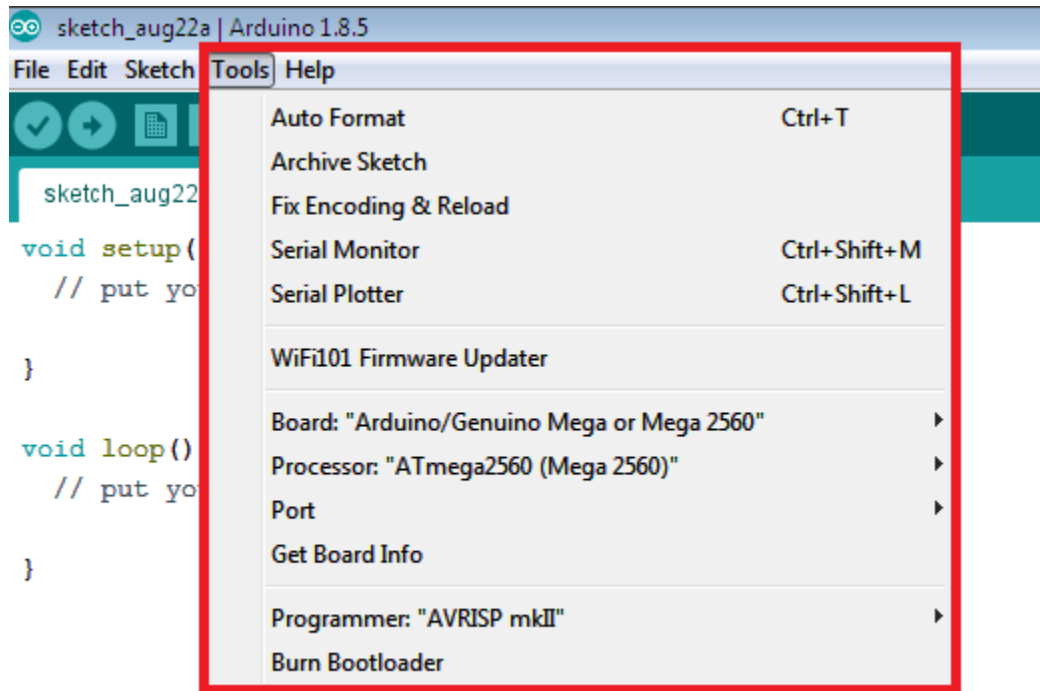
➤ ***Board***

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.

