

**BIOMETRIC ATTENDANCE SYSTEM USING
RFID READER AND IR SENSOR.**

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Engineering degree in
Electronics and Communication Engineering.

By

**CHINTHA SUSHMA (38130044)
CHITTEM VENKATA YAMINI (38130046)**



**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 - 600119



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 **C.Sushma (Reg No: 38130044)** and **CH.Venkata Yamini (Reg. No: 38130046)** who carried out the project **entitled "BIOMETRIC ATTENDANCE SYSTEM USING RFID READER AND IR SENSOR"** under my supervision from December 2022 to May 2022.

INTERNAL GUIDE

Dr.G.d.Anbarasi Jebaselvi M.E., Ph.D

HEAD of the Department

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

Submitted for Viva-voce Examination held on

Internal Examiner

External Examiner

DECLARATION

We **C.SUSHMA (38130044)** and **CH.VENKATA YAMINI (38130046)** hereby declare that the Project Report entitled **"BIOMETRIC ATTENDANCE SYSTEM USING RFID READER AND IR SENSOR"** done by me under the guidance of **Dr.G.d.Anbarasi Jebaselvi M.E., Ph.D (INTERNAL)** is submitted in partial fulfillment of the requirements for the award of Bachelors of Engineering degree in Electronics and communication Engineering.

DATE: 22/03/2022

PLACE: VEMULAWADA

Chintha Sushma

SIGNATURE OF THE CANDIDATE

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. I am grateful to them.

We convey my thanks to **Dr. N.M. NANDHITHA, M.E., Ph.D Dean, School of Electrical and Electronics Engineering** and **Dr. T. Ravi M.E., Ph.D Head of the Department, Dept. of Electronics and Communication Engineering** for providing me necessary support and details at the right time during the progressive reviews.

Wewould like to express my sincere and deep sense of gratitude to my Project Guide **Dr.G.d.Anbarasi Jebaselvi M.E., Ph.D** her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project

TABLE OF CONTENTS

Chapter No.	TITLE	PAGE NO
1	Biometric attendance system	8
1.1	ABSTRACT	8
1.2	INTRODUCTION	8
2	LITERATURE SURVEY	9
2.1	RFID Based Attendance System	9
2.2	Fingerprint attendance system for classroom needs	9
2.4	Enhanced attendance Management System: A Biometrics System of Identification Based on Fingerprint	9
3	Block diagram	12
3.1.1	EMBEDDED SYSTEMS	13
3.1.3	Block diagram of an embedded system:	15
3.1.4	Characteristics of embedded systems	15
4	Heterogeneous Architectures	16
4.1	Harsh environment	17
4.2	System safety and reliability	17
5	NODE MCU	21
5.1	NodeMCU Development Board/kit v0.9	22
5.2	NodeMCU Development Board/kit v1.0	23
5.2.1	NodeMCU with ESPlorer IDE	24
5.2.2	NodeMCU with ARDUINO IDE	24
5.3	IR(INFRARED) Sensor	25
5.4	IR Sensor Circuit Diagram and Working Principle	25
5.5	IR Sensor circuit	26
6	Radio frequency identification (RFID) is a form of wireless communication that uses radio waves to identify and track objects	29
6.1	RFID Applications	30
6.3	RFID Tags	31
6.4	RFID READERS	33

6.5		35
	Creating an RFID Solution	
6.5.1	LF RFID	35
6.5.2	HF RFID	35
7	LCD DISPLAY	39
7.1	FUNCTIONAL DESCRIPTION OF THE CONTROLLER IC	40
7.1.1	REGISTERS	40
7.2	BLOCK DIAGRAM	41
7.3	Buzzer	42
8	ARDUINO	48
8.1	Windows	50
	ZIP	51
	CONCLUSIONS	61
	References	61

CHAPTER - 1

Biometric attendance system

1.1 ABSTRACT:

The proposed research work defined for the acknowledgement on RFID based attendance system for institution faculty, as well as its significance. We mainly emphasize the problem the faculty face due to the lack of any existing system which does not use any alert or notification mechanism like sending an SMS to the faculty's mobile phone to prompt them to mark their attendance. The essential purpose of this system is to create an automated notifications based on the biometric fingerprint authentication for attendance. The system avoids attendance oriented discrepancies and overheads on a central repository.

1.2 INTRODUCTION:

- Real-time online attendance method is helpful for workers who do a lot of activities outside the office or workers with multi-schedule. The attendance system using online biometric fingerprint system will reduce the problems caused by manual system usage such as lags in data management. The purpose of the research is to constructs an online presence system that combines fingerprint modules and GPS.
- The ZFM-20 fingerprint module is used as the system's main input as well as a security tool as an entrance to get access to the entire system. GPS module is applied to determine the user's location and sends it to the smartphone. Arduino module in the system will send a text message to the parties concerned about the user's location data automatically. Each module works well and testing the entire system showed the system work reliable according to the initial scenario.
- The User can access the report using SMS, website, and application on the Android smartphone. The fingerprint sensor can determine the fingerprint stored in the database with an average response time of 1.39 seconds, and GPS can determine latitude and longitude with an average error of 0.007352% and 0.0003% respectively.

CHAPTER - 2

LITERATURE SURVEY:

2.1 : RFID Based Attendance System

Author: Raj Kumar Mistri, Kamlesh Kishore, Priyanka Nidhi, Pushpakumari, Vikrantkumar

Content: The identified fingerprint ID is then stored in memory and is made ready for transfer. The data is then sent to the database on a web server where the Database Management System (DBMS) updates the attendance tables accordingly. This database can be downloaded or viewed by the respective faculty.

2.2: Fingerprint attendance system for classroom needs

Author: Basheer K P Mohamed; C V Raghu

Content: Fingerprint attendance system aims to automate the attendance procedure of an educational institution using biometric technology. This will save time wasted on calling out names and it gives a fool-proof method of attendance marking. A hand-held device is used to mark the attendance without the intervention of teacher. The device can be passed and students can mark attendance during the lecture time.

2.3: Implementation of IOT based Attendance Management System on Raspberry Pi

Author: V Ruhitha; V N Prudhvi Raj; G Geetha

Content: In this paper we are presenting the implementation of attendance monitoring system based on the facial recognition. In this method the image database of the classroom is created first and the database is used to train the system. Then in recognition phase the camera is fixed at the focal point in the classroom and it will capture the images, then from the captured images student's faces are recognized & compared with the database and lastly the attendance is marked based on the faces identified.

2.4: Enhanced attendance Management System: A Biometrics System of Identification Based on Fingerprint

Author: Happy N. Monday; Ifeanyi D. Dike; Jian P. Li; David Agomuo; Grace U. Nneji; Abel Ogungbile

Content: This paper presents an enhanced attendance management system using fingerprint system in a university environment. It was developed using the waterfall methodology. This system consists of two procedures; enrolment and identification. During the enrollment, the fingerprint of a person is captured and its unique features extracted and stored in the database along with the user's data as template for the subject. During identification, the fingerprint of the person is again captured and the extracted feature is compared with the template in the database in a ratio 1: N-templates, to identify a match (a user) before attendance is made.

2.5: IoT Based Smart Attendance System (SAS) Using RFID

Author: Soumil Nitin Shah, Abdelshakour Abuzneid

Content: This work introduces a new paradigm of monitoring student attendance using Radio Frequency Identification (RFID) based on the Internet of Thing (IoT). Educational institutes are concerned about student irregular attendance. Truancy can affect a student's overall academic performance. The traditional method of taking attendance by calling names or signing on paper is very time consuming and inefficient. RFID based attendance system using IoT system is one of the solutions to handle the problem. The proposed work comprises of two most popular trend in technology research; IoT and RFID.

2.6: Smart Attendance System Using RFID in IOT Author:

Mahesh Sutar, Mahesh Patil , Sachin Waghmare

Content: In recent years, there have been rise in the number of applications based on Radio Frequency Identification (RFID) systems and have been successfully functional to different areas as diverse as transportation, health-care, agriculture, and hospitality industry to name a few. IOT technology facilitates automatic wireless identification using electronic passive and active tags with suitable readers. In This paper, an effort is made to solve regular lecture attendance monitoring problem in developing countries using RFID technology. The application of RFID to student attendance monitoring as developed and deployed in this study is capable of eliminating time wasted during manual collection of attendance and an opportunity for the educational administrators to capture face-to-face classroom data for allocation of proper attendance scores and for further managerial decisions

2.7 Existing system:

In existing system there is no proper maintenance of attendance system, in this system students won't stay in class room also they have an attendance.

2.8 Proposed system:

- The security system is basically an embedded one. Embedded stands for hardware controller by software. Here a software using microcontroller controls all the hardware components.

- The microcontroller plays an important role in the system. The main objective of the system is to uniquely identify and to make security for a person.
- This requires a unique product, which has the capability of distinguishing different person. This is possible by the new technology called RFID. The main parts of the RFID system are RFID tag and RFID reader. In this system the RFID reader and RFID tag used are operating at 125 KHz.
- The microcontroller internal memory is used for storing the details. This report provides a clear picture of hardware and software used in the system. It also provides an overall view with detailed discussion of the operation of the system
- In the proposed system, the student automation system project has been developed as an important application in order to maintain the attendance automatically using RFID.
- Our proposed system consists mainly 20X4 LCD, RFID reader, RFID tag, Arduino uno buzzer & momentum switches.
- Initially LCD Display unit showing five options to start the process.

These are

1: Enrollment,

2: Attendance,

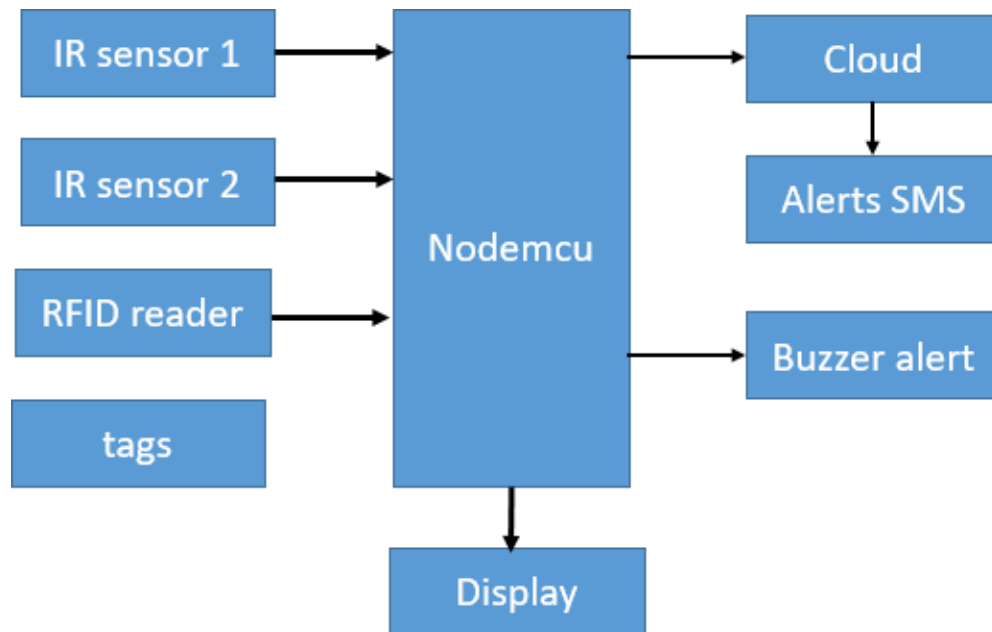
3: Deleting All Records &

4: Results.

- So in our proposed model firstly we have to enroll the students or we can link the database to its ID card details of database and then we can start the making attendance. In enrollment firstly student/employee information are save to the database. After then student can make the attendance. We have used EPROM memory of device to save the attendance of the candidate.

CHAPTER - 3

Block diagram:



3.1 Hardware required:

Nodemcu

IR sensor

RFID reader and tags

Display

Buzzer

Hardware description:

3.1.1. EMBEDDED SYSTEMS

3.1.2 Overview of embedded systems

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. For example, Handheld computers share some elements with embedded systems – such as the operating systems and microprocessors which power them – but are not truly embedded systems, because they allow different applications to be loaded and peripherals to be connected.

Embedded systems provide several functions

- Monitor the environment; embedded systems read data from input sensors. This data is then processed and the results displayed in some format to a user or users
- Control the environment; embedded systems generate and transmit commands for actuators.
- Transform the information; embedded systems transform the data collected in some meaningful way, such as data compression/decompression

Although interaction with the external world via sensors and actuators is an important aspect of embedded systems, these systems also provide functionality specific to their applications. Embedded systems typically execute applications such as control laws, finite state machines, and signal processing algorithms. These systems must also detect and react to faults in both the internal computing environment as well as the surrounding electromechanical systems.

