**SMART HELEMT USING IOT**

Abstract:

As we know India is second most populated country and has a large youth population, nowadays youth are fond of bikes and because of fashion, they neglect wearing helmet. Because of these, bike accidents are increasing day by day which causes deaths. Major deaths are due to head injuries which can be prevented by wearing a helmet. Drunk and drive cases are becoming more, which causes accidents and due to lack of negligence where an accident occurs and people are dying. These incidents made us develop a smart helmet using internet of things which reduce the accidents and risk of deaths, which has following features, the bike starts only if the rider wears a helmet if the rider is over drunken then the ignition will be automatically offed and if any accident occurs then through IOT modem it will send the message to the registered contact number by using a Telegram bot.

INTRODUCTION

Technology is the word where we hear every corner of the world, mainly in the fields of education, manufacturing of the products, transportation, communication and health. In the field of transportation industry was always an essential part of the economy, and a tool used by the government. We have different ways of transportation for moving around the world, but motorcycles are the craziest vehicle in the young generation and as well as to the world. Motorcycle safety related to different features of the vehicle such as equipment model, design of the vehicle and as well as operator skill is special for motorcycle rider has towards the motorbikes. But they are the most unsafe road users, without a protective body, even the slightest careless can have serious injuries or may lead to the death of the rider. Not only because of the careless, but the death of the people may occur due to over speed, rash driving, over consumption of alcohol and violation of traffic rules. But the main reason for brain damage and this leads to immediate death, was the absence of helmet on the person. If the rider wears the helmet, 80% chances for avoiding head injuries and we can save a life from accidents. With the help of new technologies such as IoT, dangerous traffic situations will not occur. And modelling the motorcycles with the sensors, alert system to the rider and surroundings by a sending message, and to make it mandatory for the bike rider to wear a helmet during his/her ride. In a recent survey, every hour 4 people die in road accidents, 70% due to not wearing a helmet.

Based on statistics from around the world, increasing safety regulations and by using the innovative technology, being developed to avoid such instances, and to ensure the safety of riders. The idea behind our project is to ensure the Safety on Two Wheels for a safe journey.

INTRODUCTION TO EMBEDDED SYSTEMS

Each day, our lives become more dependent on 'embedded systems', digital information technology that is embedded in our environment. More than 98% of processors applied today are in embedded systems, and are no longer visible to the customer as 'computers' in the ordinary sense. An Embedded System is a special-purpose system in which the computer is completely encapsulated by or dedicated to the device or system it controls. Unlike a general-purpose computer, such as a personal computer, an embedded system performs one or a few pre-defined tasks, usually with very specific requirements. Since the system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product. Embedded systems are often mass-produced, benefiting from economies of scale. The increasing use of PC hardware is one of the most important developments in high-end embedded systems in recent years. Hardware costs of high-end systems have dropped dramatically as a result of this trend, making feasible some projects which previously would not have been done because of the high cost of non-PC-based embedded hardware. But software choices for the embedded PC platform are not nearly as attractive as the hardware.

Typically, an embedded system is housed on a single [microprocessor](http://www.webopedia.com/TERM/E/microprocessor.html)[board](http://www.webopedia.com/TERM/E/board.html) with the [programs](http://www.webopedia.com/TERM/E/program.html) stored in [ROM](http://www.webopedia.com/TERM/E/ROM.html). Virtually all appliances that have a [digital interface](http://www.webopedia.com/TERM/E/digital.html) -- watches, microwaves, VCRs, cars -- utilize embedded systems. Some embedded systems include an [operating system](http://www.webopedia.com/TERM/E/operating_system.html), but many are so specialized that the entire logic can be implemented as a single program.

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.

In terms of complexity embedded systems can range from very simple with a single microcontroller chip, to very complex with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

Definition of an Embedded System

Embedded system is defined as, for a particular/specific application implementing the software code to interact directly with that particular hardware what we built. Software is used for providing features and flexibility,Hardware = {Processors, ASICs, Memory,etc} is used for Performance (& sometimes security)

There are many definitions of embedded system but all of these can be combined into a single concept. An embedded system is a special purpose computer system that is used for particular task.

Features of Embedded Systems

The versatility of the embedded computer system lends itself to utility in all kinds of enterprises, from the simplification of deliverable products to a reduction in costs in their development and manufacture. Complex systems with rich functionality employ special operating systems that take into account major characteristics of embedded systems. Embedded operating systems have minimized footprint and may follow real-time operating system specifics.

The special computers system is usually less powerful than general-purpose systems, although some expectations do exist where embedded systems are very powerful and complicated. Usually a low power consumption CPU with a limited amount of memory is used in embedded systems. Many embedded systems use very small operating systems; most of these provide very limited operating system capabilities.

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](http://www.answers.com/topic/economies-of-scale-2).

Some embedded systems have to operate in extreme environment conditions such as very high temperature & humidity.

For high volume systems such as portable music players or mobile phones, minimizing cost is usually the primary design consideration. Engineers typically select hardware that is just “good enough” to implement the necessary functions.

For low volume or prototype embedded systems, general purpose computers may be adapted by limiting the programs or by replacing the operating system with a real-time operating system.

Characteristics of Embedded Systems

Embedded computing systems generally exhibit rich functionality—complex functionality is usually the reason for introducing CPUs into the design. However, they also exhibit many non-functional requirements that make the task especially challenging:

• Real-time deadlines that will cause system failure if not met;

• Multi-rate operation;

• In many cases, low power consumption;

• Low manufacturing cost, which often means limited code size.

Workstation programmers often concentrate on functionality. They may consider the performance characteristics of a few computational kernels of their software, but rarely analyze the total application. They almost never consider power consumption and manufacturing cost. The need to juggle all these requirements makes embedded system programming very challenging and is the reason why embedded system designers need to understand computer architecture.

Overview of an Embedded System Architecture

Every Embedded system consists of a custom-built hardware built around a central processing unit. This hardware also contains memory chips onto which the software is loaded.

The operating system runs above the hardware and the application software runs above the operating system. The same architecture is applicable to any computer including desktop computer. However these are significant differences. It is not compulsory to have an operating system in every embedded system. For small applications such as remote control units, air conditioners, toys etc.

Applications of Embedded Systems

Some of the most common embedded systems used in everyday life are

Small embedded controllers: 8-bit CPUs dominate, simple or no operating system

(e.g., thermostats)

Control systems: Often use DSP chip for control computations

(e.g., automotive engine control)

Distributed embedded control: Mixture of large and small nodes on a real-time

Embedded networks

(e.g., cars, elevators, factory automation)

System on chip: ASIC design tailored to application area

(e.g., consumer electronics, set-top boxes)

Network equipment: Emphasis on data movement/packet flow

(e.g., network switches; telephone switches)

Critical systems: Safety and mission critical computing

(e.g., pacemakers, automatic trains)

Signal processing: Often use DSP chips for vision, audio, or other signal

Processing (e.g., face recognition)

Robotics: Uses various types of embedded computing (especially

Vision and control) (e.g., autonomous vehicles)

Computer peripherals: Disk drives, keyboards, laser printers, etc.

Wireless systems: Wireless network-connected “sensor networks” and

“Motes” to gather and report information

Embedded PCs: Palmtop and small form factor PCs embedded into

Equipment

Command and control: Often huge military systems and “systems of systems”

(e.g., a fleet of warships with interconnected

Computers)

Home Appliances, intercom, telephones, security systems,garage door openers, answering machines, faxmachines, home computers, TVs, cable TV tuner,VCR, camcorder, remote controls, video games,cellular phones, musical instruments, sewingmachines, lighting control, paging, camera, pinballmachines, toys, exercise equipment

Office Telephones, computers, security systems, faxmachines, microwave, copier, laser printer, colorprinter, paging

Auto Trip computer, engine control, air bag, ABS,instrumentation, security system, transmissioncontrol, entertainment, climate control, cellularphone, keyless entry

TYPES OF EMBEDDED SYSTEMS

Based on functionality and performance embedded systems categorized as 4 types

1. Stand alone embedded systems

2. Real time embedded systems

3. Networked information appliances

4. Mobile devices

1. Stand alone embedded systems:-

As the name implies, stand alone systems work in stand alone mode. They take i/p, process them and produce the desire o/p. The i/p can be an electrical signal from transducer or temperature signal or commands from human being. The o/p can be electrical signal to drive another system an led or LCD display

ex digital camera, microwave oven, CD player, Air conditioner etc

1. Real time embedded systems:-

In this type of an embedded system a specific work has to be complete in a particular period of time.