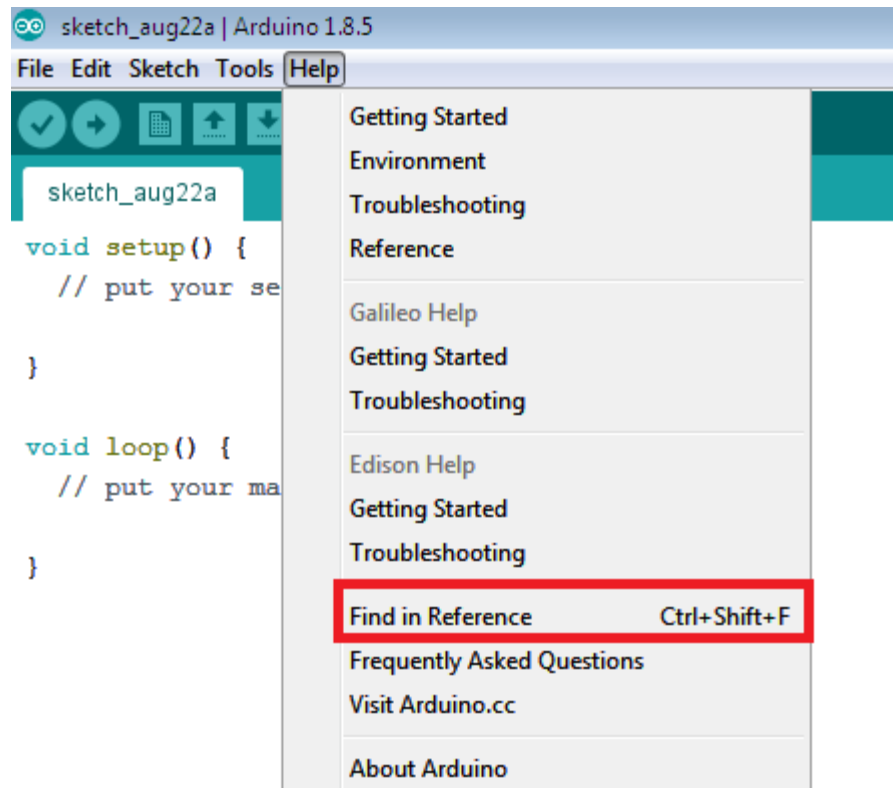
➤ ***Burn Boot loader***

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 or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally comes without a boot loader). Ensure that you've selected the correct board from

the **Boards** menu before burning the boot loader on the target board. This command also set the right fuses.

Help:

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

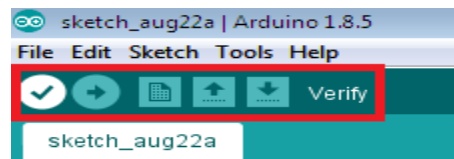


➤ *FindinReference*

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

SKETCHBOOK:

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

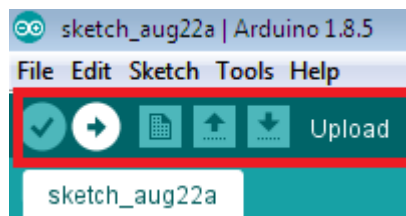


Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

TABS, MULTIPLE FILES, AND COMPILATION:

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

UPLOADING:

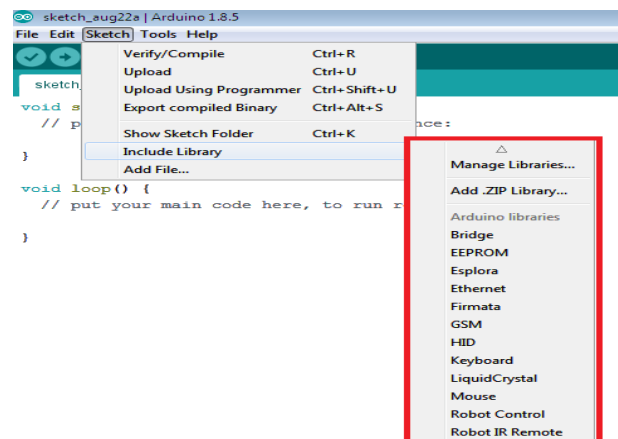


Before uploading your sketch, you need to select the correct items from the **Tools > Board** and **Tools > Port** menus. The boards are described below. On the Mac, the serial port is probably something like **/dev/tty.usbmodem241** (for an Uno or Mega2560 or Leonardo) or **/dev/tty.usbserial-1B1**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

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

LIBRARIES:

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the **Sketch > Import Library** menu. This will insert one or more **#include** statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its **#include** statements from the top of your code.



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

THIRD-PARTY HARDWARE:

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

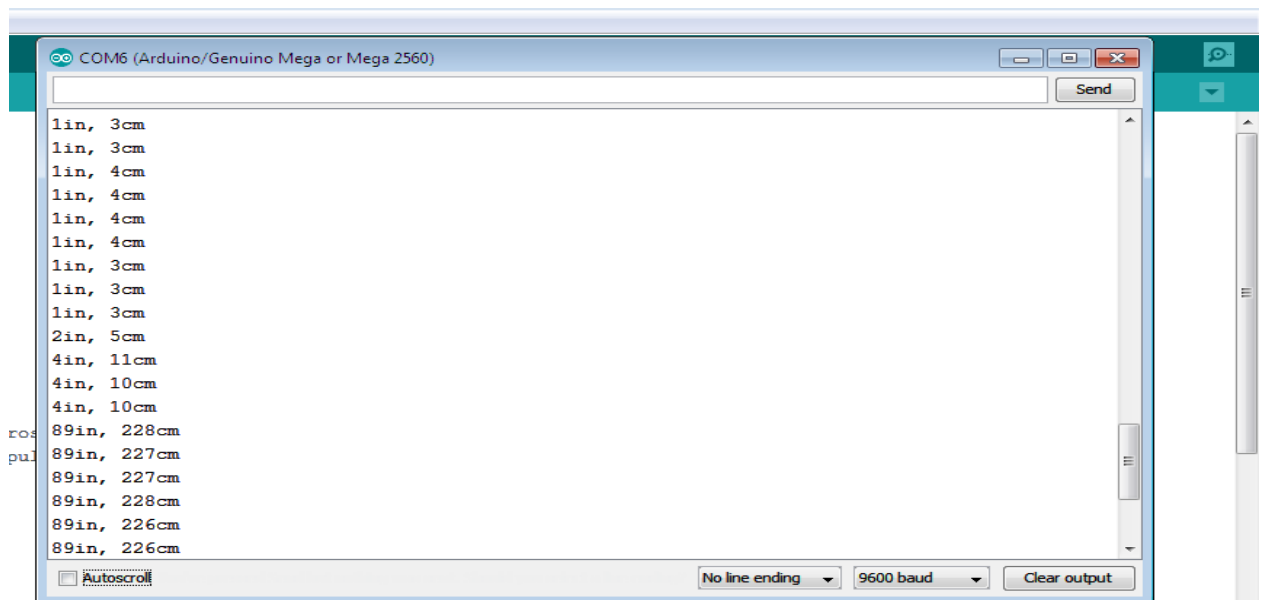
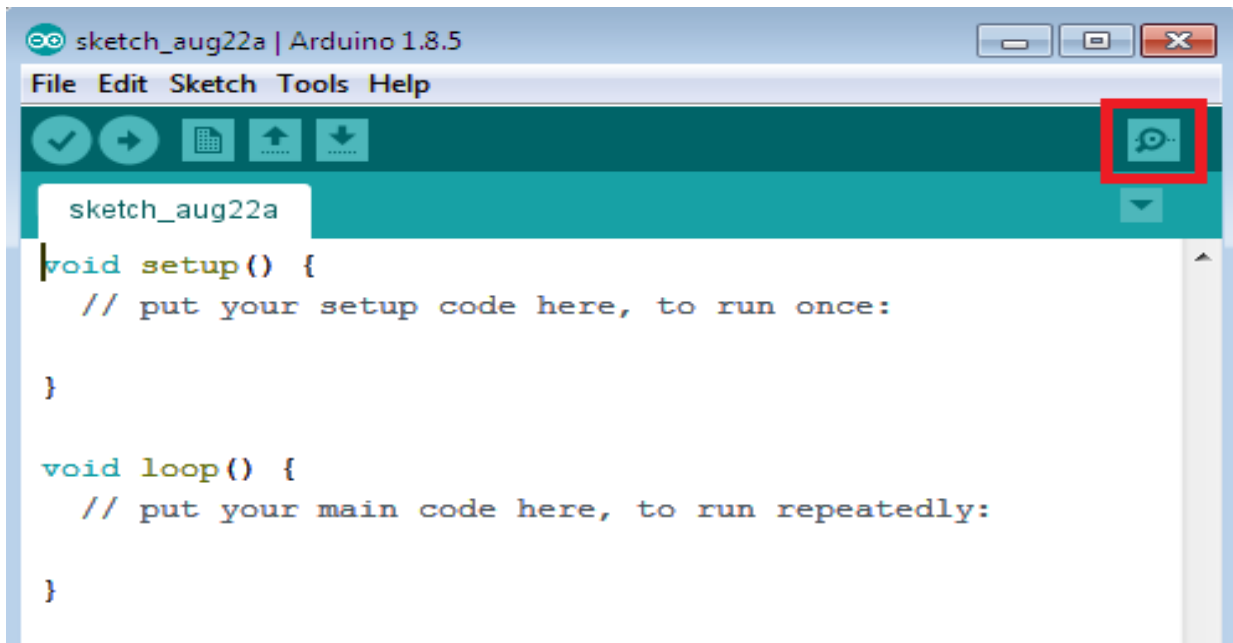
sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