There are many categories of embedded systems, from communication devices to home appliances to control systems. Examples include;

- Communication devices

e. g.: modems, cellular phones

- Home Appliances

e. g.: CD player, VCR, microwave oven

- Control Systems

e. g.: Automobile anti-lock braking systems, robotics, satellite control

f. **3.1.3 Block diagram of an embedded system:**

An embedded system usually contains an embedded processor. Many appliances that have a digital interface -- microwaves, VCRs, cars -- utilize embedded systems. Some embedded systems include an operating system. Others are very specialized resulting in the entire logic being implemented as a single program. These systems are embedded into some device for some specific purpose other than to provide general purpose computing . A typical embedded system is shown in Fig 1.1

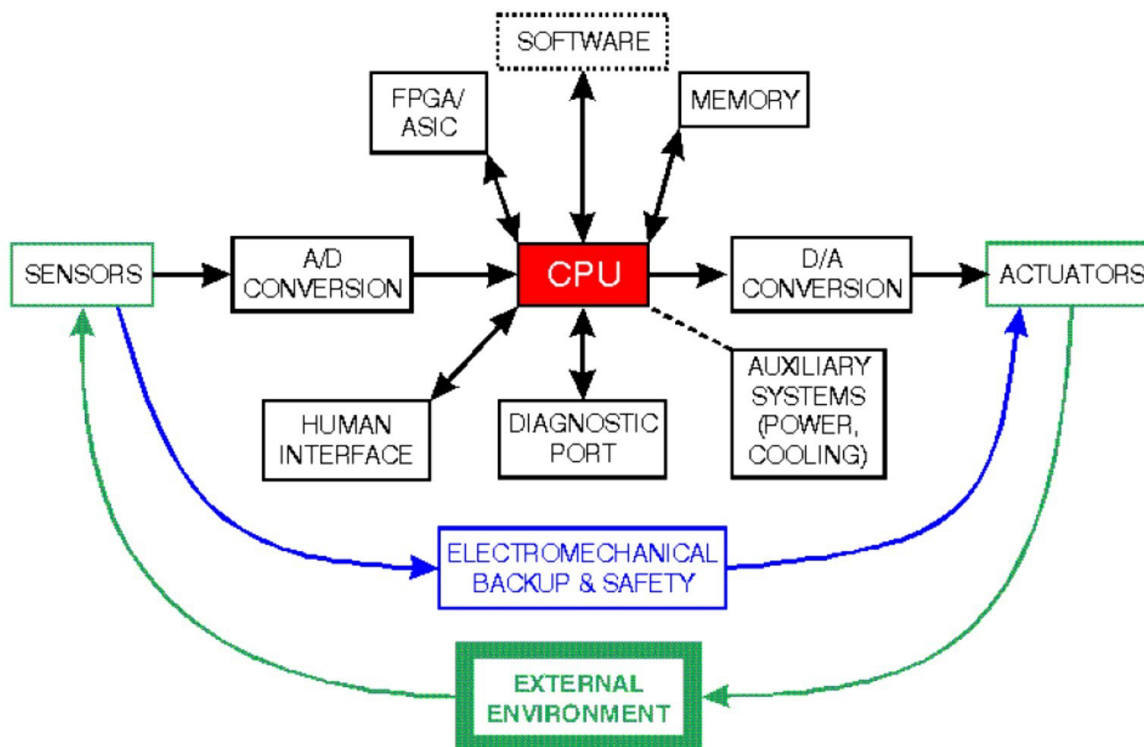


Fig 1.1 Block diagram of a typical embedded system

3.1.4 Characteristics of embedded systems

Embedded systems are characterized by a unique set of characteristics. Each of these characteristics imposed a specific set of design constraints on embedded systems designers. The challenge to designing embedded systems is to conform to the specific set of constraints for the application.

Application Specific Systems:

Embedded systems are not general-purpose computers. Embedded system designs are optimized for a specific application. Many of the job characteristics are known before the hardware is designed. This allows the designer to focus on the specific design constraints of a well-defined application. As such, there is limited user reprogram ability. Some embedded systems, however, require the flexibility of reprogram ability. Programmable DSPs are common for such applications.

Reactive Systems

As mentioned earlier, a typical embedded systems model responds to the environment via sensors and control the environment using actuators. This requires embedded systems to run at the speed of the environment. This characteristic of embedded system is called “reactive”. Reactive computation means that the system (primarily the software component) executes in response to external events. External events can be either periodic or aperiodic. Periodic events make it easier to schedule processing to guarantee performance. Aperiodic events are harder to schedule. The maximum event arrival rate must be estimated in order to accommodate worst case situations. Most embedded systems have a significant reactive component. One of the biggest challenges for embedded system designers is performing an accurate worst case design analysis on systems with statistical performance characteristics (e.g., cache memory on a DSP or other embedded processor). Real time system operation means that the correctness of a computation depends, in part, on the time at which it is delivered. Systems with this requirement must often design to worst case performance. But accurately predicting the worst case may be difficult on complicated architectures. This often leads to overly pessimistic estimates erring on the side of caution. Many embedded systems have a significant requirement for real time operation in order to meet external I/O and control stability requirements. Many real-time systems are also reactive systems.

Distributed Systems

A common characteristic of an embedded system is one that consists of communicating processes executing on several CPUs or ASICs which are connected by communication links. The reason for this is economy. Economical 4 8-bit microcontrollers may be cheaper than a 32-bit processors. Even after adding the cost of the communication links, this approach may be

preferable. In this approach, multiple processors are usually required to handle multiple time-critical tasks. Devices under control of embedded systems may also be physically distributed.

CHAPTER - 4

Heterogeneous Architectures

Embedded systems often are composed of heterogeneous architectures (Fig 1.2). They may contain different processors in the same system solution. They may also be mixed signal systems. The combination of I/O interfaces, local and remote memories, and sensors and actuators makes embedded system design truly unique. Embedded systems also have tight design constraints, and heterogeneity provides better design flexibility.

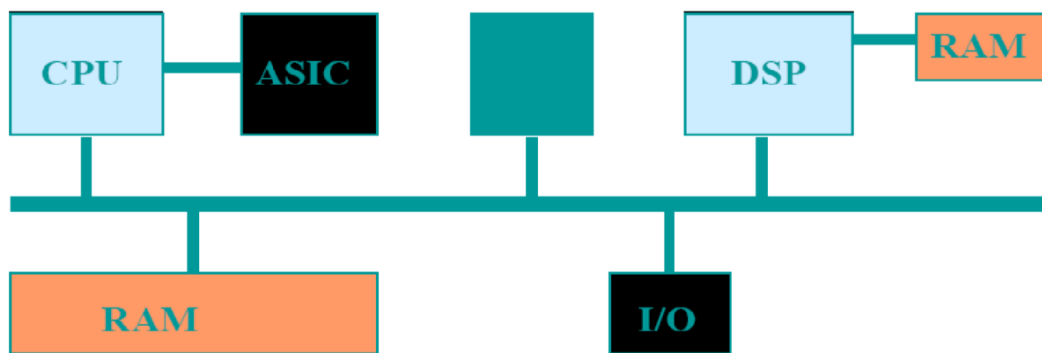


Fig 1.2 Embedded Systems having Heterogeneous Architectures

4.1 Harsh environment

Many embedded systems do not operate in a controlled environment. Excessive heat is often a problem, especially in applications involving combustion (e.g., many transportation applications). Additional problems can be caused for embedded computing by a need for protection from vibration, shock, lightning, power supply fluctuations, water, corrosion, fire, and general physical abuse.

4.2 System safety and reliability

As embedded system complexity and computing power continue to grow, they are starting to control more and more of the safety aspects of the overall system. These safety measures may be in the form of software as well as hardware control. Mechanical safety backups are normally activated when the computer system loses control in order to safely shut down system operation. Software safety and reliability is a bigger issue. Software doesn't normally "break" in the sense of hardware. However software may be so complex that a set of unexpected circumstances can cause software failures leading to unsafe situations. Discussion of this topic is outside the scope of this book, but the challenges for embedded designers include designing reliable software and building cheap, available systems using unreliable components. The main challenge for embedded system designers is to obtain low-cost reliability with minimal

redundancy.

4.3 Control of physical systems

One of the main reasons for embedding a computer is to interact with the environment. This is often done by monitoring and controlling external machinery. Embedded computers transform the analog signals from sensors into digital form for processing. Outputs must be transformed back to analog signal levels. When controlling physical equipment, large current loads may need to be switched in order to operate motors and other actuators. To meet these needs, embedded systems may need large computer circuit boards with many non-digital components. Embedded system designers must carefully balance system tradeoffs among analog components, power, mechanical, network, and digital hardware with corresponding software.

4.4 Small and low weight

Many embedded computers are physically located within some larger system. The form factor for the embedded system may be dictated by aesthetics. For example, the form factor for a missile may have to fit inside the nose of the missile. One of the challenges for embedded systems designers is to develop non-rectangular geometries for certain solutions. Weight can also be a critical constraint. Embedded automobile control systems, for example, must be light weight for fuel economy. Portable CD players must be light weight for portability purposes.

4.5 Cost sensitivity

Cost is an issue in most systems, but the sensitivity to cost changes can vary dramatically in embedded systems. This is mainly due to the effect of computer costs have on profitability and is more a function of the proportion of cost changes compared to the total system cost.

4.6 Power management

Embedded systems have strict constraints on power. Given the portability requirements of many embedded systems, the need to conserve power is important to maintain battery life as long as possible. Minimization of heat production is another obvious concern for embedded systems.

4.7 POWER SUPPLY

All electronic circuits works only in low DC voltage, so we need a power supply unit to provide the appropriate voltage supply for their proper functioning .This unit consists of transformer, rectifier, filter & regulator. AC voltage of typically 230volts rms is connected to a transformer voltage down to the level to the desired ac voltage. A diode rectifier that provides the full wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation . A regulator circuit can use this dc input to provide dc voltage that not only has much less ripple voltage but also remains the same dc value even the dc voltage varies some what, or the load connected to the output dc voltages changes.

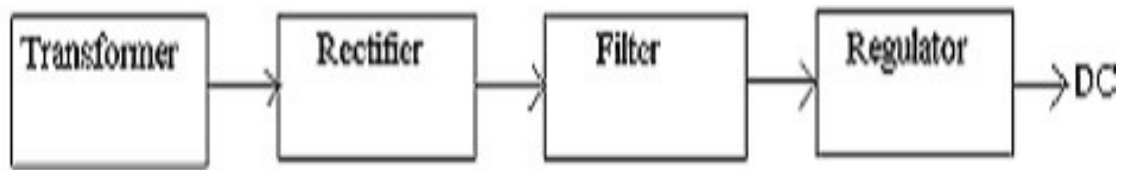


Fig 1.General Block Of Power Supply Unit

4.8 TRANSFORMER:

A transformer is a static piece of which electric power in one circuit is transformed into electric power of same frequency in another circuit. It can raise or lower the voltage in the circuit, but with a corresponding decrease or increase in current. It works with the principle of mutual induction. In our project we are using a step down transformer to providing a necessary supply for the electronic circuits. Here we step down a 230volts ac into 12volts ac.

4.9 RECTIFIER:

A dc level obtained from a sinusoidal input can be improved 100% using a process called full wave rectification. Here in our project for full wave rectification we use bridge rectifier. From the basic bridge configuration we see that two diodes(say D2 & D3) are conducting while the other two diodes (D1 & D4) are in off state during the period $t = 0$ to $T/2$. Accordingly for the negative cycle of the input the conducting diodes are D1 & D4 .Thus the polarity across the load is the same.

In the bridge rectifier the diodes may be of variable types like 1N4001, 1N4003, 1N4004, 1N4005, 1N4007 etc.... can be used. But here we use 1N4007, because it can withstand up to 1000v.

4.10 FILTERS:

In order to obtain a dc voltage of 0 Hz, we have to use a low pass filter. So that a capacitive filter circuit is used where a capacitor is connected at the rectifier output & a dc is obtained across it. The filtered waveform is essentially a dc voltage with negligible ripples & it is ultimately fed to the load.

4.11 REGISTERS:

The controller IC has two 8 bit registers, an instruction register (IR) and a data register (DR). The IR stores the instruction codes and address information for display data RAM (DD RAM) and character generator RAM (CG RAM). The IR can be written, but not read by the MPU.

The DR temporally stores data to be written to /read from the DD RAM or CG RAM. The data written to DR by the MPU, is automatically written to the DD RAM or CG RAM as an internal operation.

When an address code is written to IR, the data is automatically transferred from the DD RAM or CG RAM to the DR. data transfer between the MPU is then completed when the MPU reads the DR. likewise, for the next MPU read of the DR, data in DD RAM or CG RAM at the address is sent to the DR automatically. Similarly, for the MPU write of the DR, the next DD RAM or CG RAM address is selected for the write operation.

The dot-matrix liquid crystal display controller and driver LSI displays alphanumeric, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver

4.12 REGULATORS:

The output voltage from the capacitor is more filtered & finally regulated. The voltage regulator

is a device, which maintains the output voltage constant irrespective of the change in supply

variations, load variations & temperature changes. Here we use fixed voltage regulator namely LM7805. The IC LM7805 is a +5v regulator which is used for microcontroller.

4.13 FEATURES AND DESCRIPTION OF REGULATORS

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

CHAPTER - 5

NODE MCU

NodeMCU is an open source IoT platform.^{[4][5]} It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.^{[6][7]} The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson^[8] and SPIFFS

NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems^[6] began production of the ESP8266.^[10] The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core,^[citation needed] widely used in IoT applications (see related projects). NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub.^[11] Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9.^[12] Later that month, Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform,^[13] and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib^[14] to NodeMCU project,^[15] enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

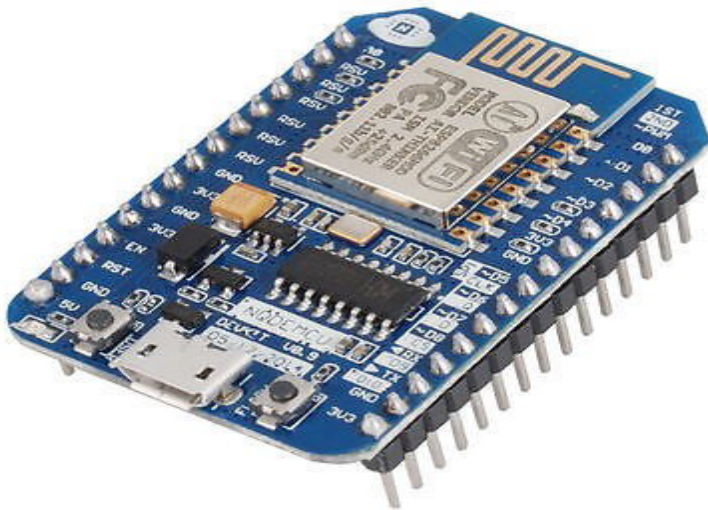
In summer 2015 the creators abandoned the firmware project and a group of independent contributors took over. By summer 2016 the NodeMCU included more than 40 different modules. Due to resource constraints users need to select the modules relevant for their project and build a firmware tailored to their needs.

ESP8266 Arduino Core[edit]

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE".^[16] This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs.

Introduction

NodeMCU is an open source LUA based firmware developed for ESP8266 wifi chip. By exploring functionality with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board/kit i.e. NodeMCU Development board.