Hard Real time systems: - embedded real time used in missiles

Soft Real time systems: - DVD players

3. Networked information appliances:-

Embedded systems that are provided with n/w interfaces and accessed by n/w such as local area n/w or internet are called Network Information Appliances

Ex A web camera is connected to the internet. Camera can send pictures in real time to any computers connected to the internet

4. Mobile devices:-

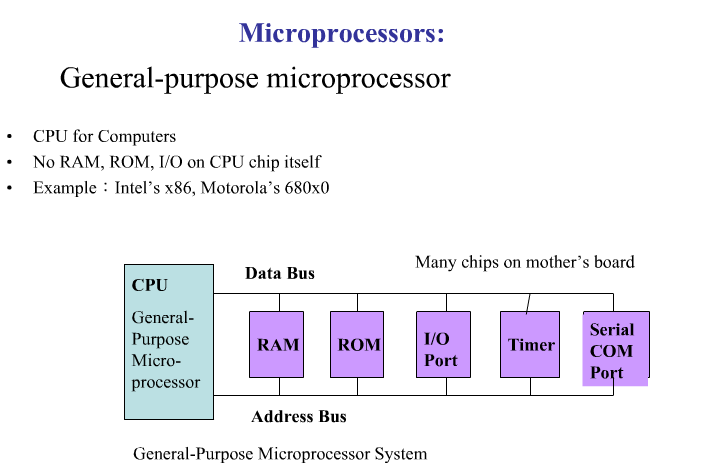
Actually it is a combination of both VLSI and Embedded System

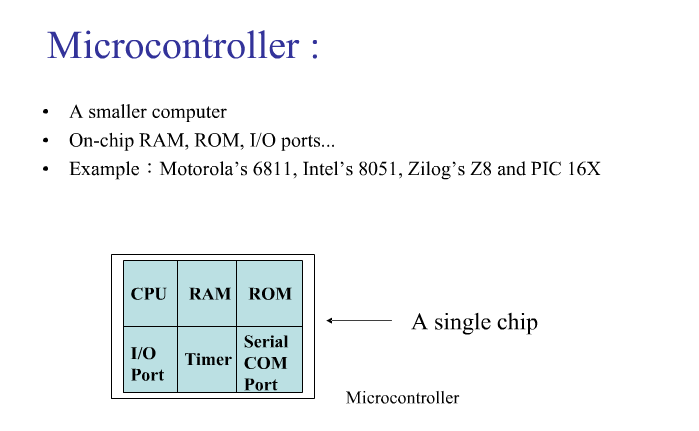
Mobile devices such as Mobile phone, Personal digital assistants, smart phones etc are special category of embedded system.

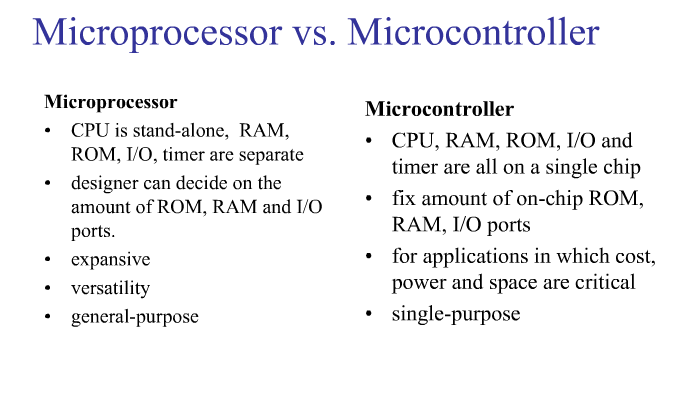
MICROCONTROLLER VERSUS MICROPROCESSOR

A system designer using a general-purpose microprocessor such as the Pentium or the 68040 must add RAM, ROM, I/O ports, and timers externally to make them functional. Although the addition of external RAM, ROM, and I/O ports makes these systems bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand.

A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, the RAM, ROM, I/O ports and the timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O ports, or timer to it. The fixed amount of on-chip ROM, RAM, and number of I/O ports in Microcontrollers makes them ideal for many applications in which cost and space are critical.







### CPU platform:

Embedded processors can be broken into two distinct categories: microprocessors (μP) and microcontrollers (μC). Microcontrollers have built-in peripherals on the chip, reducing size of the system.

There are many different CPU architectures used in embedded designs such as ARM, MIPS, Coldfire/68k, PowerPC, x86, PIC, 8051, Atmel AVR, Renesas H8, SH, V850, FR-V, M32R, Z80, Z8, etc. This in contrast to the desktop computer market, which is currently limited to just a few competing architectures.

PC/104 and PC/104+ are a typical base for small, low-volume embedded and ruggedized system design. These often use DOS, Linux, NetBSD, or an embedded real-time operating system such as QNX or VxWorks.

A common configuration for very-high-volume embedded systems is the system on a chip (SoC), an application-specific integrated circuit (ASIC), for which the CPU core was purchased and added as part of the chip design. A related scheme is to use a field-programmable gate array (FPGA), and program it with all the logic, including the CPU.

Embedded systems are based on the concept of the microcontroller, a single integrated circuit that contains all the technology required to run an application. Microcontrollers make integrated systems possible by combining several features together into what is effectively a complete computer on a chip, including:   
\* Central Processing Unit   
\* Input/Output interfaces (such as serial ports)   
\* Peripherals (such as timers)   
\* ROM, EEPROM or Flash memory for program storage   
\* RAM for data storage   
\* Clock generator   
  
By integrating all of these features into a single chip it is possible to greatly reduce the number of chips and wiring necessary to control an electronic device, dramatically reducing its complexity, size and cost.

\* Size & Weight: Microcontrollers are designed to deliver maximum performance for minimum size and weight. A centralized on-board computer system would greatly outweigh a collection of microcontrollers.   
\* Efficiency: Microcontrollers are designed to perform repeated functions for long periods of time without failing or requiring service.

Based on the Processor side Embedded Systems is mainly divided into 3 types

1. Micro Processor: - are for general purpose eg: our personal computer

2. Micro Controller: - are for specific applications, because of cheaper cost we will go for these

3. DSP (Digital Signal Processor):- are for high and sensitive application purpose

By integrating all of these features into a single chip it is possible to greatly reduce the number of chips and wiring necessary to control an electronic device, dramatically reducing its complexity, size and cost.

\* Size & Weight: Microcontrollers are designed to deliver maximum performance for minimum size and weight. A centralized on-board computer system would greatly outweigh a collection of microcontrollers.   
\* Efficiency: Microcontrollers are designed to perform repeated functions for long periods of time without failing or requiring service.

Based on the Processor side Embedded Systems is mainly divided into 3 types

1. Micro Processor: - are for general purpose eg: our personal computer

2. Micro Controller: - are for specific applications, because of cheaper cost we will go for these

3. DSP (Digital Signal Processor):- are for high and sensitive application purpose

Proposed System

Block Diagram:

Sw

LCD

IR/LDR

ARDUINO UNO

Alcohol

GPS

Buzzer

Mems

RF tx

IOT

Motor

RF rx

HARDWARE DESCRIPTION

Arduino

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

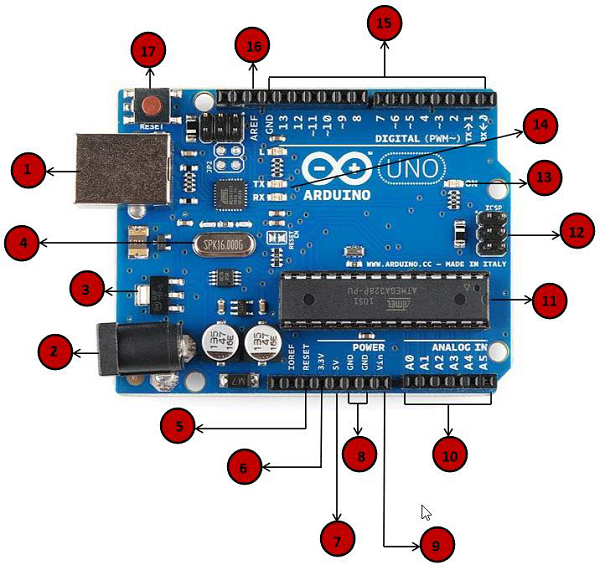
* Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
* You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
* Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
* Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
* Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

Board Types

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.



|  |  |
| --- | --- |
| Power USB | Power USB  Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1). |
| Barrel Jack | Power (Barrel Jack)  Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2). |
| Voltage Regulator | Voltage Regulator  The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements. |
| Crystal Oscillator | Crystal Oscillator  The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz. |
| Arduino Reset | Arduino Reset  You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5). |
| Pins | Pins (3.3, 5, GND, Vin)   * 3.3V (6) − Supply 3.3 output volt * 5V (7) − Supply 5 output volt * Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. * GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. * Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply. |
| Analog pins | Analog pins  The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor. |
| Main microcontroller | Main microcontroller  Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet. |
| ICSP pin | ICSP pin  Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus. |
| Power LED indicator | Power LED indicator  This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection. |
| TX and RX LEDs | TX and RX LEDs  On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process. |
| Digital I/O | Digital I/O  The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM. |
| AREF | AREF  AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins. |

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 − First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

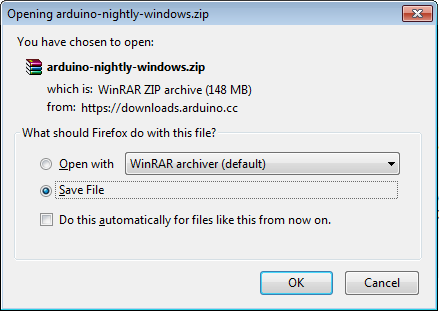


In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image.