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

SERIAL MONITOR:

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

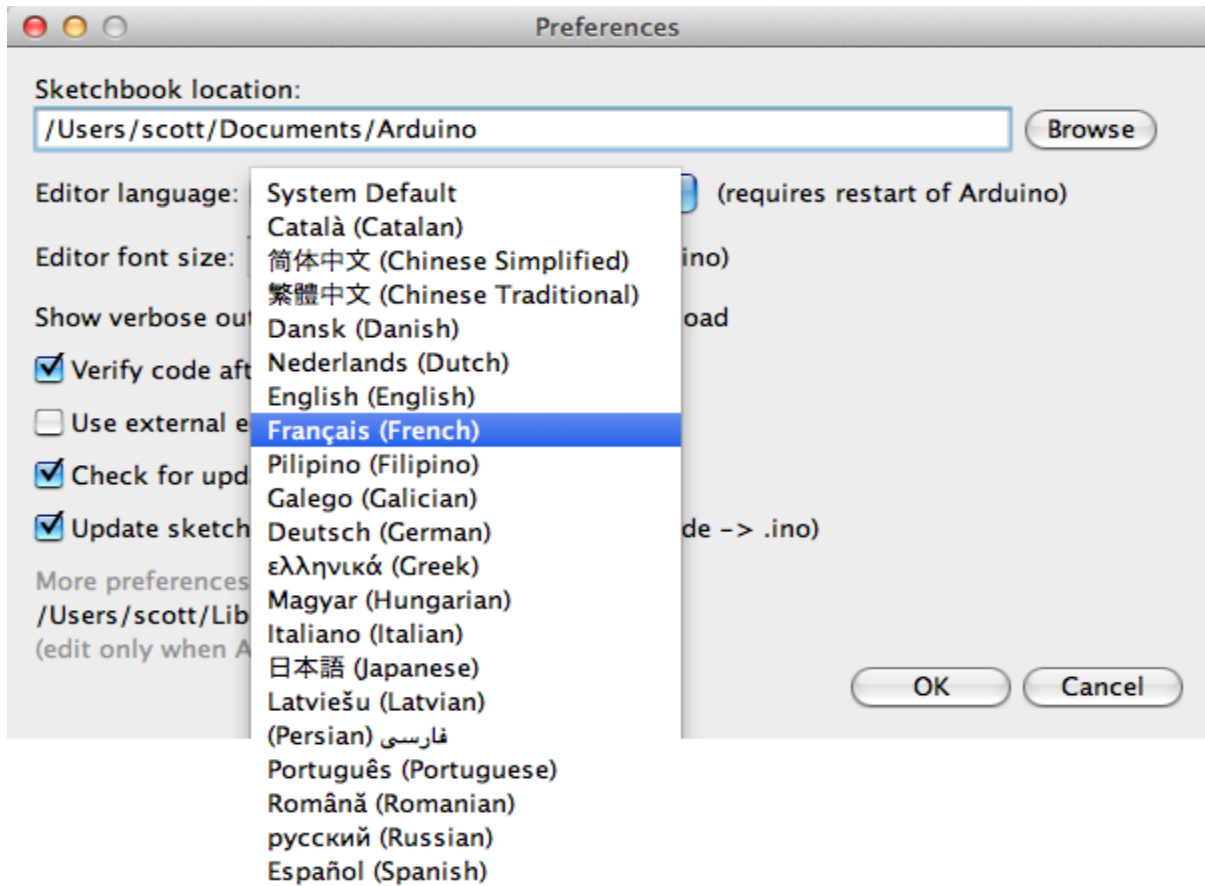
You can also talk to the board from Processing, Flash, MaxMSP, etc (see the [interfacing page](#) for details).