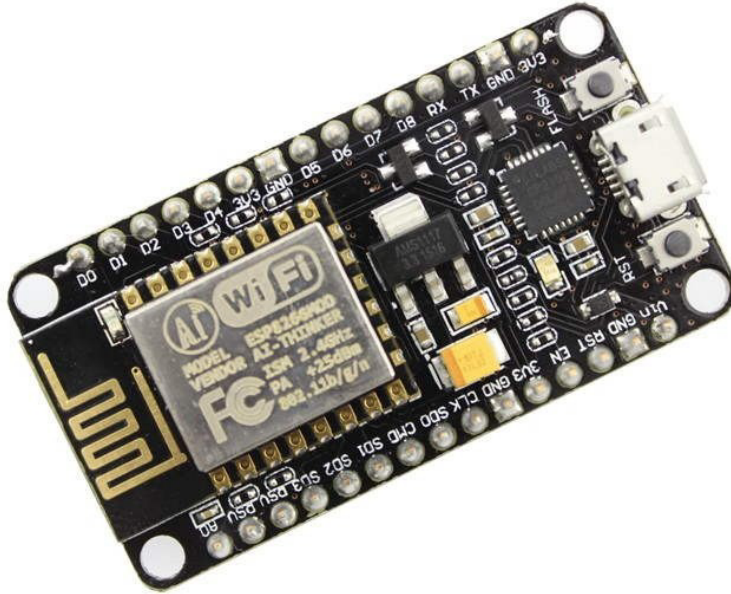


5.1 NodeMCU Development Board/kit v0.9 (Version1)

Since NodeMCU is open source platform, their hardware design is open for edit/modify/build.

NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 WiFi Module.

There is Version2 (V2) available for NodeMCU Dev Kit i.e. **NodeMCU Development Board v1.0 (Version2)**, which usually comes in black colored PCB.



5.2 NodeMCU Development Board/kit v1.0 (Version2)

For more information about NodeMCU Boards available in market refer [NodeMCU Development Boards](#)

NodeMCU Dev Kit has **Arduino like** Analog (i.e. A0) and Digital (D0-D8) pins on its board. It supports serial communication protocols i.e. UART, SPI, I2C etc.

Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc.

How to start with NodeMCU?

NodeMCU Development board is featured with wifi capability, analog pin, digital pins and serial communication protocols.

To get start with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.

There is online NodeMCU custom builds available using which we can easily get our custom NodeMCU firmware as per our requirement.

To know more about how to build custom NodeMCU firmware online and download it refer [Getting started with NodeMCU](#)

How to write codes for NodeMCU?

After setting up ESP8266 with Node-MCU firmware, let's see the IDE (Integrated Development Environment) required for development of NodeMCU.

5.2.1 NodeMCU with ESPlorer IDE

[Lua](#) scripts are generally used to code the NodeMCU. Lua is an open source, lightweight, embeddable scripting language built on top of C programming language.

For more information about how to write Lua script for NodeMCU refer [Getting started with NodeMCU using ESPlorerIDE](#)

5.2.2 NodeMCU with Arduino IDE

Here is another way of developing NodeMCU with a well-known IDE i.e. Arduino IDE. We can also develop applications on NodeMCU using Arduino development environment. This makes easy for Arduino developers than learning new language and IDE for NodeMCU.

For more information about how to write Arduino sketch for NodeMCU refer [Getting started with NodeMCU using ArduinoIDE](#)

Difference in using ESPlorer and Arduino IDE

Well, there is a programming language difference we can say while developing application for NodeMCU using ESPlorer IDE and Arduino IDE.

We need to code in C\C++ programming language if we are using Arduino IDE for developing NodeMCU applications and Lua language if we are using ESPlorer IDE.

Basically, NodeMCU is Lua Interpreter, so it can understand Lua script easily. When we write Lua scripts for NodeMCU and send/upload it to NodeMCU, then they will get executes sequentially. It will not build binary firmware file of code for NodeMCU to write. It will send Lua script as it is to NodeMCU to get execute.

In Arduino IDE when we write and compile code, ESP8266 toolchain in background creates binary firmware file of code we wrote. And when we upload it to NodeMCU then it will flash all NodeMCU firmware with newly generated binary firmware code. In fact, it writes the complete firmware.

That's the reason why NodeMCU not accept further Lua scripts/code after it is getting flashed by Arduino IDE. After getting flashed by Arduino sketch/code it will be no more Lua interpreter and we got error if we try to upload Lua scripts. To again start with Lua script, we need to flash it with NodeMCU firmware.

Since Arduino IDE compile and upload/writes complete firmware, it takes more time than ESPlorer IDE.

5.3 IR (Infrared) Sensor

An infrared sensor is an electronic device that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measures only infrared radiation, rather than emitting it that is called as a passive IR sensor. Usually in the infrared spectrum, all the objects radiate some form of thermal radiations. These types of radiations are invisible to our eyes that can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and these output voltages, change in proportion to the magnitude of the IR light received.

5.4 IR Sensor Circuit Diagram and Working Principle

An infrared sensor circuit is one of the basic and popular sensor module in an electronic device. This sensor is analogous to human's visionary senses, which can be used to detect obstacles and it is one of the common applications in real time. This circuit comprises of the following components

- LM358 IC 2 IR transmitter and receiver pair
- Resistors of the range of kilo ohms.
- Variable resistors.
- LED (Light Emitting Diode).

In this project, the transmitter section includes an IR sensor, which transmits continuous IR rays to be received by an IR receiver module. An IR output terminal of the receiver varies depending upon its receiving of IR rays. Since this variation cannot be analyzed as such,

therefore this

output can be fed to a comparator circuit. Here an operational amplifier (op-amp) of LM 339 is used as comparator circuit. When the IR receiver does not receive a signal, the potential at the inverting input goes higher than that non-inverting input of the comparator IC (LM339). Thus the output of the comparator goes low, but the LED does not glow. When the IR receiver module receives signal the potential at the inverting input goes low. Thus the output of the comparator (LM 339) goes high and the LED starts glowing.

5.5 IR Sensor circuit

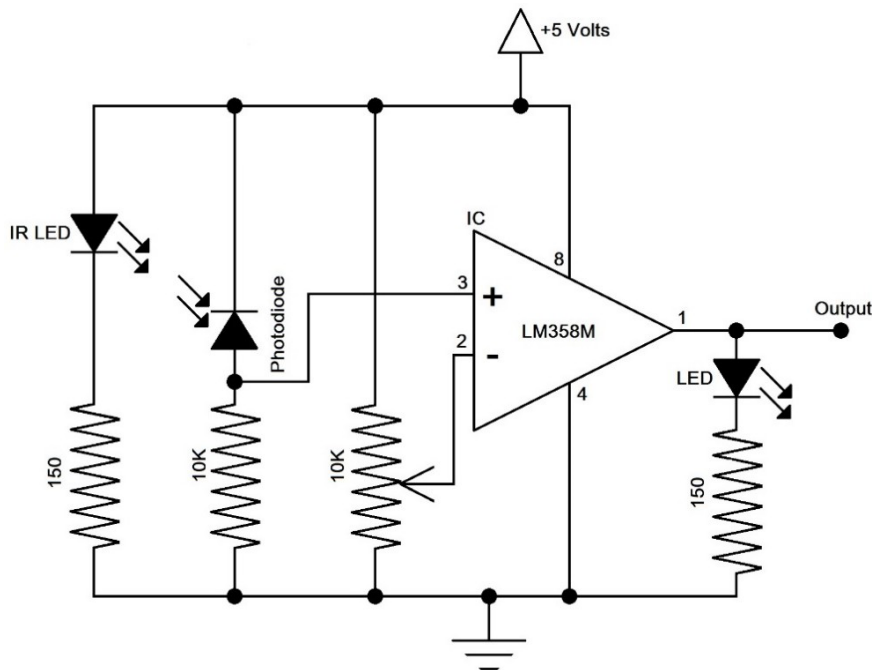


Fig. IR Sensor Circuit diagram

Resistor R1 (100), R2 (10k) and R3 (330) are used to ensure that minimum 10 mA current passes through the IR LED Devices like Photodiode and normal LEDs respectively. Resistor VR2 (preset=5k) is used to adjust the output terminals. Resistor VR1 (preset=10k) is used to set the sensitivity of the circuit Diagram. Read more about IR sensors.

5.5.1 Different Types of IR Sensors and Their Applications

IR sensors are classified into different types depending on the applications. The speed sensor is used for synchronizing the speed of multiple motors. The temperature sensor is used for industrial temperature control. PIR sensor is used for automatic door opening system and Ultrasonic sensor are used for distance measurement.

5.5.2 IR Sensor Applications

IR sensors are used in various Sensor based projects and also in various electronic devices which measures the temperature that are discussed in the below.

5.5.3 Radiation Thermometers

IR sensors are used in radiation thermometers to measure the temperature depend upon the temperature and the material of the object and these thermometers have some of the following features

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

5.6 Flame Monitors

These types of devices are used for detecting the light emitted from the flames and to monitor how the flames are burning. The Light emitted from flames extend from UV to IR region types. PbS, PbSe, Two-color detector, pyro electric detector are some of the commonly employed detector used in flame monitors.

5.7 Moisture Analyzers

Moisture analyzers use wavelengths which are absorbed by the moisture in the IR region. Objects are irradiated with light having these wavelengths (1.1 μm , 1.4 μm , 1.9 μm , and 2.7 μm) and also with reference wavelengths. The Lights reflected from the objects depend upon the moisture content and is detected by analyzer to measure moisture (ratio of reflected light at these wavelengths to the reflected light at reference wavelength). In GaAs PIN photodiodes, Pbs photoconductive detectors are employed in moisture analyzer circuits.

5.8 IR Imaging Devices

IR image device is one of the major applications of IR waves, primarily by virtue of its property that is not visible. It is used for thermal imagers, night vision devices, etc.



For examples Water, rocks, soil, vegetation, an atmosphere, and human tissue all features emit IR radiation. The Thermal infrared detectors measure these radiations in IR range and map the spatial temperature distributions of the object/area on an image. Thermal imagers usually composed of a Sb (indium antimonite), Gd Hg (mercury-doped germanium), Hg Cd Te (mercury - cadmium-telluride) sensors.

An electronic detector is cooled to low temperatures using liquid helium or liquid nitrogen's. Then the Cooling the detectors ensures that the radiant energy (photons) recorded by the detectors comes from the terrain and not from the ambient temperature of objects within the scanner itself an IR imaging electronic devices.

Thus, this is all about IR sensor circuit with working and applications. These sensors are used in many sensor based electronics projects.

CHAPTER - 6

Radio frequency identification (RFID) is a form of wireless communication that uses radio waves to identify and track objects. RFID takes the barcoding concept and digitizes it for the modern world providing the ability to:

- Uniquely identify an individual item beyond just its product type
- Identify items without direct line-of-sight
- Identify many items (up to 1,000s) simultaneously
- Identify items within a vicinity of between a few centimetres to several meters

An RFID system has *readers* and *tags* that communicate with each other by radio. RFID tags are so small and require so little power that they don't even need a battery to store information and exchange data with readers. This makes it easy and cheap to apply tags to all kinds of things that people would like to identify or track.

RFID technology has the capability to both greatly enhance and protect the lives of consumers, and also revolutionize the way companies do business. As the most flexible auto-identification technology, RFID can be used to track and monitor the physical world automatically and with accuracy.

RFID can tell you what an object is, where it is, and even its condition, which is why it is integral to the development of the Internet of Things—a globally interconnected web of objects allowing the physical world itself to become an information system, automatically sensing what is happening, sharing related data, and responding.

RFID use is increasing rapidly with the capability to “tag” any item with an inexpensive communications chip and then read that tag with a reader. Endless applications range from supply chain management to asset tracking to authentication of frequently counterfeited pharmaceuticals. Applications are limited, in fact, only by the imagination of the user.



6.1 RFID Applications



6.2 RFID can help:

- Automate inventory and asset-tracking in healthcare, manufacturing, retail, and business sectors
- Identify the source of products, enabling intelligent recall of defective or dangerous items, such as tainted foods, defective toys, and expired or compromised medication
- Prevent use of counterfeit products in the supply chain
- Improve shopping experience for consumers, with fewer out-of-stock items and easier returns
- Provide visibility into the supply chain, yielding a more efficient distribution channel and reduced business costs
- Decrease business revenue lost to theft or inaccurate accounting of goods
- Improve civilian security through better cargo monitoring at ports
- Wirelessly lock, unlock and configure electronic devices
- Enable access control of certain areas or devices

Whatever the application, RFID has the potential to increase efficiency of operations, improve asset visibility and traceability, decrease reliance on manual processes, reduce operations costs, and provide useful data for business analytics.

A basic RFID solution includes:

Tag

- Tag chips or integrated circuits (ICs)
- Tag antennas

Reader

- Reader antenna
- Reader control & application software

In a basic RFID system, tags are attached to all items that are to be tracked. These tags are made from a tiny tag-chip, sometimes called an integrated circuit (IC), that is connected to an antenna that can be built into many different kinds of tags including apparel hang tags, labels, and security tags, as well as a wide variety of industrial asset tags. The tag chip contains memory which stores the product's electronic product code (EPC) and other variable information so that it can be read and tracked by RFID readers anywhere.

An RFID reader is a network connected device (fixed or mobile) with an antenna that sends power as well as data and commands to the tags. The RFID reader acts like an access point for RFID tagged items so that the tags' data can be made available to business applications.

6.3 RFID Tags



An RFID tag is comprised of an integrated circuit (called an IC or chip) attached to an antenna that has been printed, etched, stamped or vapor-deposited onto a mount which is often a paper substrate or PolyEthylene Therephtalate (PET). The chip and antenna combo, called an inlay, is then converted or sandwiched between a printed label and its adhesive backing or inserted into a more durable structure.