Step 2 − Download Arduino IDE Software.

You can get different versions of Arduino IDE from the [Download page](https://www.arduino.cc/en/Main/Software) on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.



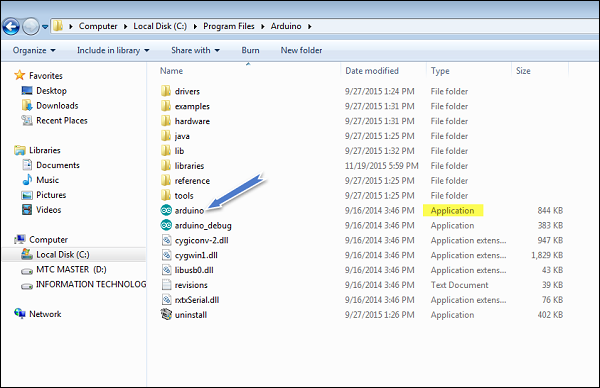
Step 3 − Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

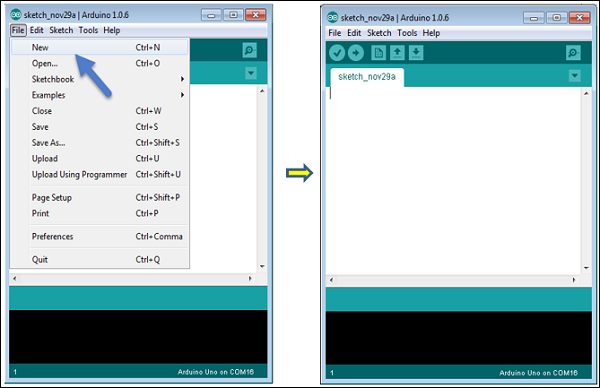


Step 5 − Open your first project.

Once the software starts, you have two options −

* Create a new project.
* Open an existing project example.

To create a new project, select File → New.



To open an existing project example, select File → Example → Basics → Blink.

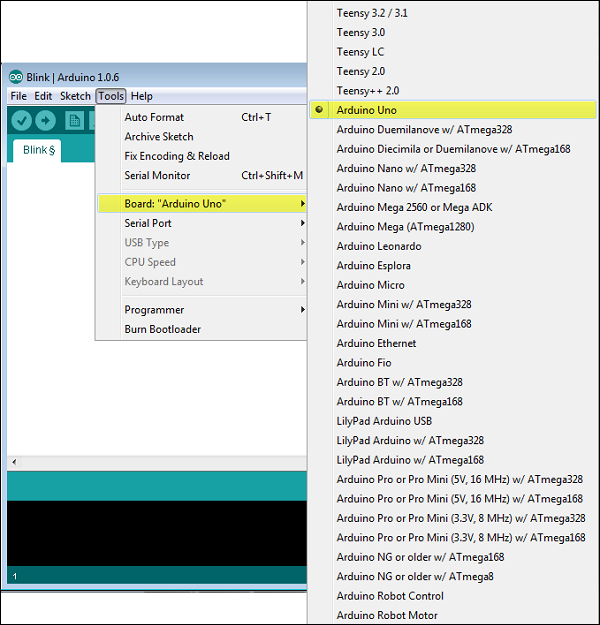
Open Project

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6 − Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

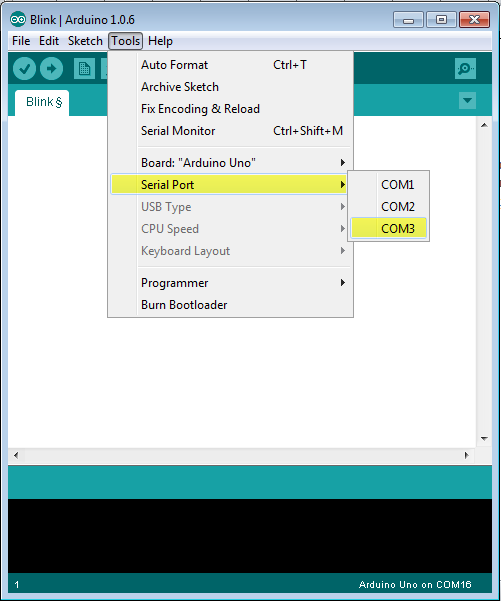
Go to Tools → Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using.

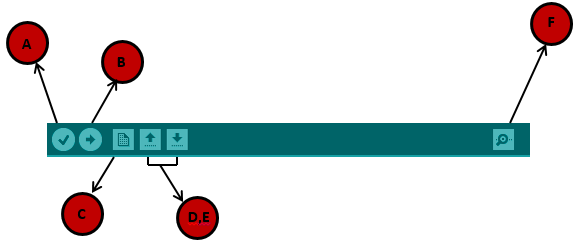
Step 7 − Select your serial port.

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 8 − Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A − Used to check if there is any compilation error.

B − Used to upload a program to the Arduino board.

C − Shortcut used to create a new sketch.

D − Used to directly open one of the example sketch.

E − Used to save your sketch.

F − Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

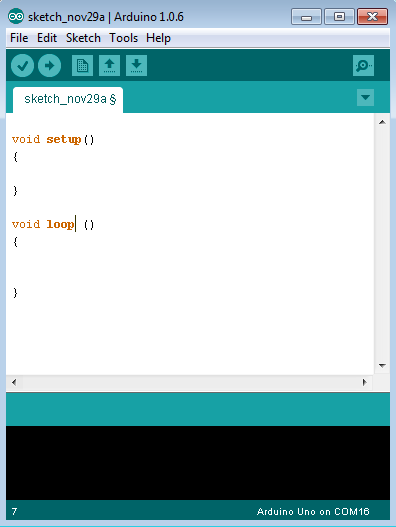
Sketch − The first new terminology is the Arduino program called “sketch”.

## Structure

Arduino programs can be divided in three main parts: Structure, Values (variables and constants), and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the Structure. Software structure consist of two main functions −

* Setup( ) function
* Loop( ) function



Void setup ( ) {

}

* PURPOSE − The setup() function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.
* INPUT − -
* OUTPUT − -
* RETURN − -

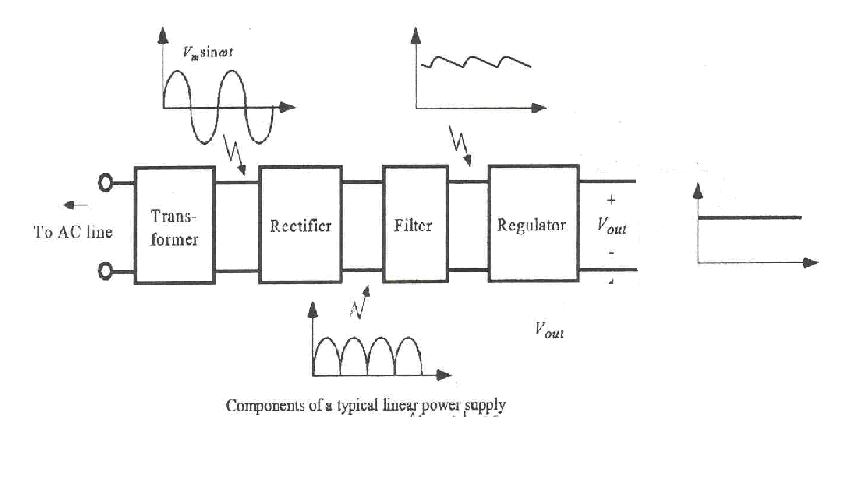
Void Loop () {

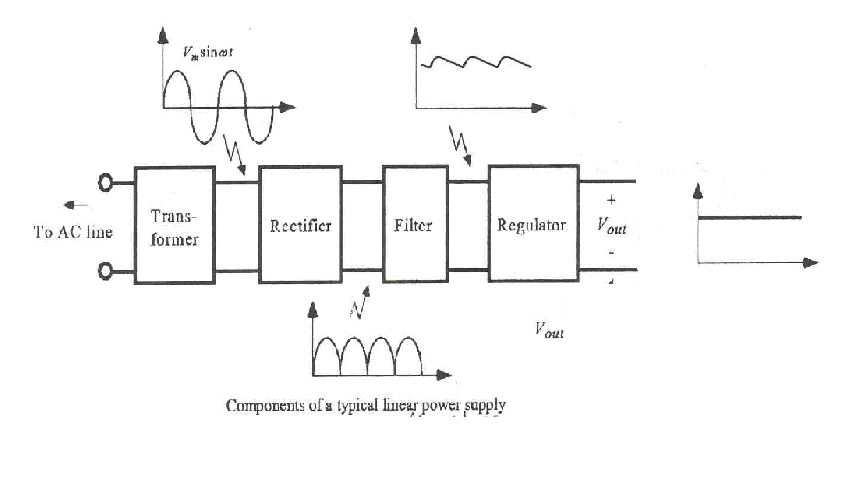
}

* PURPOSE − After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.
* INPUT − -
* OUTPUT − -
* RETURN − -