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.

CHAPTER: 4 - OUTPUT

1.1 OUTPUT:

CHAPTER: 5 -FUTURE ENHANCEMENT

5.1APPLICATIONS

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

5.2FUTURE ENHANCEMENT:

- ✓ No Fuel, Cheaper To Maintain.
- ✓ More Eco-Friendly, Lower Carbon Footprint.
- ✓ Less Noise Pollution, Smoother Ride.
- ✓ Range Anxiety, Lack of Charging Infrastructure.
- ✓ Long Charging Times.
- ✓ Lower Battery Life, High Battery Costs.

5.3ADVANTAGES:

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

CONCLUSION:

The Share&Charge Foundation develops tools based on distributed technologies, like blockchain technology, to support solutions around Electric

Vehicle charging. Different to fueling an ICE vehicle, charging an Electric Vehicle increases the amount of transactions between users of vehicles and their provider of energy. Charging will take place at many new locations, be it at home, along the journey or at any destination (e.g. work place). These use-cases create the necessity of a simple and efficient payment method for charging sessions – including the possibility for micro-transactions and instant payments.

REFERENCES OR BIBLIOGRAPHY:

- [1] Y. Yan, Q. Li, W. R. Chen, W. Q. Huang, and J. W. Liu, “Hierarchical management control based on equivalent fitting circle and equivalent energy consumption method for multiple fuel cells hybrid power system,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 2786–2797, Apr. 2020.
- [2] L. Zhang, X. S. Hu, Z. P. Wang, F. C. Sun, J. J. Deng, and D. G. Dorrell, “Multiobjective optimal sizing of hybrid energy storage system for electric vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1027–1035, Feb. 2018.
- [3] Q. Li, B. Su, Y. C. Pu, Y. Han, T. H. Wang, L. Z. Yin, and W. R. Chen, “A state machine control based on equivalent consumption minimization for fuel cell/supercapacitor hybrid tramway,” *IEEE Transactions on Transportation Electrification*, vol. 5, no. 2, pp. 552–564, Jun. 2019.
- [4] Q. Yan, B. Zhang, and M. Kezunovic, “Optimized operational cost reduction for an EV charging station integrated with battery energy storage and PV generation,” *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2096–2106, Mar. 2019.

- [5] Q. Y. Sun, S. Chen, L. Chen, and D. Z. Ma, “Quasi-Z-source networkbased hybrid power supply system for aluminum electrolysis industry,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1141–1151, Jun. 2017.
- [6] F. Nejabatkhah and Y. W. Li, “Overview of power management strategies of hybrid AC/DC microgrid,” *IEEE Transactions on Power Electronics*, vol. 30, no. 12, pp. 7072–7089, Dec. 2015.
- [7] H. Tu, H. Feng, S. Srdic, and S. Lukic, “Extreme fast charging of electric vehicles: a technology overview,” *IEEE Transactions on Transportation Electrification*, vol. 5, no. 4, pp. 861–878, Dec. 2019.
- [8] Q. Li, T. H. Wang, C. H. Dai, W. R. Chen, and L. Ma, “Power management strategy based on adaptive droop control for a fuel cellbattery-supercapacitor hybrid tramway,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5658–5670, Jul. 2018.
- [9] M. M. Mahfouz and M. R. Iravani, “Grid-integration of battery-enabled dc fast charging station for electric vehicles,” *IEEE Transactions on Energy Conversion*, vol. 35, no. 1, pp. 375–385, Mar. 2020.
- [10] T. H. Wang, Q. Li, X. T. Wang, Y. B. Qiu, M. Liu, X. Meng, J. C. Li, and W. R. Chen, “An optimized energy management strategy for fuel cell hybrid power system based on maximum efficiency range identification,” *Journal of Power Sources*, vol. 445, pp. 227333, Jan. 2020.