Tag Chip

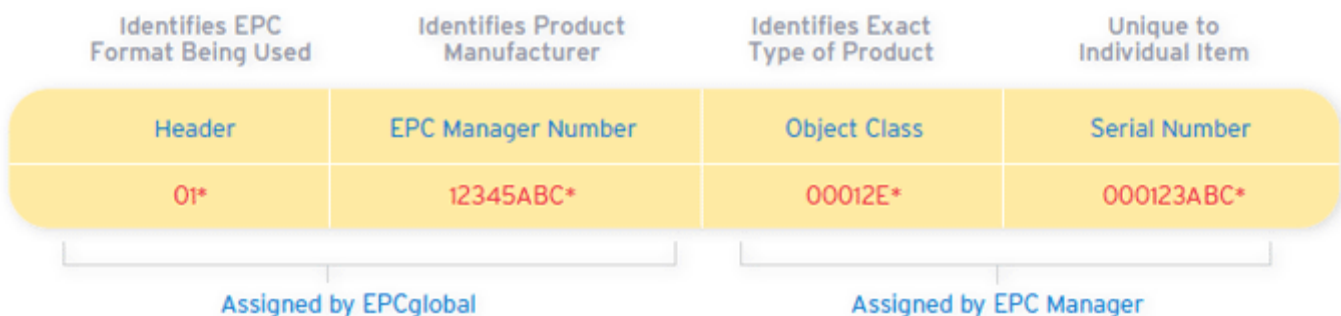
The tag's chip or integrated circuit (IC) delivers performance, memory and extended features to the tag. The chip is pre-programmed with a tag identifier (TID), a unique serial number assigned by the chip manufacturer, and includes a memory bank to store the items' unique tracking identifier (called an electronic product code or EPC).

Electronic Product Code (EPC)



950110126000000629

The electronic product code (EPC) stored in the tag chip's memory is written to the tag by an RFID printer and takes the form of 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 EPCglobal 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. The total electronic product code number can be used as a key into a global database to uniquely identify that particular product.



* Hexidecimal Number where A-F=10-15



Tag Antennas

Tag antennas collect energy and channel it to the chip to turn it on. Generally, the larger the tag antenna's area, the more energy it will be able to collect and channel toward the tag chip, and the further read range the tag will have.

There is no perfect antenna for all applications. It is the application that defines the antenna specifications. Some tags might be optimized for a particular **frequency band**, while others might be tuned for good performance when attached to materials that may not normally work well for wireless communication (certain liquids and metals, for example). Antennas can be made from a variety of materials; they can be printed, etched, or stamped with conductive ink, or even vapor deposited onto labels.

Tags that have only a single antenna are not as reliable as tags with multiple antennas. With a single antenna, a tag's orientation can result in "dead zones", or areas on the tag where incoming signals cannot be easily harvested to provide sufficient energy to power on the chip and communicate with the reader. A tag with dual antennas is able to eliminate these dead zones and increase its readability but requires a specialized chip.

6.4 RFID Readers



An RFID reader, also known as an interrogator, is a device that provides the connection between the tag data and the enterprise system software that needs the information. The reader communicates with tags that are within its field of operation, performing any number of tasks including simple continuous inventorying, filtering (searching for tags that meet certain criteria), writing (or encoding) to selected tags, etc.

The reader uses an attached antenna to capture data from tags. It then passes the data to a computer for processing. Just like RFID tags, there are many different sizes and types of RFID readers. Readers can be affixed in a stationary position in a store or factory, or integrated into a mobile device such as a portable, handheld scanner. Readers can also be embedded in electronic equipment or devices, and in vehicles.



Reader Antennas

RFID readers and reader antennas work together to read tags. Reader antennas convert electrical current into electromagnetic waves that are then radiated into space where they can be received by a tag antenna and converted back to electrical current. Just like tag antennas, there is a large variety of reader antennas and optimal antenna selection varies according to the solution's specific application and environment.

The two most common antenna types are linear- and circular-polarized antennas. Antennas that radiate linear electric fields have long ranges, and high levels of power that enable their signals to penetrate through different materials to read tags. Linear antennas are sensitive to tag orientation; depending on the tag angle or placement, linear antennas can have a difficult time reading tags. Conversely, antennas that radiate circular fields are less sensitive to orientation, but are not able to deliver as much power as linear antennas.

Choice of antenna is also determined by the distance between the RFID reader and the tags that it needs to read. This distance is called read range. Reader antennas operate in either a "near-field" (short range) or "far-field" (long range). In near-field applications, the read range is less than 30 cm and the antenna uses magnetic coupling so the reader and tag can transfer power. In near-field systems, the readability of the tags is not affected by the presence of dielectrics such as water and metal in the field.

In far-field applications, the range between the tag and reader is greater than 30 cm and can be up to several tens of meters. Far-field antennas utilize electromagnetic coupling and dielectrics can weaken communication between the reader and tags.



Reader Control and Application Software

Reader control and application software, also known as middleware, helps connect RFID readers with the applications they support. The middleware sends control commands to the reader and receives tag data from the reader.

6.5 Creating an RFID Solution

Deploying an RFID system necessitates multiple actors and many different components. Typically, installing a system requires basic hardware-including tag chips, tag antennas, readers, and reader antennas-as well as reader control and application software, and solution providers to put it all together. When all of these components come together, an infinite

number of creative

applications are possible. RFID system applications can help improve the quality of business operations, inventory and customer experience in a variety of industries.

Frequency refers to the size of the radio waves used to communicate between RFID system components. RFID systems throughout the world operate in **low frequency (LF)**, **high frequency (HF)** and **ultra-high frequency (UHF) bands**. Radio waves behave differently at each of these frequencies with advantages and disadvantages associated with using each frequency band.

If an RFID system operates at a lower frequency, it has a shorter read range and slower data read rate, but increased capabilities for reading near or on metal or liquid surfaces. If a system operates at a higher frequency, it generally has faster data transfer rates and longer read ranges than lower frequency systems, but more sensitivity to radio wave interference caused by liquids and metals in the environment.

6.5.1 LF RFID

The LF band covers frequencies from 30 KHz to 300 KHz. Typically LF RFID systems operate at 125 KHz, although there are some that operate at 134 KHz. This frequency band provides a short read range of 10 cm, and has slower read speed than the higher frequencies, but is not very sensitive to radio wave interference.

LF RFID applications include access control and livestock tracking.

Standards for LF animal-tracking systems are defined in ISO 14223, and ISO/IEC 18000-2. The LF spectrum is not considered a truly global application because of slight differences in frequency and power levels throughout the world.

6.5.2 HF RFID

The HF band ranges from 3 to 30 MHz. Most HF RFID systems operate at 13.56 MHz with read ranges between 10 cm and 1 m. HF systems experience moderate sensitivity to interference.

HF RFID is commonly used for ticketing, payment, and data transfer applications.

There are several HF RFID standards in place, such as the ISO 15693 standard for tracking items, and the ECMA-340 and ISO/IEC 18092 standards for Near Field Communication (NFC), a shortrange technology that is commonly used for data exchange between devices. Other HF standards include the ISO/IEC 14443 A and ISO/IEC 14443 standards for MIFARE technology, which used in smart cards and proximity cards, and the JIS X 6319-4 for FeliCa, which is a smart card system commonly used in electronic money cards.

6.6 UHF RFID

The UHF frequency band covers the range from 300 MHz to 3 GHz. Systems complying with the UHF Gen2 standard for RFID use the 860 to 960 MHz band. While there is some variance in frequency from region to region, UHF Gen2 RFID systems in most countries operate between 900 and 915 MHz.

The read range of passive UHF systems can be as long as 12 m, and UHF RFID has a faster data transfer rate than LF or HF. UHF RFID is the most sensitive to interference, but many UHF product manufacturers have found ways of designing tags, antennas, and readers to keep performance high even in difficult environments. Passive UHF tags are easier and cheaper to manufacture than LF and HF tags.

UHF RFID is used in a wide variety of applications, ranging from retail inventory management, to pharmaceutical anti-counterfeiting, to wireless device configuration. The bulk of new RFID projects are using UHF opposed to LF or HF, making UHF the fastest growing segment of the RFID market.

The UHF frequency band is regulated by a single global standard called the ECPglobal Gen2 (ISO 18000-6C) UHF standard.

UHF • Single worldwide Gen2 standard • 20x the range and speed of HF • Labels cost 5¢–15¢ in 2012 • The technology for item tagging	HF and LF • Multiple competing standards • HF-based NFC for secure payment • Labels, cards, inlays cost 50¢–\$2 • Used in immobilizers, ticketing, payment
---	--

6.6.1 **Passive, Active, and BAP RFID Systems**

Active RFID Systems

In active RFID systems, tags have their own transmitter and power source. Usually, the power source is a battery. Active tags broadcast their own signal to transmit the information stored on their microchips.

Active RFID systems typically operate in the **ultra-high frequency (UHF) band** and offer a range of up to 100 m. In general, active tags are used on large objects, such as rail cars, big reusable containers, and other assets that need to be tracked over long distances.

There are two main types of active tags: transponders and beacons. **Transponders** are “woken up” when they receive a radio signal from a reader, and then power on and respond by

transmitting a signal back. Because transponders do not actively radiate radio waves until they receive a reader signal, they conserve battery life.

Beacons are used in most **real-time locating systems (RTLS)**, in order to track the precise location of an asset continuously. Unlike transponders, beacons are not powered on by the reader's signal. Instead, they emit signals at pre-set intervals. Depending on the level of locating accuracy required, beacons can be set to emit signals every few seconds, or once a day. Each beacon's signal is received by reader antennas that are positioned around the perimeter of the area being monitored, and communicates the tag's ID information and position.

6.6.2 Passive RFID Systems

In passive RFID systems, the reader and reader antenna send a radio signal to the tag. The RFID tag then uses the transmitted signal to power on, and reflect energy back to the reader.

Passive RFID systems can operate in the **low frequency (LF)**, **high frequency (HF)** or **ultra-high frequency (UHF)** radio bands. As passive system ranges are limited by the power of the tag's **backscatter** (the radio signal reflected from the tag back to the reader), they are typically less than 10 m. Because passive tags do not require a power source or transmitter, and only require a tag chip and antenna, they are cheaper, smaller, and easier to manufacture than active tags. Passive tags can be packaged in many different ways, depending on the specific RFID application requirements. For instance, they may be mounted on a substrate, or sandwiched between an adhesive layer and a paper label to create smart RFID labels. Passive tags may also be embedded in a variety of devices or packages to make the tag resistant to extreme temperatures or harsh chemicals.

Passive RFID solutions are useful for many applications, and are commonly deployed to track goods in the supply chain, to inventory assets in the retail industry, to authenticate products such as pharmaceuticals, and to embed RFID capability in a variety of devices. Passive RFID can even be used in warehouses and distribution centers, in spite of its shorter range, by setting up readers at choke points to monitor asset movement.

6.6.3 Battery-Assisted Passive (BAP) Systems

A Battery-Assisted Passive RFID tag is a type of passive tag which incorporates a crucial active tag feature. While most passive RFID tags use the energy from the RFID reader's signal to power on the tag's chip and backscatter to the reader, BAP tags use an integrated power source (usually a battery) to power on the chip, so all of the captured energy from the reader can be used for backscatter. Unlike transponders, BAP tags do not have their own transmitters.

	Active RFID	Passive RFID	Battery-Assisted Passive (BAP)
Tag Power Source	Internal to	Energy transfer from	Tag uses internal power source to power on, and energy transferred from the reader via RF to backscatter
Tag Battery	Yes	No	Yes
Availability of Tag Power	Continuous	Only within field of reader	Only within field of reader
Required Signal Strength from Reader to Tag	Very Low	Very high (must power the tag)	Moderate (does not need to power tag, but must power
Available Signal Strength from Tag to Reader	High	Very Low	Moderat
Communication Range	Long Range (100m or more)	Short range (up to 10m)	Moderate range (up to
Sensor Capability	Ability to continuously monitor and record sensor input	Ability to read and transfer sensor values only when tag is powered by reader	Ability to read and transfer sensor values only when tag receives RF signal from reader

CHAPTER - 7

LCD Display

Liquid crystal displays (LCDs) have materials which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal.

An LCD consists of two glass panels, with the liquid crystal material sandwiched in between them. The inner surface of the glass plates are coated with transparent electrodes which define the character, symbols or patterns to be displayed. Polymeric layers are present in between the electrodes and the liquid crystal, which makes the liquid crystal molecules to maintain a defined orientation angle.

On each polariser is pasted outside the two glass panels. These polarisers would rotate the light rays passing through them to a definite angle, in a particular direction.

When the LCD is in the off state, light rays are rotated by the two polarisers and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent.

When sufficient voltage is applied to the electrodes, the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarisers, which would result in activating / highlighting the desired characters.

The LCD's are lightweight with only a few millimeters thickness. Since the LCD's consume less power, they are compatible with low power electronic circuits, and can be powered for long durations.

The LCD's don't generate light and so light is needed to read the display. By using backlighting, reading is possible in the dark. The LCD's have long life and a wide operating temperature range.

Changing the display size or the layout size is relatively simple which makes the LCD's more customer friendly.

The LCDs used exclusively in watches, calculators and measuring instruments are the simple seven-segment displays, having a limited amount of numeric data. The recent advances in technology have resulted in better legibility, more information displaying capability and a wider temperature range. These have resulted in the LCDs being extensively used in

telecommunications and entertainment electronics. The LCDs have even started replacing the

cathode ray tubes (CRTs) used for the display of text and graphics, and also in small TV applications.

Crystalonics dot –matrix (alphanumeric) liquid crystal displays are available in TN, STN types, with or without backlight. The use of C-MOS LCD controller and driver ICs result in low power consumption. These modules can be interfaced with a 4-bit or 8-bit micro processor /Micro controller.

The built-in controller IC has the following features:

- Correspond to high speed MPU interface (2MHz)
- 80 x 8 bit display RAM (80 Characters max)
- 9,920 bit character generator ROM for a total of 240 character fonts. 208 character fonts (5 x 8 dots) 32 character fonts (5 x 10 dots)
- 64 x 8 bit character generator RAM 8 character generator RAM 8 character fonts (5 x 8 dots) 4 characters fonts (5 x 10 dots)
- Programmable duty cycles
 - 1/8 – for one line of 5 x 8 dots with cursor
 - 1/11 – for one line of 5 x 10 dots with
 - cursor 1/16 – for one line of 5 x 8 dots with
 - cursor
- Wide range of instruction functions display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift.