POWER SUPPLY MODULES:

* STEP DOWN TRANSFORMER
* BRIDGE RECTIFIER WITH FILTER
* VOLTAGE REGULATORS



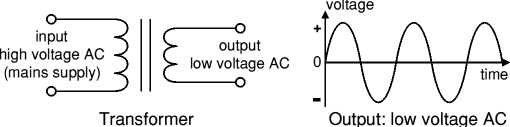


### Transformer

Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC. Step-up transformers increase voltage, step-down transformers reduce voltage.

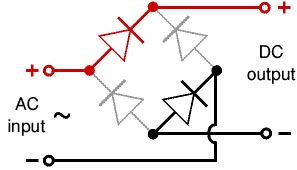
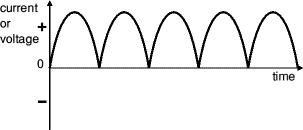
A step down power transformer is used to step down the AC voltage from the [line voltagegrey_loader](http://www.electronics-project-design.com/PowerSupplies.html" \t "_top)

of 110 VAC or 220 VAC i.e, it converts higher voltage at the input side to a lower voltage at the output.



### Rectifier

There are several ways of connecting diodes to make a rectifier to convert AC to DC. The [bridge rectifier](http://www.kpsec.freeuk.com/powersup.htm#bridgerectifier) is the most important and it produces full-wave varying DC



Bridge rectifier Output: full-wave varying DC  
Alternate pairs of diodes conduct, changing over (using all the AC wave)  
the connections so the alternating directions of  
AC are converted to the one direction of DC.

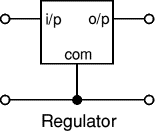
### Filter

### Filtering is performed by a large value [electrolytic capacitor](http://www.kpsec.freeuk.com/components/capac.htm#polarised) connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The diagram shows the unfiltered varying DC (dotted line) and the filtered DC (solid line). The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output.

### Typically 1000 μf capacitor is used in microcontroller of 8051.

### Regulator

This is a simple DC regulated supply project using 7805 voltage regulator to obtain a variable DC voltage range from 5V to 15V



Pin out of the 7805 regulator IC.

1. Unregulated voltage in

2. Ground

3. Regulated voltage out

If you need other voltages than +5V, you can modify the circuit by replacing the 7805 chips with another regulator with different output voltage from regulator 78xx chip family. The last numbers in the the chip code tells the output voltage. Remember that the input voltage must be at least 3V greater than regulator output voltage ot otherwise the regulator does not work well.

Light Emitting Diodes

Light Emitting Diodes or LEDs, are among the most widely used of all the types of diodes available. They are the most visible type of diode that emits a fairly narrow bandwidth of either visible colored light, invisible infra-red or laser type light when a forward current is passed through them. A "Light Emitting Diode" or LED as it is more commonly called, is basically just a specialized type of PN-junction diode, made from a very thin layer of fairly heavily doped semiconductor material. When the diode is Forward Biased, electrons from the semiconductors conduction band combine with holes from the valence band, releasing sufficient energy to produce photons of light. Because of this thin layer a reasonable number of these photons can leave the junction and radiate away producing a coloured light output.

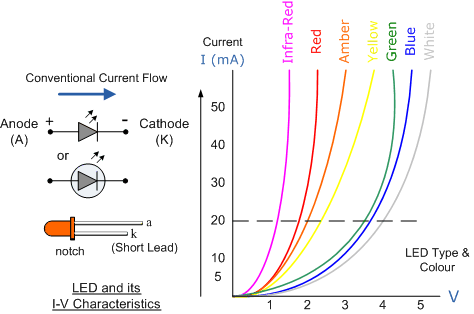
Unlike normal diodes which are made for detection or power rectification, and which are generally made from either Germanium or Silicon semiconductor material, Light Emitting Diodes are made from compound type semiconductor materials such as Gallium Arsenide (GaAs), Gallium Phosphide (GaP), Gallium Arsenide Phosphide (GaAsP), Silicon Carbide (SiC) or Gallium Indium Nitride (GaInN). The exact choice of the semiconductor material used will determine the overall wavelength of the photon light emissions and therefore the resulting colour of the light emitted, as in the case of the visible light coloured LEDs, (RED, AMBER, GREEN etc).

|  |  |  |  |
| --- | --- | --- | --- |
| Typical LED Characteristics | | | |
| Semiconductor Material | Wavelength | Colour | VF @ 20mA |
| ++++GaAs | 850-940nm | Infra-Red | 1.2v |
| GaAsP | 630-660nm | Red | 1.8v |
| GaAsP | 605-620nm | Amber | 2.0v |
| GaAsP:N | 585-595nm | Yellow | 2.2v |
| GaP | 550-570nm | Green | 3.5v |
| SiC | 430-505nm | Blue | 3.6v |
| GaInN | 450nm | White | 4.0v |

From the table above we can see that the main P-type dopant used in the manufacture of Light Emitting Diodes is Gallium (Ga, atomic number 31) and the main N-type dopant used is Arsenic (As, atomic number 31) giving the resulting Gallium Arsenide (GaAs) crystal structure, which has the characteristics of radiating significant amounts of infrared radiation from its junction when a forward current is flowing through it. By also adding Phosphorus (P, atomic number 15), as a third dopant the overall wavelength of the emitted radiation is reduced to give visible red light to the human eye. Further refinements in the doping process of the PN-junction have resulted in a range of colours available from red, orange and amber through to yellow, and the recently developed blue LED which is achieved by injecting nitrogen atoms into the crystal structure during the doping process.

Light Emitting Diodes I-V Characteristics:

Before a light emitting diode can "emit" any form of light it needs a current to flow through it, as it is a current dependant device. As the LED is to be connected in a forward bias condition across a power supply it should be Current Limited using a series resistor to protect it from excessive current flow. From the table above we can see that each LED has its own forward voltage drop across the PN-junction and this parameter which is determined by the semiconductor material used is the forward voltage drop for a given amount of forward conduction current, typically for a forward current of 20mA. In most cases LEDs are operated from a low voltage DC supply, with a series resistor to limit the forward current to a suitable value from say 5mA for a simple LED indicator to 30mA or more where a high brightness light output is needed.

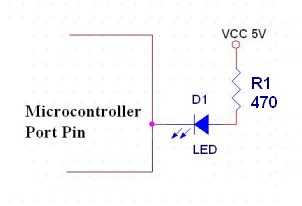


LED Series Resistance:

The series resistor value RS is calculated by simply using Ohm´s Law, knowing the required forward current IF, the supply voltage VS and the expected forward voltage drop of the LED, VF at this current level as shown below:

|  |
| --- |
| Light Emitting Diode Circuit |

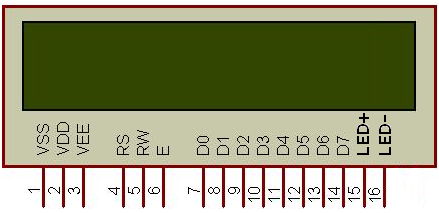
LED Typical Applications:The following figure shows how to interface the LED to microcontroller. As you can see the Anode is connected through a resistor to Vcc and the Cathode is connected to the Microcontroller pin. So when the Port Pin is HIGH the LED is OFF and when the Port Pin is LOW the LED is turned ON.



Alphanumeric LCD

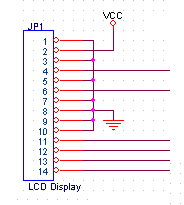
Liquid Crystal Display also called as LCD is very helpful in providing user interface as well as for debugging purpose. The most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580. The most commonly used LCDs found in the market today 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.