7.1 FUNCTIONAL DESCRIPTION OF THE CONTROLLER IC

7.1.1 REGISTERS:

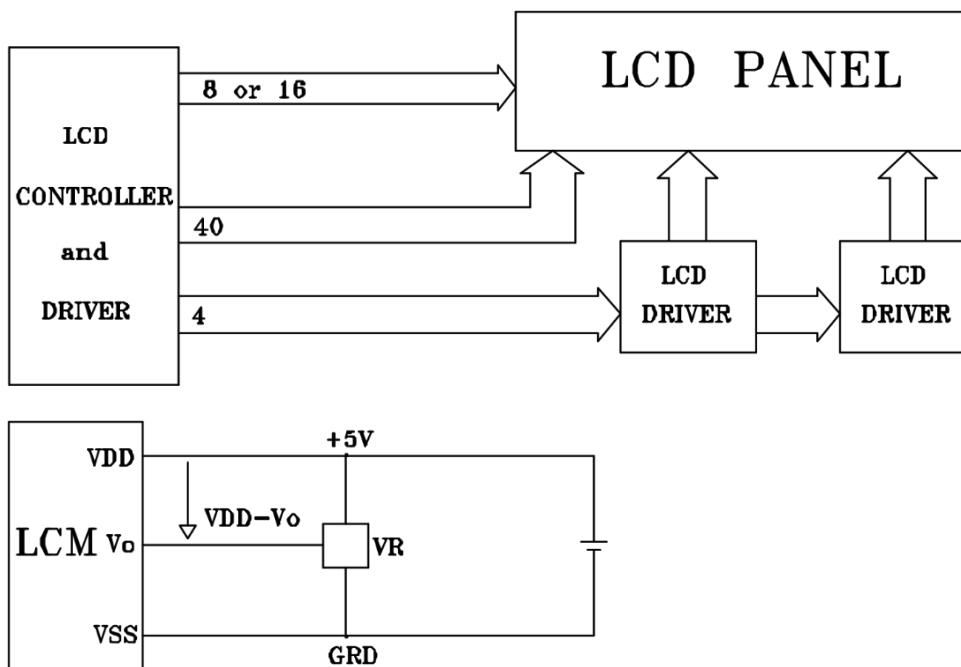
The controller IC has two 8 bit registers, an instruction register (IR) and a data register (DR). The IR stores the instruction codes and address information for display data RAM (DD RAM) and character generator RAM (CG RAM). The IR can be written, but not read by the MPU.

The DR temporally stores data to be written to /read from the DD RAM or CG RAM. The data written to DR by the MPU, is automatically written to the DD RAM or CG RAM as an internal operation.

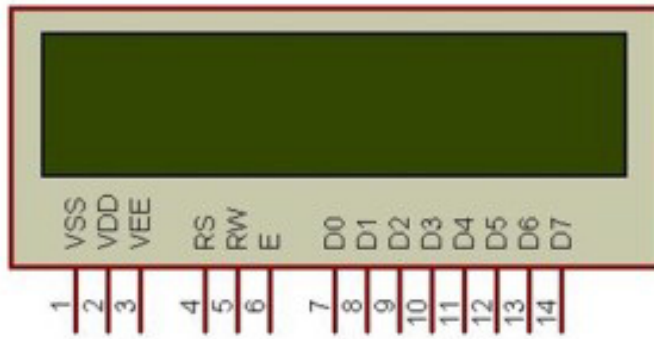
When an address code is written to IR, the data is automatically transferred from the DD RAM or CG RAM to the DR. data transfer between the MPU is then completed when the MPU reads the DR. likewise, for the next MPU read of the DR, data in DD RAM or CG RAM at the address is sent to the DR automatically. Similarly, for the MPU write of the DR, the next DD RAM or CG RAM address is selected for the write operation.

The dot-matrix liquid crystal display controller and driver LSI displays alphanumeric, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver.

7.2 Block Diagram



VDD-V_o: LCD DRIVING VOLTAGE VR: 10K-20K .



Absolute Maximum Ratings

Item	Symbol	Min	Max	Unit
Power Voltage	VDD - VSS	0	7.0	V
Input Voltage	V _{in}	V _{SS}	VDD	
Operating Temperature Range	T _{OP}	0	+50	T
Storage Temperature Range	T _{ST}	-20	+60	

7.3 Buzzer:

There are many ways to communicate between the user and a product. One of the best ways is audio communication using a buzzer IC. So during the design process, understanding some technologies with configurations is very helpful. So, this article discusses an overview of an audio signaling device like a beeper or a buzzer and its working with applications.

What is a Buzzer?

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



The **pin configuration of the buzzer** is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-' symbol or short terminal and it is connected to the GND terminal.

History

The history of an electromechanical buzzer and piezoelectric is discussed below.

Electromechanical

This buzzer was launched in the year 1831 by an American Scientist namely Joseph Henry but, this was used in doorbells until they were eliminated in 1930 in support of musical bells, which had a smooth tone.

Piezoelectric

These buzzers were invented by manufacturers of Japanese & fixed into a broad range of devices during the period of 1970s – 1980s. So, this development primarily came due to cooperative efforts through the manufacturing companies of Japanese. In the year 1951, they recognized the Application Research Committee of Barium Titanate that allows the corporations to be cooperative competitively & bring about numerous piezoelectric creations.

Specifications

The **specifications of the buzzer** include the following.

- Color is black
- The frequency range is 3,300Hz

- Operating Temperature ranges from – 20° C to +60°C
- Operating voltage ranges from 3V to 24V DC
- The sound pressure level is 85dBA or 10cm
- The supply current is below 15mA

7.3.1 Types of Buzzer

A buzzer is available in different types which include the following.

- Piezoelectric
- Electromagnetic
- Mechanical
- Electromechanical
- Magnetic

Piezoelectric

As the name suggests, the piezoelectric type uses the piezoelectric ceramic's piezoelectric effect & pulse current to make the metal plate vibrate & generate sound. This kind of buzzer is made with a resonance box, multi resonator, piezoelectric plate, housing, impedance matcher, etc. Some of the buzzers are also designed with LEDs.

The multi resonator of this mainly includes ICs and transistors. Once the supply is given to this resonator, it will oscillate and generates an audio signal with 1.5 to 2.kHz. The impedance matcher will force the piezoelectric plate to produce sound.

7.4 Electromagnetic

This type of buzzer is made with a magnet, solenoid coil, oscillator, housing, vibration diaphragm, and magnet. Once the power supply is given, the oscillator which produces the audio signal current will supply throughout the solenoid coil to generate a magnetic field.

Sometimes, the vibration diaphragm will vibrate & generates sound under the magnet & solenoid coil interaction. The frequency range of this ranges from 2 kHz to 4kHz.

7.5 Mechanical

These types of buzzers are subtypes of electromagnetic, so the components used in this type are also similar. But the main difference is that the vibrating buzzer is placed on the outside instead of the inside.

Electromechanical

The designing of these types of buzzers can be done with a bare metal disc & an electromagnet. The working principle of this is similar to magnetic and electromagnetic. It generates sound throughout the disc movement & magnetism.

7.6 Magnetic

Like a piezo type, magnetic is also used to generate a sound but they are different due to core functionality. The magnetic type is more fixed as compared to the piezo type because they work through a magnetic field.

Magnetic buzzers utilize an electric charge instead of depending on piezo materials to generate a magnetic field, after that it permits another element of the buzzer to vibrate & generate sound. The applications of magnetic buzzers are similar to the piezo type in household devices, alarms such as watches, clocks & keyboards.

Working Principle

The working principle of a buzzer depends on the theory that, once the voltage is given across a piezoelectric material, then a pressure difference is produced. A piezo type includes piezo crystals among two conductors.

Once a potential disparity is given across these crystals, then they thrust one conductor & drag the additional conductor through their internal property. So this continuous action will produce a sharp sound signal.

7.7 Mounting Configurations

The mounting configurations of buzzers include the following.

- Panel Mount
- Wire Leads
- Screw Terminals
- Through Hole
- Spring Contact
- Surface Mount

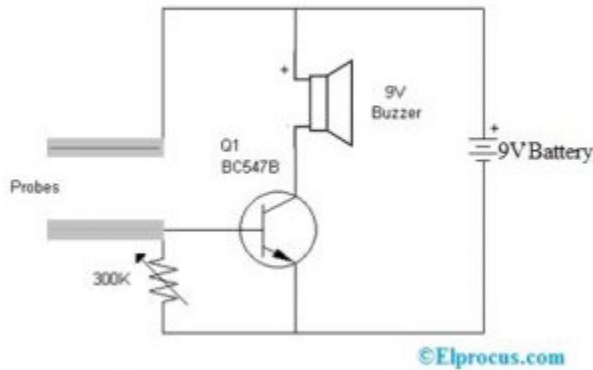
A buzzer is an efficient component to include the features of sound in our system or project. It is an extremely small & solid two-pin device thus it can be simply utilized on breadboard or PCB. So in most applications, this component is widely used.

There are two kinds of buzzers commonly available like simple and readymade. Once a simple type is power-driven then it will generate a beep sound continuously. A readymade type looks heavier & generates a Beep. Beep. Beep. This sound is because of the internal oscillating circuit within it.

This buzzer uses a DC power supply that ranges from 4V – 9V. To operate this, a 9V battery is used but it is suggested to utilize a regulated +5V/+6V DC supply. Generally, it is connected through a switching circuit to switch ON/OFF the buzzer at the necessary time interval.

7.8 Buzzer Circuit Diagram

The **circuit diagram of the water level indicator using the buzzer** is shown below. This circuit is used to sense or detect the water level within the tank or washing machine or pool, etc. This circuit is very simple to design using few components such as a transistor, buzzer, 300K variable resistor, and power supply or 9V battery.



7.9 Water Level Circuit using Buzzer

Once the two probes of the circuit are placed in the tank, it detects the level of water. Once the water level exceeds the fixed level, then it generates a beep sound through a buzzer connected to the circuit. This circuit uses a BC547B NPN transistor however we can also use any general-purpose transistor instead of using 2N3904/2N2222.

This water level sensor circuit working is very simple and the transistor used within the circuit works as a switch. Once the two probes notice the water level within the tank, then the transistor turns ON & the voltage begins flowing throughout the transistor to trigger the buzzer.

While choosing a buzzer or speaker, many principles need to consider like the following.

- Size of the product
- Consumption of Current
- Type of terminal
- Frequency Voltage
- Volume
- Type
- AC/DC Voltage
- The tone is Continuous/Pulsed
- Fixing – Pins, Leads/Surface Mount
- Output of Sound
- Feedback Option
- Piezo Elements

7.9.1 Advantages

The **advantages of a buzzer** include the following.

- Simply Compatible
- Frequency Response is Good
- Size is small
- Energy Consumption is less
- The Range of Voltage usage is Large
- Sound Pressure is high

7.9.2 Disadvantages

The **disadvantages of the buzzer** include the following.

- Controlling is a little hard
- Generates Annoying Sound
- Training is necessary to know how to repair the condition without just turning off.

7.9.3 Applications

The **applications of the buzzer** include the following.

- Communication Devices
- Electronics used in Automobiles
- Alarm Circuits
- Portable Devices
- Security Systems
- Timers
- Household Appliances
- Electronic Metronomes
- Sporting Events
- Annunciator Panels
- Game Shows

This tutorial will walk you through downloading, installing, and testing the Arduino software (also known as the Arduino IDE - short for Integrated Development Environment). Before you jump to the page for your operating system, make sure you've got all the right equipment.

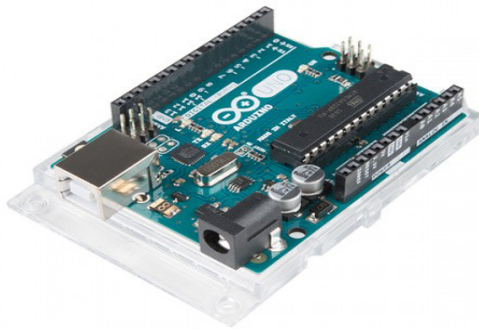
Required Materials

To follow along with this tutorial, you will need the following materials. You may not need everything though depending on what you have. Add it to your cart, read through the guide, and adjust the cart as necessary.

CHAPTER – 8

ARDUINO

- A computer (Windows, Mac, or Linux)
- An Arduino-compatible microcontroller (anything from [this guide](#) should work)
- A USB A-to-B cable, or another appropriate way to connect your Arduino-compatible microcontroller to your computer (check out this [USB buying guide](#) if you're not sure which cable to get).



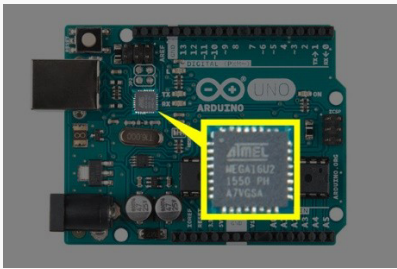
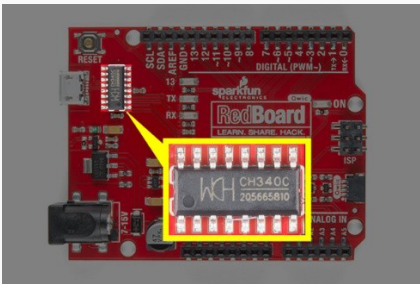
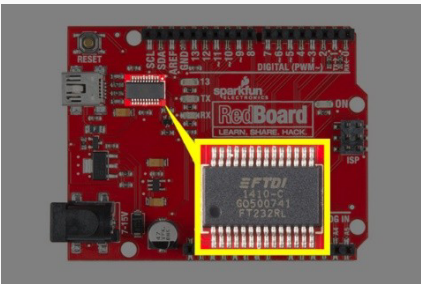
Uno

B -



Arduino
- R3
USB
Cable A to
6 Foot

Note: There are several variants that use the Arduino Uno R3 footprint. Depending on the design, you may need to install additional drivers for your USB-to-serial converter before you are able to upload code to your microcontroller. For example, the [RedBoard](#) uses an [FTDI](#) while the [RedBoard Plus](#) uses the [CH340](#).