Pin Description



|  |  |  |
| --- | --- | --- |
| **Pin No.** | **Name** | **Description** |
| 1 | **VSS** | Power supply (GND) |
| 2 | **VCC** | Power supply (+5V) |
| 3 | **VEE** | Contrast adjusts |
| 4 | **RS** | 0 = Instruction input 1 = Data input |
| 5 | **R/W** | 0 = Write to LCD module 1 = Read from LCD module |
| 6 | **EN** | Enable signal |
| 7 | **D0** | Data bus line 0 (LSB) |
| 8 | **D1** | Data bus line 1 |
| 9 | **D2** | Data bus line 2 |
| 10 | **D3** | Data bus line 3 |
| 11 | **D4** | Data bus line 4 |
| 12 | **D5** | Data bus line 5 |
| 13 | **D6** | Data bus line 6 |
| 14 | **D7** | Data bus line 7 (MSB) |
| 15 | **LED+** | Back Light VCC |
| 16 | **LED-** | Back Light GND |

LCD INTERFAC WITH MICROCONTROLLER



**PORTPINS**

Microcontroller

INTERFACING LCD TO MICROCONTROLLER

The LCD is generally interfaced in 8-bit mode or 4-bit mode. in this project LCD is connected in 4-bit mode the interface connections of LCD with microcontroller are as follows

RS of LCD is connected to p0.0 of microcontroller

EN of LCD is connected to p0.1 of microcontroller

D4 of LCD is connected to p0.4 of microcontroller

D5 of LCD is connected to p0.5 of microcontroller

D6 of LCD is connected to p0.6 of microcontroller

D7 of LCD is connected to p0.7 of microcontroller

In 8-bit mode, the complete ASCII code is sent at once along with the control signals. But in 4-bit mode, the data is divided into two parts, i.e. MSB & LSB, and are called upper nibble & lower nibble.

The control signals are RS, R/W & E. RS is used to select the internal registers i.e. data register & command register. R/W is used to set the mode of LCD to read mode or write mode. E is used as chip select and is used to push the data internally to the corresponding registers.

To transfer the data/command in 8-bit mode, the data is written to the 8-bit data bus after selecting the required register and setting the mode to write mode. The E signal pin is then given a high to low signal to transfer the data.

To transfer the data/command in 4-bit mode, the higher nibble is first written to the MSB of the data port and the E is given a high to low signal. After a little delay or when the LCD is not busy,

# Relay

A relay is an [electrically](http://en.wikipedia.org/wiki/Electric) operated [switch](http://en.wikipedia.org/wiki/Switch). Many relays use an electromagnet to operate a switching mechanism, but other operating principles are also used. Relays find applications where it is necessary to control a circuit by a low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays found extensive use in telephone exchanges and early computers to perform logical operations. A type of relay that can handle the high power required to directly drive an electric motor is called a [contactor](http://en.wikipedia.org/wiki/Contactor). Solid-state relays control power circuits with no moving parts, instead using a semiconductor device triggered by light to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protection relays".

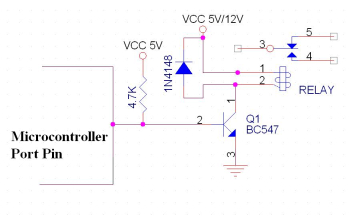
### Solid-state relay

A [solid state relay](http://en.wikipedia.org/wiki/Solid_state_relay) (SSR) is a [solid state](http://en.wikipedia.org/wiki/Solid_state_(electronics)) electronic component that provides a similar function to an [electromechanical](http://en.wikipedia.org/wiki/Electromechanical) relay but does not have any moving components, increasing long-term reliability. With early SSR's, the tradeoff came from the fact that every transistor has a small voltage drop across it. This voltage drop limited the amount of current a given SSR could handle. As transistors improved, higher current SSR's, able to handle 100 to 1,200 [Amperes](http://en.wikipedia.org/wiki/Ampere), have become commercially available. Compared to electromagnetic relays, they may be falsely triggered by transients.



Relay interfacing with 8051

Relays are devices which allow low power circuits to switch a relatively high Current/Voltage ON/OFF. For a relay to operate a suitable pull-in & holding current should be passed through its coil. Generally relay coils are designed to operate from a particular voltage often its 5V or 12V. **The function of relay driver circuit is to provide the necessary current (typically 25 to 70ma) to energize the relay coil**.



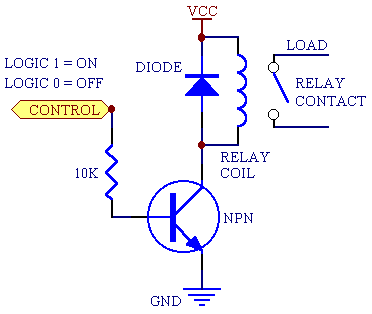
Figure

Figure 1 shows the basic relay driver circuit. As you can see an **NPN transistor BC547** is being used to control the relay. The transistor is driven into saturation (turned ON)  when a LOGIC 1 is written on the PORT PIN thus turning ON the relay. The relay is turned OFF by writing LOGIC 0 on the port pin.

A diode (1N4007/1N4148) is connected across the relay coil, this is done so as to protect the transistor from damage due to the **BACK** **EMF** generated in the relay's inductive coil when the transistor is turned OFF. When the transistor is switched OFF the energy stored in the inductor is dissipated through the diode & the internal resistance of the relay coil. As you can see we have used a pull up resistor at the base of the transistor. AT8951/52/55 has an internal pull up resistor of 10k so when the pin is pulled high the current flows through this resistor so the maximum output current is 5v/10K = 0.5ma, the DC current gain of BC547 is 100 so the maximum collector current we can get is 0.5ma x 100 = 50ma, but most of the relays require more than 70ma-130ma current depending on the relay that we have used, 0.5ma of base current is not suitable enough for turning ON the relay, so we have used an external pull up resistor. When the controller pin is high current flows through the controller pin i.e. 5v/10k=0.5ma as well as through the pull up resistor. We have used 4.7k pull up resistor so 5v/4.7k=1.1ma so maximum base current can be 0.5ma + 1.1ma=1.6ma i.e. collector current =1.6ma x 100 = 160ma which is enough to turn ON most of the relays.

The right relay depends on what you want to switch and how fast. What voltage, current, etc...

You won't be able to drive it directly from the Arduino pin, as most general purpose relays require at least 150mW to switch which is >30mA @ 5V. You will need to use something like this:

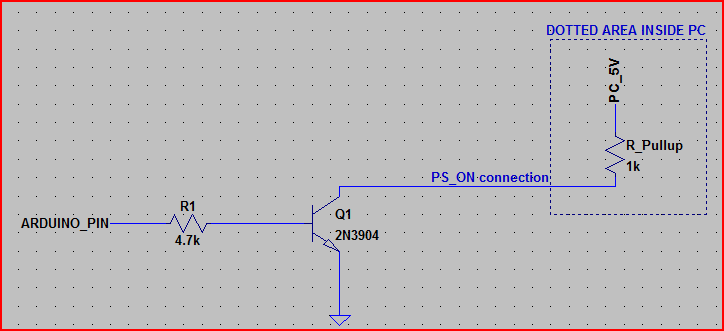


The NPN can be just about any general purpose NPN (2N2222, BC337, etc) and the diode can be most general purpose diodes (1N4001 or similar) VCC is your +5V.

If you go to somewhere like Farnell, and use the parametric search to narrow down you options, you will get hundreds of choices, here is an [example search](http://uk.farnell.com/jsp/search/browse.jsp?N=202502%20110201776%20110167107%20110163005%20110130931%20110231475%20110152465%20110221806%20110193822%20110167020%20110244683%20110156559%20110201355%20110167275%20110202659%20110177917%20110145968%20110145919%20110197892%20110186059%20110187956%20110365278%20110135015%20110145910%20110135123%20110145899%20110145909%20110135263%20110145903%20110199540%20110173668%20110124580%20110280202%20110268411%20110146139%20110124527&Ns=P_PRICE_FARNELL_UK%7C0&locale=en_UK&appliedparametrics=true&getResults=true&suppressRedirect=true&isRedirect=&originalQueryURL=/jsp/search/browse.jsp?N=202502&No=0&getResults=true&appliedparametrics=true&locale=en_UK&divisionLocale=en_UK&catalogId=&skipManufacturer=false&skipParametricAttributeId=&prevNValues=202502) with 5VDC general purpose relays capable of >10A and >250VAC selected.