		
<i>Atmega16U2 on the Arduino Uno R3</i>	<i>FTDI on the RedBoard</i>	<i>CH340 on the RedBoard Qwiic</i>

Both use different drivers compared to the Arduino Uno R3. Make sure to look closely at your board and its respective hookup guide to determine USB-to-serial converter that is on board. You will probably have either an FTDI or CH340 populated on the board.

How to Install FTDI Drivers

How to install drivers for the FTDI Basic on Windows, Mac OS X, and Linux.

How to Install CH340 Drivers

How to install CH340 drivers (if you need them) on Windows, Mac OS X, and Linux.

Other boards using a different architecture like the RedBoard Turbo (SAMD21) have a built-in USB communication, eliminating the need to have a separate piece of hardware. For more information, check out the blog post on drivers: What Drives your SparkFun Inventor's Kit?

Downloading the Arduino IDE

You can download the Arduino IDE from their website. They have installation instructions, but we will also go over the installation process as well. Make sure you download the version that matches your operating system.

[CLICK FOR ARDUINO IDE DOWNLOAD PAGE](#)

The installation procedure is fairly straightforward, but it does vary by OS. Here are some tips to help you along.

Troubleshooting Tips

- We recommend using a computer with a full desktop operating system like Windows 7/10 (**avoid** Windows 8 if you can), Mac OSX, and certain flavors Linux (check the Arduino FAQ

- If you are not a technical or computer savvy individual and you have your choice of computers, I highly recommend using a **Windows 7 or 10** computer. You will usually run into the least issues, if any, with these operating systems.
- We do **NOT** recommend using a Chromebook, Netbook, tablet, phone, or the Arduino Web IDE in general. You will be responsible for troubleshooting any driver or Arduino Web IDE issues.
- As of the writing of this tutorial (updated 12-14-2018), the most recent and stable release of the Arduino IDE is version 1.8.5. We recommend using that version of the Arduino IDE; you can download the [previous releases here](#).
- On Windows 10, we do **NOT** recommend installing the Arduino IDE from the app store. You may run into issues because the OS will automatically update to the most recent release of the Arduino IDE, which may have unknown bugs (like the compiler errors in versions 1.8.6 and 1.8.7).
- Raspberry Pi users with Raspbian installed should use the **Linux ARM** download. We do not recommend using the command line installation. It will install the oldest release of Arduino, which is useless when it comes to installing new boards definitions or libraries.
- For additional troubleshooting tips, here is a [troubleshooting guide from Arduino](#).

8.1 Windows

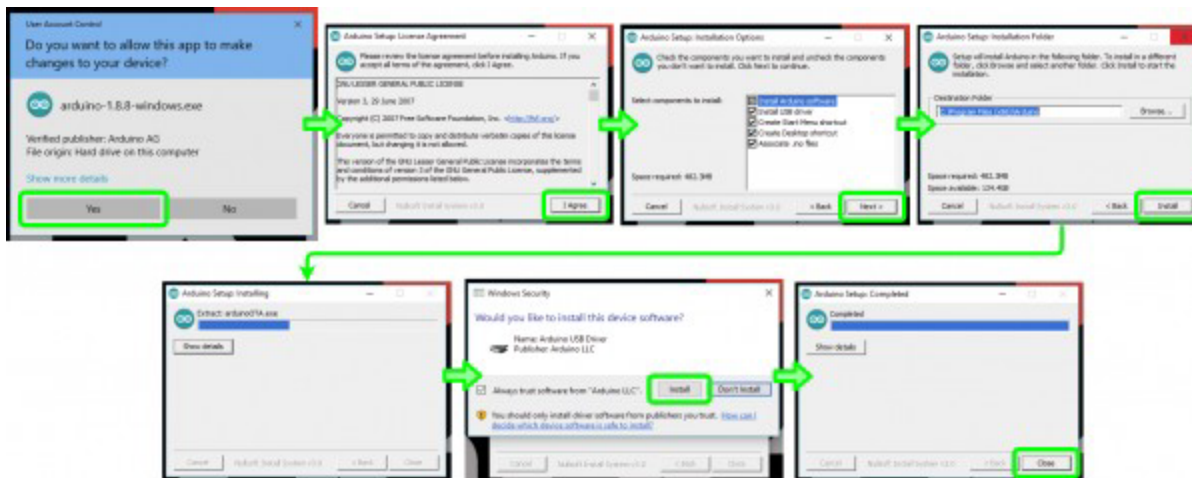
This page will show you how to install and test the Arduino software with a Windows operating system (Windows 10, Windows 7, Vista, and XP).

Go to the Arduino [download page](#) and download the latest version of the Arduino software for Windows if you have not already.

[CLICK FOR ARDUINO IDE DOWNLOAD PAGE](#)

Installer

The Windows version of Arduino is offered in two options: an installer or a zip file. The **installer** is the easier of the two options, just download that, and run the executable file to begin the installation.



Windows install steps. Click the image to get a bigger view.

When you're prompted to install a driver during installation, select "Install". This will install drivers for Arduino specific boards (like the Uno, Nano, etc.) that you may use in the future.

8.2 ZIP

If you choose to download the **zip file** version of Arduino, you'll need to extract the files yourself. Don't forget which folder you extract the files into! You will need to run the executable Arduino file in the folder to start the Arduino IDE.

When the download is finished, un-zip it and open up the Arduino folder to confirm that yes, there are indeed some files and sub-folders inside. The file structure is important so don't be moving any files around unless you really know what you're doing.

Note: On Windows 10, there is an option to install Arduino through their app store. we do not recommend installing the Arduino IDE from the app store. You may run into issues because the OS will automatically update to the most recent release of the Arduino IDE, which may have unknown bugs.

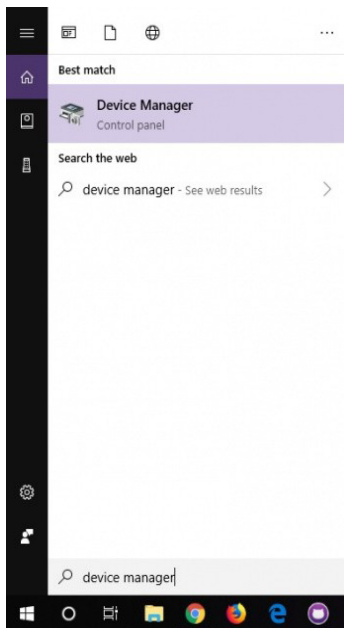
Connecting Your Arduino

Power up your Arduino by connecting your Arduino board to your computer with a USB cable (or FTDI cable if you're using an Arduino Pro). You should see the an LED labeled 'ON' light up. ([this diagram](#) shows the placement of the power LED on the UNO).

Drivers for Arduino Uno on Windows

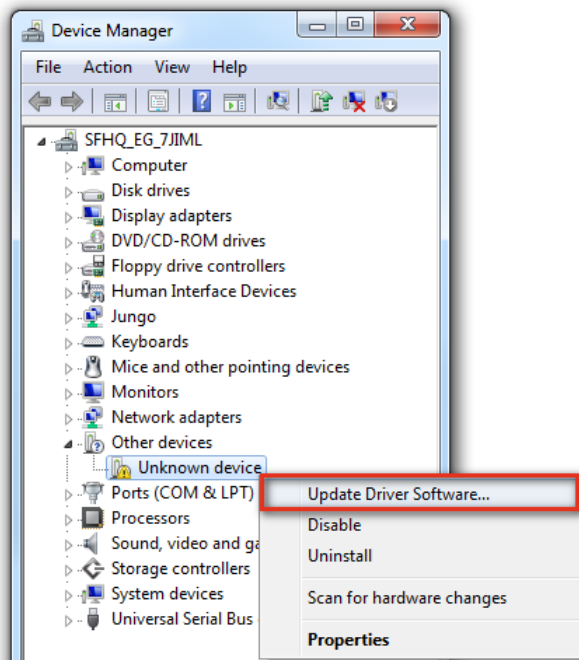
To install the drivers for the Arduino Uno, you will need to plug in your board to your computer's USB port. Once the board is connected, you will need to wait for Windows to begin it's driver installation process. After a few moments, the process will probably fail, despite its best efforts.

Open up a search, type in **Device Manager**, and hit **ENTER**.

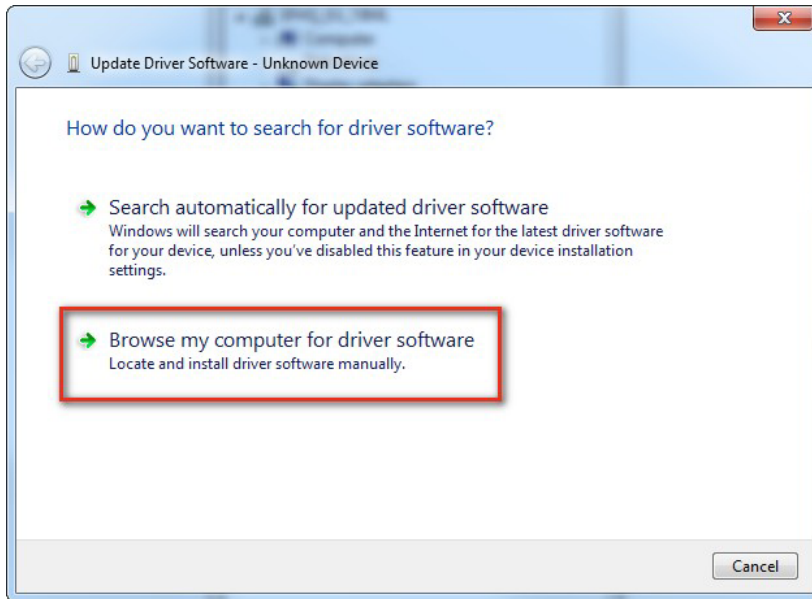


Note: Searching is the easiest method to open the Device Manager. However, there is more than one method of opening the device manager. The longer method is to click on the **Start Menu > Windows System > System and Security > System > Device Manager**.

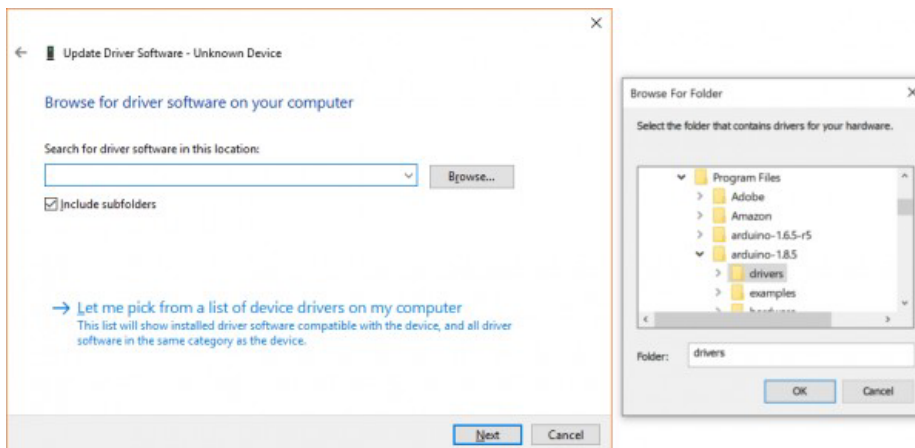
Look under Ports (COM & LPT) tree. You should see an open port named "**Arduino UNO (COMxx)**". If there is no COM & LPT section, look under "**Other devices**" for "**Unknown Device**". Right click on the "**Arduino UNO (COMxx)**" or "**Unknown device**" port and choose the "**Update Driver Software**" option.



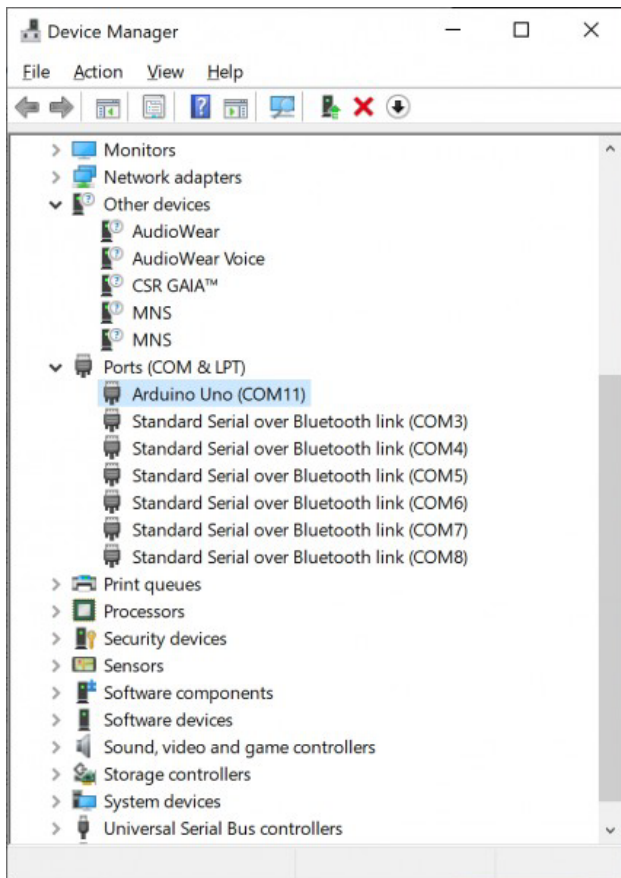
Next, choose the "**Browse my computer for Driver software**" option



Finally, navigate to the Arduino IDE folder. This should be where you unzipped the Arduino IDE (e.g. it should be similar to the following path with a different version number: *C:\Program Files\arduino-1.8.5\drivers*). Depending on what version of Windows you have, you may be able to select the Uno's driver file, named "**Arduino.inf**", located in the "**Drivers**" folder (not the "FTDI USB Drivers" sub-directory). If you cannot see the ***.inf** file, it is probably just hidden. You can select the "**drivers**" folder with the "**include sub-folders**" option selected instead.



Windows will finish up the driver installation from there! Your computer will enumerate with a COM port. You may see a COM port number depending on what is currently saved in your computer. Try to remember what the number is when uploading. If not, you can always navigate back to the device manager and power cycle the Arduino to determine what number your Arduino enumerated on.



For earlier versions of the Arduino boards (e.g. Arduino Duemilanove, Nano, or Diecimila) check out [this page](#) for specific directions.

8.3 ARDUINO.CC - GUIDE: WINDOWS

Drivers for RedBoard on Windows

Depending on your board, there may be different USB-to-serial converters on your board. If you are using a RedBoard you will need to go to the [How to Install FTDI Drivers](#) tutorial. If you are using the or RedBoard Qwiic, you will need to go to the [How to Install CH340 Drivers](#) tutorial. Make sure to look closely at your board and its respective hookup guide to determine USB-to-serial converter that is on board.

How to Install FTDI Drivers

JUNE 4, 2013

How to install drivers for the FTDI Basic on Windows, Mac OS X, and Linux.

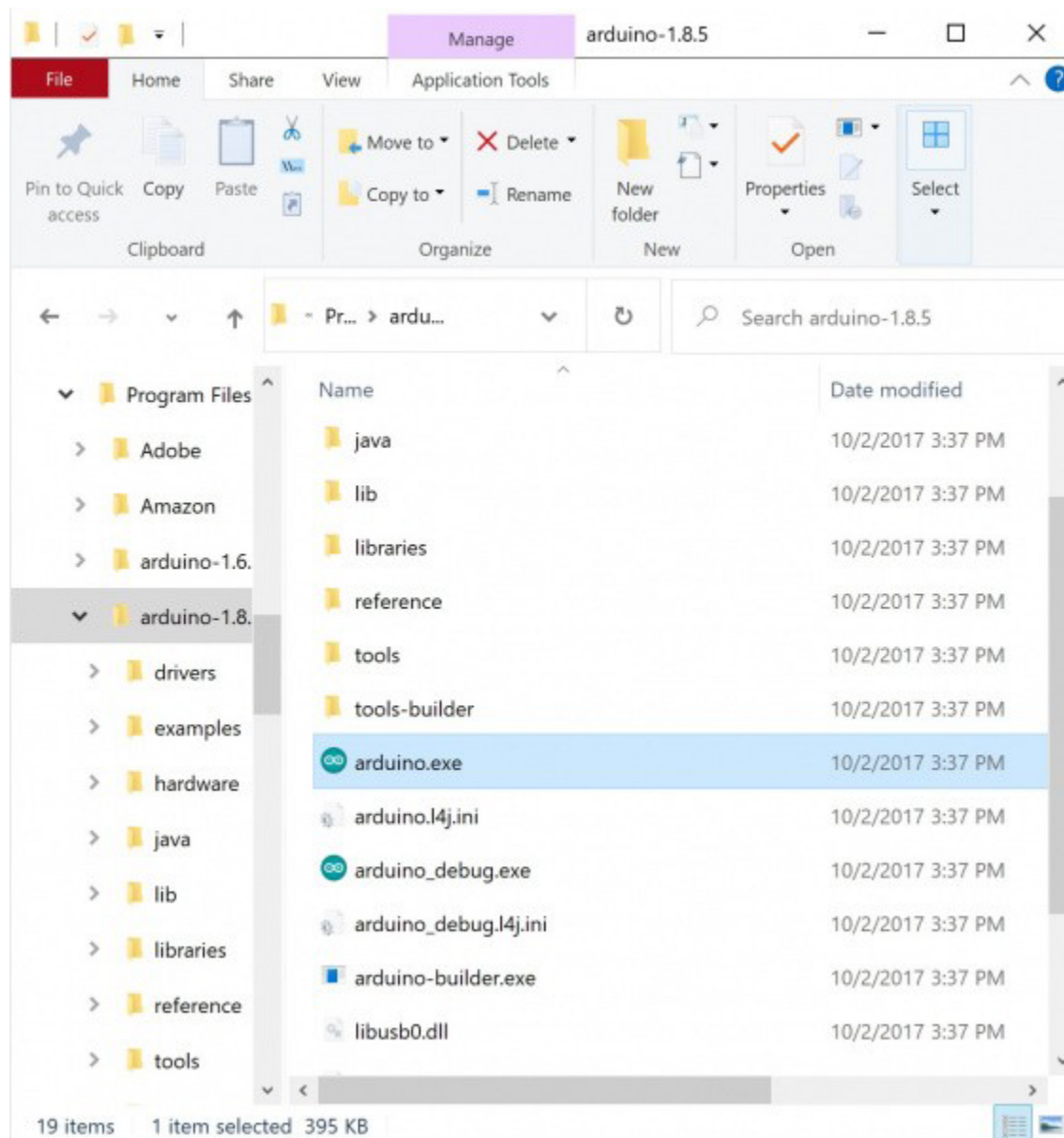
How to Install CH340 Drivers

AUGUST 6, 2019

How to install CH340 drivers (if you need them) on Windows, Mac OS X, and Linux.

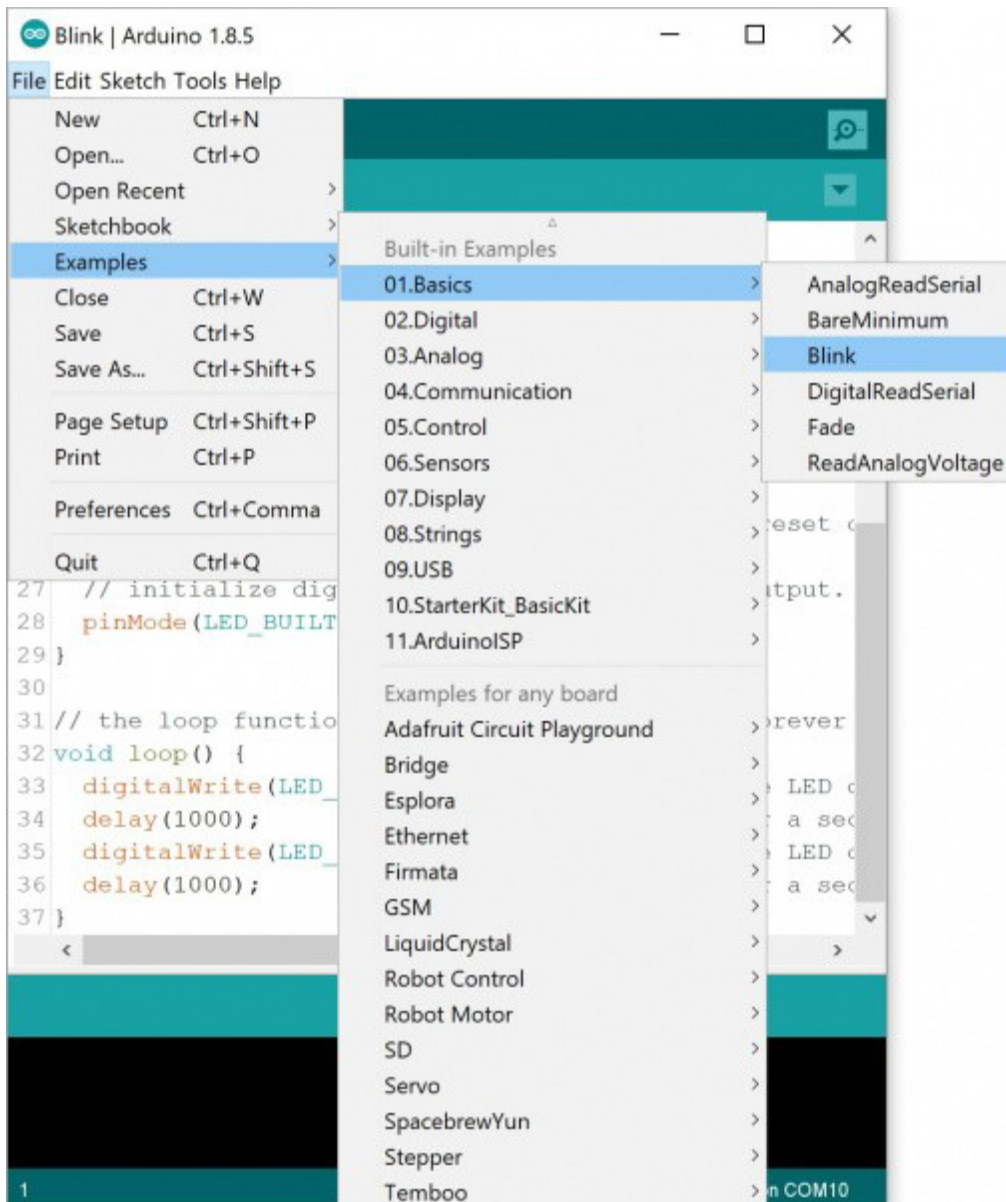
Blink!

After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board! Launch the Arduino application. If you disconnected your board, plug it back in.

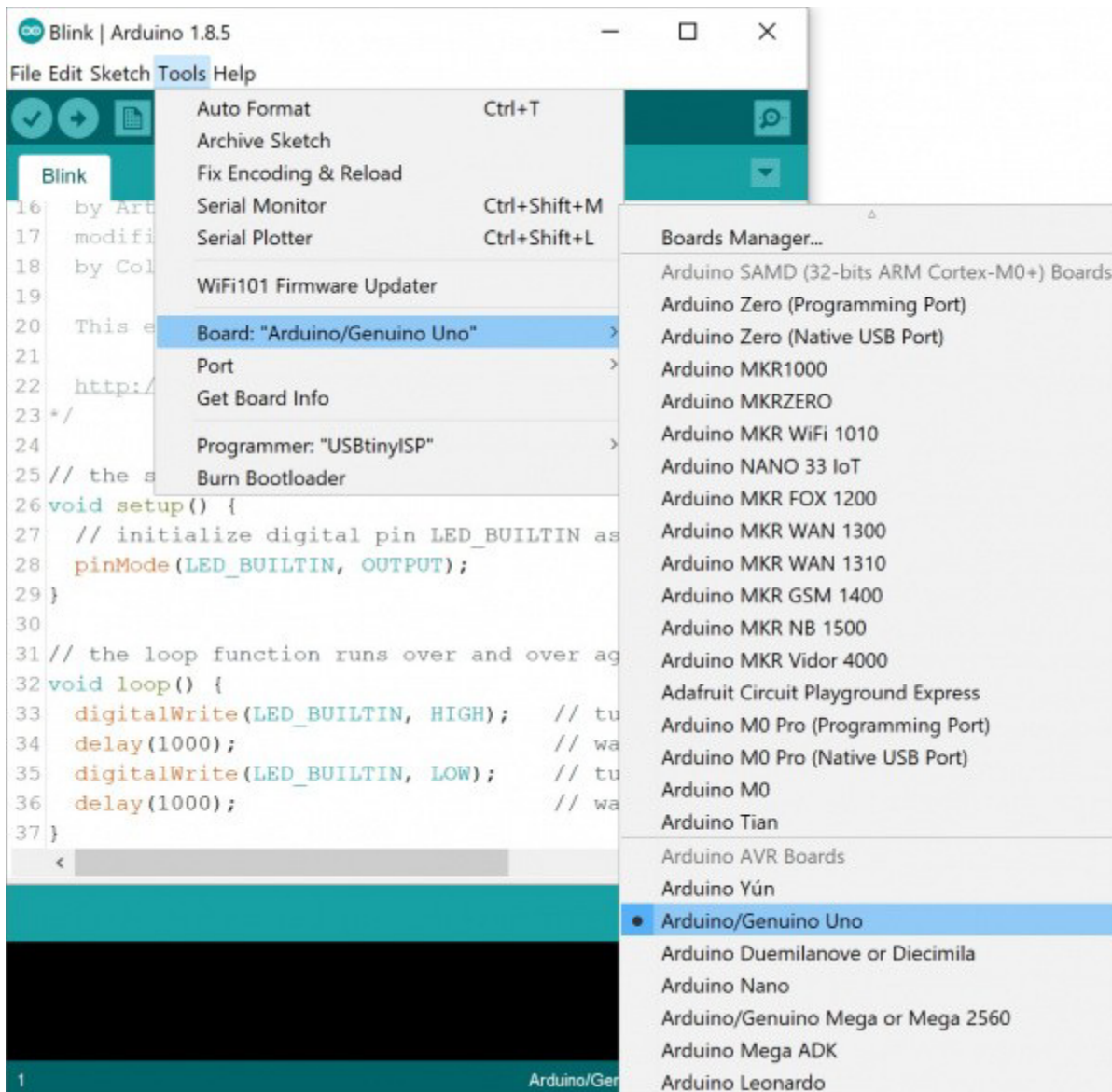


Note: Depending on your method of installing the Arduino IDE, the application may be on your desktop or the program folder.