EDIT

It seems this is to turn an ATX supply on by pulling the PC\_ON (usually green) connection to ground. In this case the relay is a bit overkill, and a simple open collector NPN transistor circuit can be used:



The dotted area is inside the PC, so all you need is the NPN transistor (almost any general purpose will do) and the resistor (4.7kOhm is shown, but depending on the transistor gain, R1 can be between say, 50kOhm and 1kOhm - between 1kOhm and 10kOhm should work with just about anything though)  
The R\_pull up of 1kOhm is assuming about the worst case - it will probably be between 2kOhm and 10kOhm. The circuit as shown would work with a pull up down to around 100 ohms though if needed.

|  |
| --- |
|  |

Nodemcu

NodeMCU is an open-source firmware for which open-source prototyping board designs are available. The name “NodeMCU” combines “node” and “MCU” (micro-controller unit). The term “NodeMCU” strictly speaking refers to the firmware rather than the associated development kits. Both the firmware and prototyping board designs are open source. Nodemcu ESP8266 and Nodemcu ESP32 are becoming very popular and are almost used in more then 50% IoT based projects today.

The firmware uses the Lua scripting language. The firmware 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 and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit [ESP32](https://www.electroniclinic.com/esp32-arduino-ide-board-manager-installation-espressif-esp32-wroom/) has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially was based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications.

The components and tools used in this tutorial can be purchased from Amazon, the components Purchase links are given below:

[Nodemcu ESP8266 WiFi Module:](https://amzn.to/3sx24l4)

Other Tools and Components:

[Super Starter kit for Beginners](https://amzn.to/3cGmLEv)

[Digital Oscilloscopes](https://amzn.to/3bSarSk)

[Variable Supply](https://amzn.to/2NwnfVm)

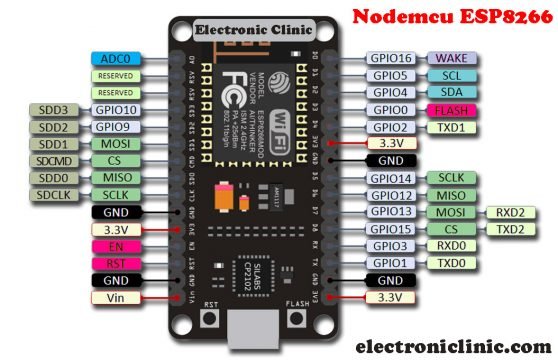
[Digital Multimeter](https://amzn.to/3lmU4k5)

[Soldering iron kits](https://amzn.to/3s1LcTy)

[PCB small portable drill machines](https://amzn.to/3loq478)

*\*Please Note: These are affiliate links. I may make a commission if you buy the components through these links. I would appreciate your support in this way!*

About the Nodemcu ESP8266 Pinout:

[](https://www.electroniclinic.com/wp-content/uploads/2020/06/NODEMCU-ESP8266-Pinout-features-and-specifications.jpg)

NodeMCU ESP8266 Wifi Module is an open-source Lua based firmware and development board specially targeted for IoT based applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

Nodemcu ESP8266 Specifications & Features

Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106

Operating Voltage: 3.3V

Input Voltage: 7-12V

Digital I/O Pins (DIO): 16

Analog Input Pins (ADC): 1

UARTs: 1

SPIs: 1

I2Cs: 1

Flash Memory: 4 MB

SRAM: 64 KB

Clock Speed: 80 MHz

USB-TTL based on CP2102 is included onboard, Enabling Plug n Play

PCB Antenna

Small Sized module to fit smartly inside your IoT projects

Nodemcu ESP8266 Pinout:

For practical purposes ESP8266 NodeMCU V2 and V3 boards present identical pinouts. While working on the NodeMCU based projects we are interested in the following pins.

Power pins (3.3 V).

Ground pins (GND).

Analog pins (A0).

Digital pins (D0 – D8, SD2, SD3, RX, and TX – GPIO XX)

Most ESP8266 NodeMCU boards have one input voltage pin (Vin), three power pins (3.3v), four ground pins (GND), one analog pin (A0), and several digital pins (GPIO XX).

Pin      Code               Arduino alias

A0       A0                               A0

D0       GPIO 16                     16

D1       GPIO 5                       5

D2       GPIO 4                       4

D3       GPIO 0                       0

D4       GPIO 2                       2

D5       GPIO 14                     14

D6       GPIO 12                     12

D7       GPIO 13                     13

D8       GPIO 15                     15

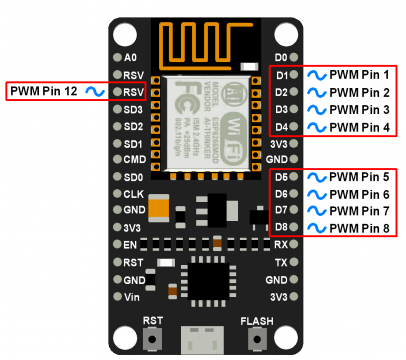
SD2     GPIO 9                       9

SD3     GPIO 10                     10

RX      GPIO 3                       3

TX       GPIO 1                       1

Nodemcu ESP8266 PWM Pins:

[](https://www.electroniclinic.com/wp-content/uploads/2020/06/NodeMCU_PWM_Pins.png)

Applications of Nodemcu

Prototyping of IoT devices

Low power battery operated applications

Network projects

Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities

MOTOR

Whenever a motorics hobbyist talk about making a motor, the first thing comes to his mind is making the motor move on the ground. And there are always two options in front of the designer whether to use a [DC motor](http://www.8051projects.net/dc-motor-interfacing/introduction.php) or a stepper motor. When it comes to speed, weight, size, cost... DC motors are always preferred over stepper motors. There are many things which you can do with your DC motor when interfaced with a microcontroller. For example you can control the speed of motor, you can control the direction of rotation, you can also do encoding of the rotation made by DC motor i.e. keeping track of how many turns are made by your motors etc. So you can see DC motors are no less than a stepper motor.

### DETAILS ABOUT DC MOTORS

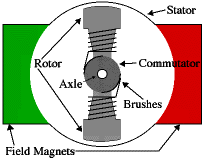
### Permanent magnet DC motor responds to both voltage and current. The steady state voltage across a motor determines the motor’s running speed, and the current through its armature windings determines the torque. Apply a voltage and the motor will start running in one direction; reverse the polarity and the direction will be reversed. If you apply a load to the motor shaft, it will draw more current, if the power supply does not able to provide enough current, the voltage will drop and the speed of the motor will be reduced. However, if the power supply can maintain voltage while supplying the current, the motor will run at the same speed. In general, you can control the speed by applying the appropriate voltage, while current controls torque. In most cases, DC motors are powered up by using fixed DC power supply, therefore; it is more efficient to use a chopping circuit. Consider what happens when a voltage applied to a motor’s windings is rapidly turned ON and OFF in such a way that the frequency of the pulses produced remains constant, but the width of the ON pulse is varied. This is known as Pulse Width Modulation (PWM). Current only flows through the motor during the ON portion of the PWM waveform. If the frequency of the PWM input is high enough, the mechanical inertia of the motor cannot react to the ripple wave; instead, the motor behaves as if the current were the DC average of the ripple wave. Therefore, by changing the width of pulse, we can control the motor speed.

### At the most basic level, electric motors exist to convert electrical energy into mechanical energy. This is done by way of two interacting magnetic fields -- one stationary, and another attached to a part that can move. A number of types of electric motors exist, but most BEAM bots use DC motors in some form or another. DC motors have the potential for very high torque capabilities (although this is generally a function of the physical size of the motor), are easy to miniaturize, and can be "throttled" via adjusting their supply voltage. DC motors are also not only the simplest, but the oldest electric motors. Oersted, Gauss, and Faraday discovered the basic principles of electromagnetic induction in the early 1800’s. By 1820, Hans Christian Oersted and Andre Marie Ampere had discovered that an electric current produces a magnetic field. The next 15 years saw a flurry of cross-Atlantic experimentation and innovation, leading finally to a simple DC rotary motor. A number of men were involved in the work, so proper credit for the first DC motor is really a function of just how broadly you choose to define the word "motor

### Principles of operation

### In any electric motor, operation is based on simple electromagnetism. A current carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current -carrying conductor and an external magnetic field to generate rotational motion. Let's start by looking at a simple 2-pole DC electric motor (here dark black represents a magnet or winding with a "North" polarization, while light color represents a magnet or winding with a "South" polarization).

### Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet’s, and brushes. In most common DC motors, the external magnetic field is produced by high-strength permanent magnets. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotates with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.



The geometry of the brushes, commutator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commutator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.