Open the Blink example sketch by going to: **File > Examples > 01.Basics > Blink.**

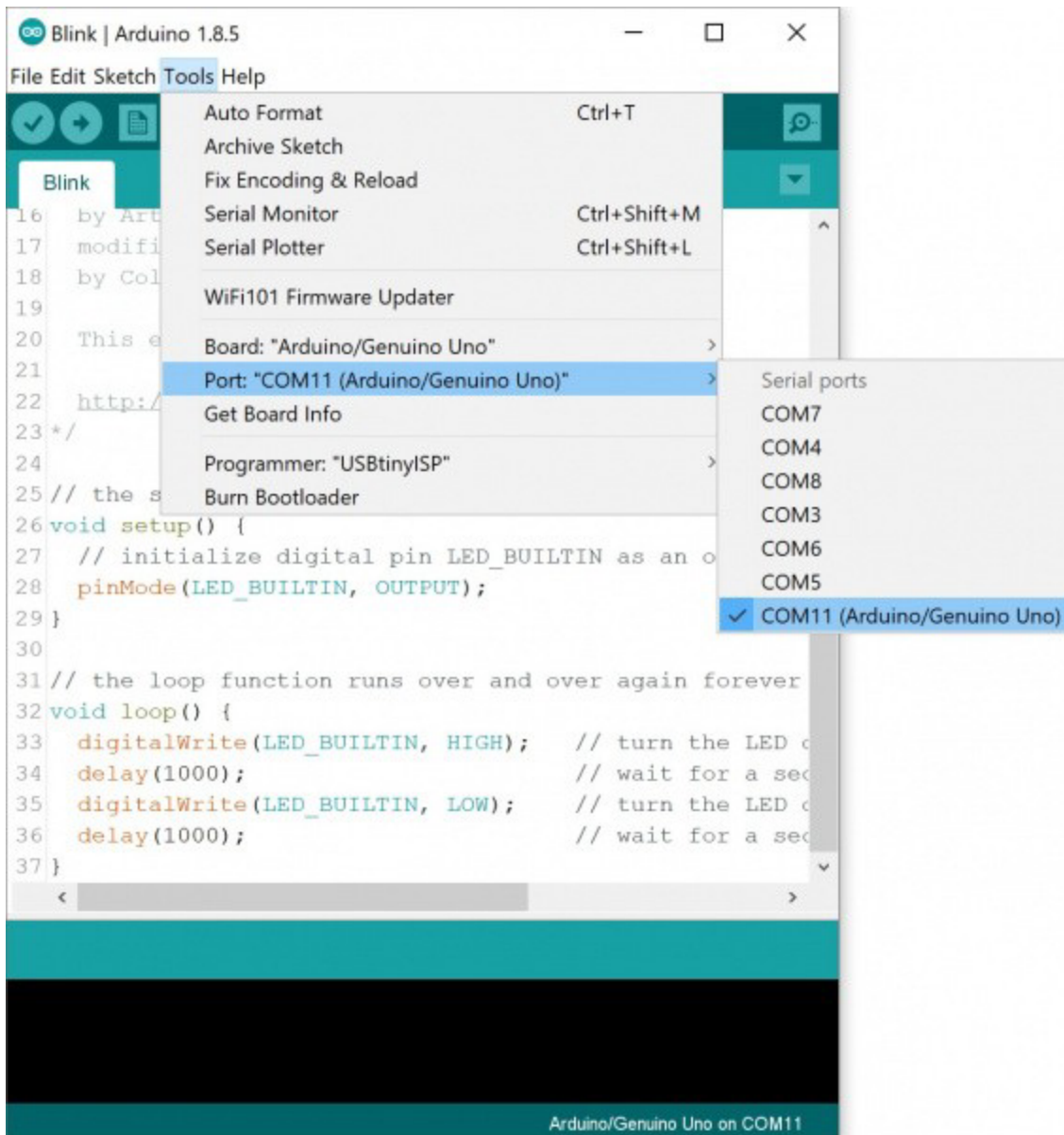


Select the type of Arduino board you're using: **Tools > Board > Arduino Uno.**



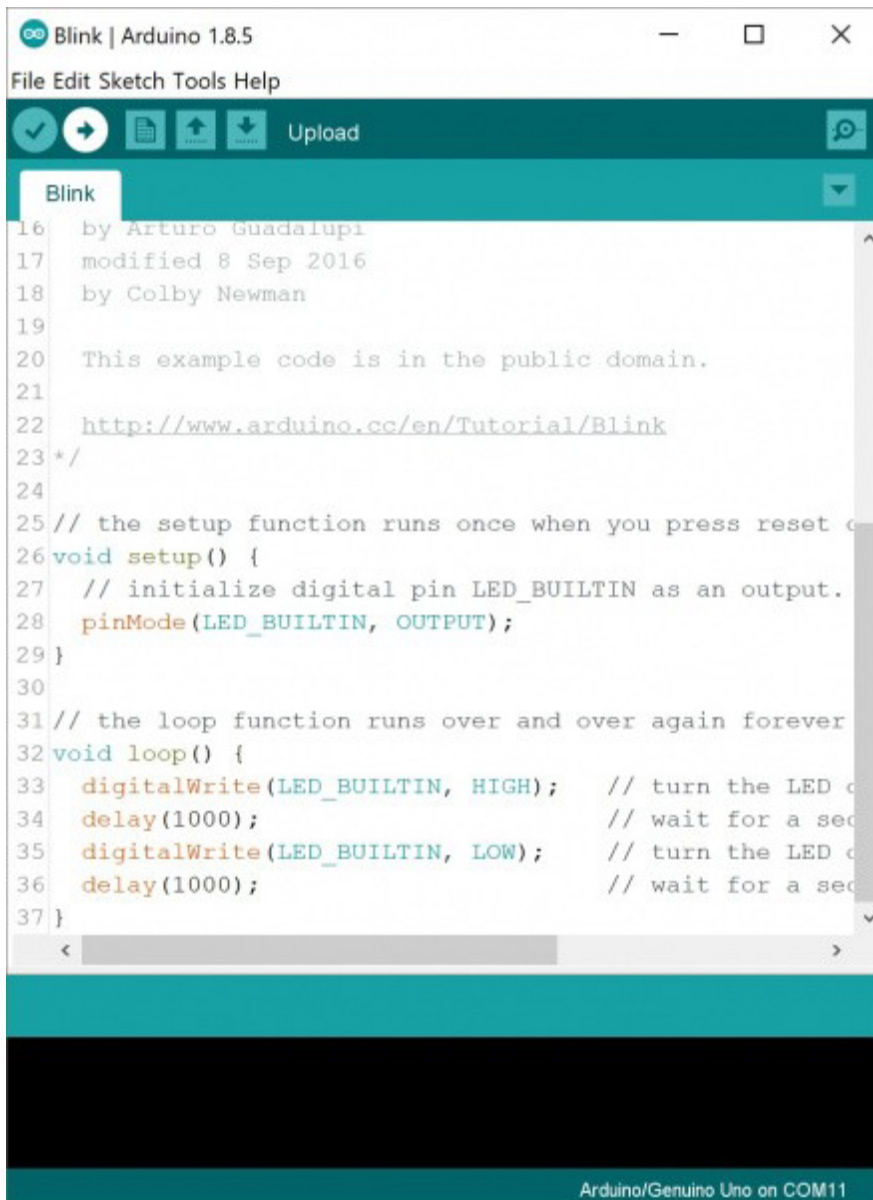
Note: As you move to other architectures, you may need to select a different board definition depending on your development board. For the Arduino Uno R3 and RedBoard development boards with ATmega328P, you can simply select **Arduino Uno**. Certain Arduino IDE versions may have you select **Arduino/ Genuino Uno**.

Select the serial/COM port that your Arduino is attached to: **Tools > Port > COMxx**. In this case it was *COM11*.

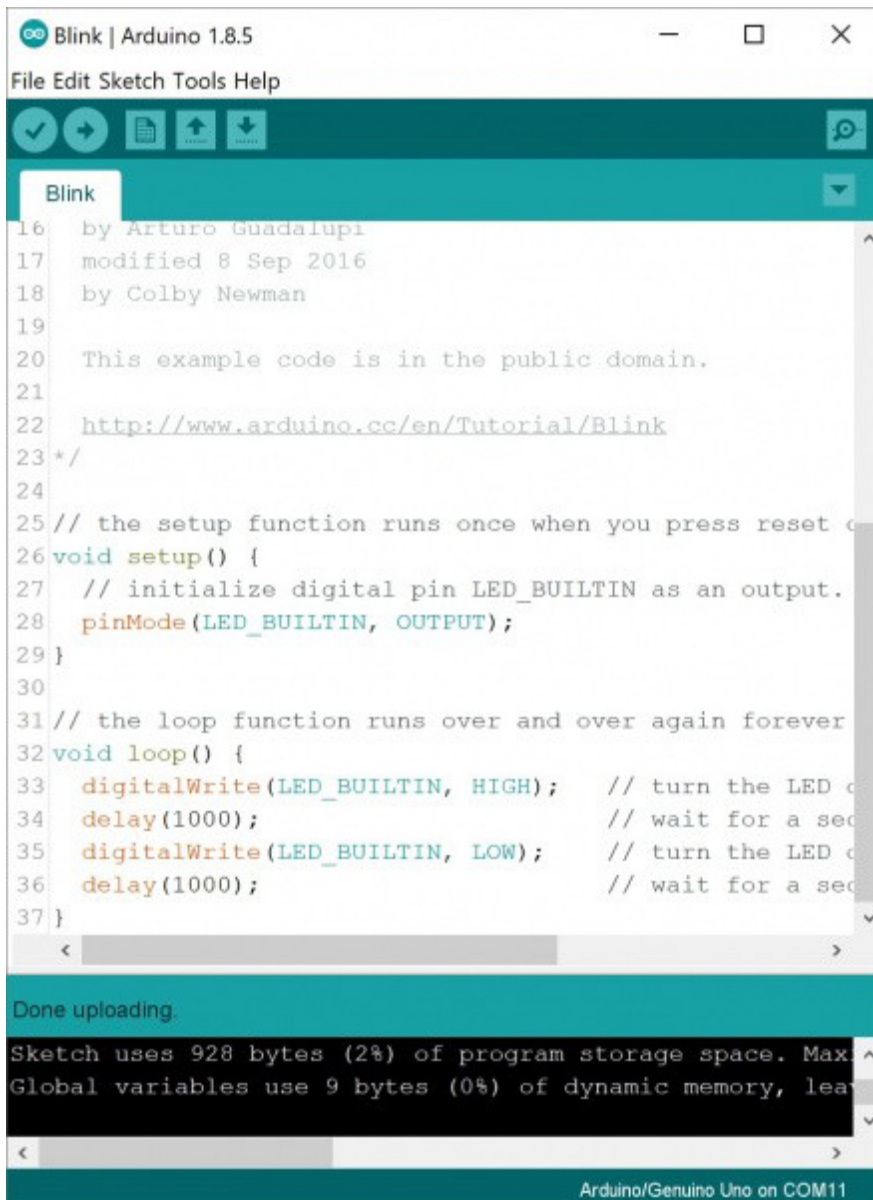


Note: If you're not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino.

With your Arduino board connected, and the Blink sketch open, press the "**Upload**" button.



After a second, you should see some LEDs flashing on your Arduino, followed by the message **"Done Uploading"** in the status bar of the Blink sketch.

The image is a screenshot of the Arduino IDE interface. At the top, the window title is "Blink | Arduino 1.8.5". Below the title bar is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". Underneath the menu bar is a toolbar with icons for checking, running, saving, uploading, and downloading. The main text area shows the "Blink" sketch. It includes a header comment with author information (Arturo Guadalupi, modified 8 Sep 2016, by Colby Newman) and a link to the Arduino tutorial. The code defines a setup function to initialize the built-in LED as an output and a loop function that turns the LED on and off with 1000ms delays. At the bottom, a status bar shows "Done uploading." and "Sketch uses 928 bytes (2%) of program storage space. Max Global variables use 9 bytes (0%) of dynamic memory, leaving 1024 bytes free." The bottom-most status bar indicates the board is "Arduino/Genuino Uno on COM11".

```
Blink
16  by Arturo Guadalupi
17  modified 8 Sep 2016
18  by Colby Newman
19
20  This example code is in the public domain.
21
22  http://www.arduino.cc/en/Tutorial/Blink
23  */
24
25  // the setup function runs once when you press reset
26  void setup() {
27    // initialize digital pin LED_BUILTIN as an output.
28    pinMode(LED_BUILTIN, OUTPUT);
29  }
30
31  // the loop function runs over and over again forever
32  void loop() {
33    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34    delay(1000);                      // wait for a second
35    digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
36    delay(1000);                      // wait for a second
37  }
```

Done uploading.

Sketch uses 928 bytes (2%) of program storage space. Max Global variables use 9 bytes (0%) of dynamic memory, leaving 1024 bytes free.

Arduino/Genuino Uno on COM11

If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

Note: Depending on the architecture and development board, the built-in LED may be defined on a different pin. You may need to adjust `LED_BUILTIN` or pin 13 to a different value before uploading.

Conclusion:

We have given an incredibly capable way to deal and to develop a smart system for vehicles to diminish number of disasters caused in light of alcoholic driving. As the creating insight among people is that vehicle security is dynamically critical. Future degree of this structure is to control the setbacks caused due to alcohol use. This system improves the security of individual and in this manner giving the convincing progression in the vehicle business regarding decrease setbacks caused in light of driving

References:

- [1] Mr. Godson Michael D'silva, Mr. Anoop Kunjumon Scariah, Mr. Lukose Roy Pannapara and Ms. Jessica John Joseph "Smart Ticketing System for Railways in Smart Cities using Software as a Service Architecture" International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (ISMAC 2017)
- [2] Md. Foisal Mahedi Hasan, *Golam Tangim, *Md. Kafiul Islam, *Md. Rezwanul Haque Khandokar, *** Arif Ullah "RFID-based Ticketing for Public Transport System" 978-1-4244-5540- 9/10/\$26.00 ©2016 IEEE
- [3] Kirti Dhiman¹, Er. CK Raina² "IoT Based Ticket Checking System" Vol. 6, Issue 3, March 2017
- [4] Vijaysanthi. R, Radha. N, Jaya Shree. M, Sindhuja. V "Fingerprint Authentication using Raspberry Pi based On IoT"
- [5] Dhvani. K. Shah, Dr. Vinayak A. Bharadi, V. J. Kaul, Sameer Amrutia "End-to-end Encryption based Biometric SaaS" International Journal of Pure and Applied Mathematics Special Issue
- [6] Syamala, M., Nalini, N.J., Ragupathy, R., Magulur, L.P., "Random forest classifier approach for blurred images", (2017) International Journal of Pure and Applied Mathematics, 116 (6 Special Issue), pp. 67-72.
- [7] Utkam, S.G., Parasa, R., Kondapaneni, A., Tulabandula, P.R.K., Kuppala, D.R., "A secured symmetric key encryption technique using images as secret keys", (2017) International Journal of Pure and Applied Mathematics, 116 (6 Special Issue), pp. 149-153.
- [8] Veeraiah, T., Mahamkali, A., Kishore, B.N.P., Rao, A.N., Rao, S.K.M., "Image denoising by profundity map method using diagram base change and group scanty", (2017) Mathematics, 115 (8 Special Issue), pp. 143-149.
- [9] Veeraiah, T., Rao, S.K.M., "Classification of fuzzy rule based order-statistic filters for color images", (2017) International Journal of Pure and Applied Mathematics, 115 (8 Special Issue), pp