In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

D.C. Motors with field coils are classified as series. Shunt, compound and separately excited according to how the field windings and armature windings are connected. With the series wound motor the armature and fields coils are in series. Such a motor exerts the highest starting torque and has the greatest no-load speed. With light loads there is a danger that a series wound motor might run at too high a speed. Reversing the polarity of the supply to the coils has no effect on the direction of rotation of the motor, it will continue rotating in the same direction since both the field and armature currents have been reversed.

### With the shunt wound motor the armature and field coils are in parallel. it provides the lowest starting torque, a much lower no- load speed and has good speed regulation. Because of this almost constant speed regardless of load, shunt wound motors are very widely used to reverse the direction of rotation, either the armature or field supplied must be reversed. For this reason, the separately excited windings are preferable for such a situation. The compound motor has two field windings, one in series with the armature and one in parallel. Compound wound motors aim to got the best features of the series and shunt wound motors, namely a high starting torque and good speed regulation. The separately excited motor has separate control of the armature and field currents and can be considered to be a special case of the shunt wound motor. The torque-speed characteristics of the above motors and the speed of such D.C. Motors can be changed by either changing the armature current or the field current. Generally it is the armature current that is varied. The choice of motor will depend on its application. For example, with a robot, manipulator, and the robot wrist might use a series wound motor because the speed decreases as the load increases. a shunt wound motor would be used where a constant speed was required, regardless of the load. The speed of a permanent magnet motor depends on the current through the armature coil. With a field coil motor either varying the armature current or the field current can change the speed; generally it is the armature current that is varied. Thus speed control can be obtained by controlling the voltage applied to the armature. However, because fixed voltage supplies are often used, an electronic circuit obtains a variable voltage.

### With an alternating current supply, the thyristor circuit can be used to control the average voltage applied to the armature. However, we are often concerned with the control of D.C. Motors by means of control signals emanating from microprocessors. In such cases the technique known as pulse width modulation (PWM) is generally used. This basically involves taking a constant D.C. supply voltage and chopping it so that the average value is varied.

GPS:

A GPS tracking unit, geo-tracking unit, or simply tracker is a navigation device normally on a vehicle, asset, person or animal that uses the Global Positioning System (GPS) to determine its movement and determine its WGS84 UTM geographic position (geo-tracking) to determine its location.GPS tracking devices send special satellite signals that are processed by a receiver.

Locations are stored in the tracking unit or transmitted to an Internet-connected device using the cellular network (IOT/GPRS/CDMA/LTE or SMS), radio, or satellite modem embedded in the unit or Wi-Fi work worldwide.

GPS (GobalPostioning system):

Introduction

Global Positioning System (GPS) technology is changing the way we work and play. You can use GPS technology when you are driving, flying, fishing, sailing, hiking, running, biking, working, or exploring. With a GPS receiver, you have an amazing amount of information at your fingertips. Here are just a few examples of how you can use GPS technology.

* Know precisely how far you have run and at what pace while tracking your path so you can find your way home
* Pinpoint the perfect fishing spot on the water and easily relocate it
* Get the closest location of your favorite restaurant when you are out-of-town
* Find the nearest airport or identify the type of airspace in which you are flying

What is GPS ?

The Global Positioning System (GPS) is a satellite-based navigation system that sends and receives radio signals. A GPS receiver acquires thesesignals and provides you with information. Using GPS technology, you can determine location, velocity, and time, 24 hours a day, in any weather conditions anywhere in the world—for freeGPS, formally known as the NAVSTAR (Navigation Satellite Timing and Ranging) Global Positioning System, originally was developed for the military. Because of its popular navigation capabilities and because you can access GPS technology using small inexpensive equipment, the government made the system available for civilian use. The USA owns GPS technology and the Department of Defense maintains it.

GPS technology requires the following three segments

* Space segment
* Control segment
* User segment

Space Segment

At least 24 GPS satellites orbit the earth twice a day in a specific pattern. They travel at approximately 7,000 miles per hour about 12,000 miles above the earth’s surface. These satellites are spaced so that a GPS receiver anywhere in the world can receive signals from at least four of them.

Each GPS satellite constantly sends coded radio signals to the earth. These GPS satellite signals contain the following information

* The particular satellite that is sending the information
* Where that satellite should be at any given time (the precise location of the satellite iscalled ephemeris data)
* Whether or not the satellite is working properly
* The date and time that the satellite sent the signal

The signals can pass through clouds, glass, and plastic. Most solid objects such as buildings attenuate (decrease the power of) the signals. These signals cannot pass through objects that contain a lot of metal or objects that contain water (such as underwater locations)

The GPS satellites are powered by solar energy. If solar energy is unavailable, for example, when the satellite is in the earth’s shadow, thesatellites use backup batteries to continue running. Each GPS satellite is built to last about 10 years. The Department of Defense monitors and replaces the satellites to ensure that GPS technology continues to run smoothly for years to come

Control Segment

The control segment is responsible for constantly monitoring satellite health, signal integrity, and orbital configuration from the ground. The control segment includes the following sections.

* Master control station
* Monitor stations
* Ground antennas

Monitor Stations

At least six unmanned monitor stations are located around the world. Each station constantly monitors and receives information from the GPS satellites and then sends the orbital and clock information to the master control station (MCS).

Master Control Station (MCS)

The MCS is located near Colorado Springs in Colorado. The MCS constantly receives GPS satellite orbital and clock information from themonitor stations. The controllers in the MCS make precise corrections to the data as necessary, and send the information (known as ephemeris data) to the GPS satellites using the ground antennas

Ground Antennas

Ground antennas receive the corrected orbital and clock information from the MCS, and then send the corrected information to the appropriate satellites.

User Segment

The GPS user segment consists of your GPS receiver. Your receiver collects and processes signals from the GPS satellites that are in viewand then uses that information to determine and display your location, speed, time, and so forth. Your GPS receiver does not transmit any information back to the satellites.

How Does GPS Technology Work?

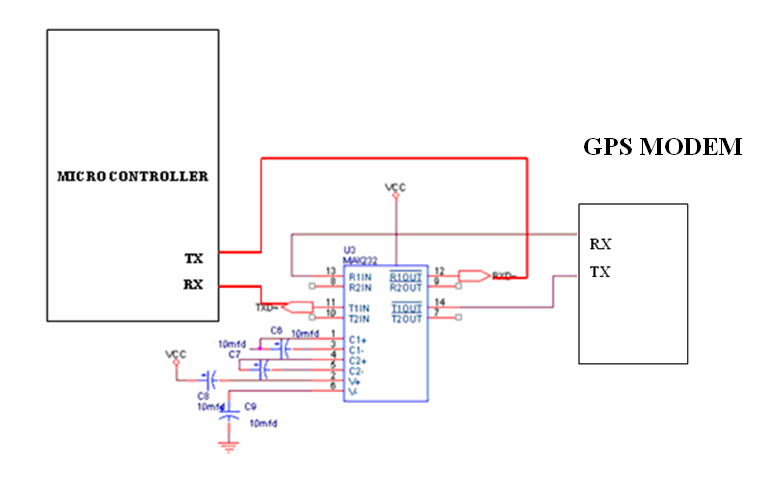
The following points provide a summary of the technology at work:

* The control segment constantly monitors the GPS constellation and uploads information to satellites to provide maximum user accuracy.
* Your GPS receiver collects information from the GPS satellites that are in view.
* Your GPS receiver accounts for errors. For more information, refer to the Sources
* Your GPS receiver determines your current location, velocity, and time.
* Your GPS receiver can calculate other information, such as bearing, track, trip distance, distance to destination, sunrise and sunset time, andso forth
* Your GPS receiver displays the applicable information on the screen

APPLICATION OF THE GPS TRACKING SYSTEM

* Fleet control. For example, a delivery or taxi company may put such a tracker in every of its vehicles, thus allowing the staff to know if a vehicle is on time or late, or is doing its assigned route. The same applies for armored trucks transporting valuable goods, as it allows to pinpoint the exact site of a possible robbery.
* Stolen vehicle searching . Owners of expensive cars can put a tracker in it, and "activate" them in case of theft. "Activate" means that a command is issued to the tracker, via SMS or otherwise, and it will start acting as a fleet control device, allowing the user to know where the thieves are.
* Animal control. When put on a wildlife animal (e.g. in a collar), it allows scientists to study its activities and migration patterns. Vaginal implant transmitters are used to mark the location where pregnant females give birth.Animal tracking collars may also be put on domestic animals, to locate them in case they get lost.
* Race control . In some sports, such as gliding, participants are required to have a tracker with them. This allows, among other applications, for race officials to know if the participants are cheating, taking unexpected shortcuts or how far apart they are. This use has been featured in the movie "Rat Race", where some millionaires see the position of the racers in a wall map.
* Espionage/surveillance. When put on a person, or on his personal vehicle, it allows the person monitoring the tracking to know his/her habits. This application is used by private investigators, and also by some parents to track their children.
* Internet Fun . Some Web 2.0 pioneers have created their own personal web pages that show their position constantly, and in real-time, on a map within their website. These usually use data push from a GPS enabled cell phone.

GPS INTERFACE WITH MICROCONTROLLER:



In this project GPS Modem is interfaced with the microcontroller through rs232 interface. Since the voltage levels of the microcontroller are different with that of the GPS modem we use a voltage converter or the line driver such as MAX232 to make them rs232 compatible.

SMS:

SMS technology originated from radio telegraphy in radio memo pagers that used standardized phone protocols .These were defined in 1986 as part of the Global system for mobile communication (IOT)series of standard of series of standard. The first test SMS message was sent on December 3,1992 ,When Neil papworth , a test engineer for sema group uses a personal computer to send “merry Christmas “ to the phone of colleague Richard Jarvis .SMS rolled out commercially on many cellular networks that decade and become hugely popular worldwide as a method of text communication.



GPS antenna size limits tracker size, often smaller than a half-dollar (diameter 30.61 mm). In 2020 tracking is a $2 billion business plus military-in the gulf war 10% or more targets used trackers. Virtually every cellphone tracks its movements and per most cell user agreements uploads the track data, creating trillions of sellable locations and tracks, value varies from fractions of a mil to dollars per point and user association. Tracks can be map displayed in real time, using GPS tracking software and devices with GPS capability.

**RF Transmitter**

**RF 1 433/315 MHz (SP)**

[Radio](http://www.electronics-manufacturers.com/info/) frequency (RF) transmitters are widely used in radio frequency communications systems. With the increasing availability of efficient, low cost [electronic](http://www.electronics-manufacturers.com/info/) modules, mobile communication systems are becoming more and more widespread. [Wireless communications](http://www.electronics-manufacturers.com/info/) systems, including cellular phones, paging devices, personal communication services (PCS) systems, and wireless data networks, have become ubiquitous in society. Generally, a radio transmitter and receiver is used for performing a radio transmission and receiving operation, whereby a high frequency signal outputted from a modulator is transmitted to an antenna of the radio transmitter and is transmitted there from to a remote radio transmitter and receiver, or the thusly transmitted signal is received through another antenna. The transmitting baseband signal is subjected to a predetermined signal process, input to a modulator, which modulates a carrier wave signal. The modulated carrier wave signal is converted into a radio frequency by a transmitting radio-frequency circuit and amplified to a predetermined transmitting power. In general, the function of a radio frequency (RF) transmitter is to modulate, up convert, and amplify signals for transmission into free space. An RF transmitter generally includes a modulator that modulates an input signal and a radio frequency power amplifier that is coupled to the modulator to amplify the modulated input signal. The radio frequency power amplifier is coupled to an antenna that transmits the amplified modulated input signal.

**Specifications**

Voltage : 3 to 12 volts

Current : 10 to 15 mA

Working Mode : AM

Speed : 4 Kbps

Frequency : 315/433 MHz

External Antenna : 315 MHz

**Pin Out**

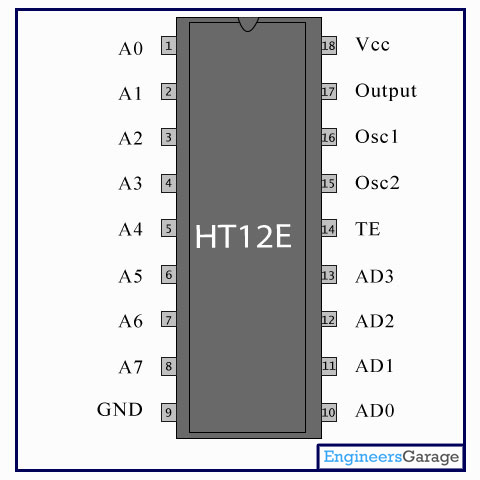
1. Data
2. Vdd
3. Vss
4. Antena

**Encoder HT 12E**

**General Description**

Encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding 18 bits of information which consists of N address bits and 18\_N data bits. Encoders offer flexible combinations of programmable address/data to meet various application needs. The programmable address/ data is transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal.

**Pin Assignment:**

****

**Features**

* Operating voltage: 2.4V~12V
* Low power and high noise immunity CMOS technology
* Low standby current
* Three words transmission
* Built-in oscillator needs only 5% resistor
* Easy interface with an RF or infrared transmission media
* Minimal external components

**Applications**

* Burglar alarm system
* Smoke and fire alarm system
* Garage door controllers
* Car door controllers
* Car alarm system
* Security system
* Cordless telephones
* Other remote control systems

**RF Receiver Module**

**RF 1 433/315 MHz (SP)**

**Specifications**

Voltage : 5v

Frequency : 315/433 MHz

External Antenna : 315 MHz

Speed : 4KB/S

Current : 0.5-0.8 mA

**PIN OUT**

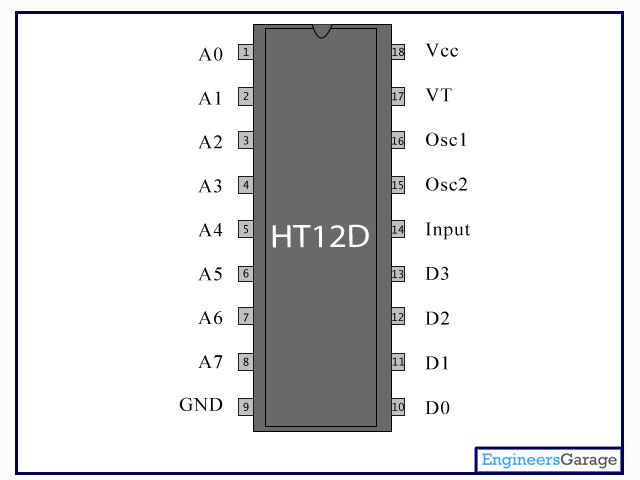
1. Vdd
2. Data Out
3. Vss
4. Antenna

**Decoders HT 12D**

**General Description**

The decoders are a series of CMOS LSIs for remote control system applications. They are paired with the series of encoders. For proper operation a pair of encoder/decoder pair with the same number of address and data format should be selected (refer to the encoder/decoder cross reference tables). The series of decoders receives serial address and data from that series of encoders that are transmitted by a carrier using an RF or an IR 1transmission medium. It then compares the serial input data twice continuously with its local address. If no errors or unmatched codes are encountered, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission. The decoders are capable of decoding 18 bits of information that consists of N bits of address and 18–N bits of data. To meet various applications they are arranged to provide a number of data pins whose range is from 0 to 8 and an address pin whose range is from 8 to 18.

**Pin Assignment:**

****

**Features**

Operating voltage: 2.4V~12V

Low power and high noise immunity CMOS technology

Low standby current

Capable of decoding 18 bits of information

Pairs with HOLTEK’s 318 series of encoders

8~18 address pins

0~8 data pins

Ternary address setting

Two times of receiving check

Built-in oscillator needs only a 5% resistor

Valid transmission indictor

Easily interface with an RF or an infrared transmission medium

Minimal external components

**Applications**

Burglar alarm system

Smoke and fire alarm system

Garage door controllers

Car door controllers

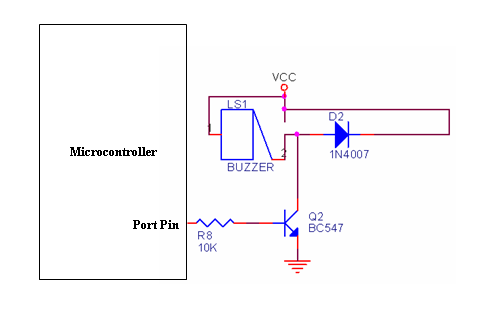
Car alarm system

Buzzer:

A buzzer or beeper is an [audio](https://en.wikipedia.org/wiki/Sound) signaling device, which may be [mechanica](https://en.wikipedia.org/wiki/Machine)l, [electromechanical](https://en.wikipedia.org/wiki/Electromechanics), or [piezoelectric](https://en.wikipedia.org/wiki/Piezoelectricity) (*piezo* for short). Typical uses of buzzers and beepers include [alarm devices](https://en.wikipedia.org/wiki/Alarm_devices), [timers](https://en.wikipedia.org/wiki/Timer), and confirmation of user input such as a mouse click or keystroke.



BUZZER INTERFACE WITH MICROCONTROLLER



**Software**

Arduino idle

Arduino is a both an open-source software library and an open-source breakout board for the popular AVR micro-controllers. The **Arduino IDE** (Integrated Development Environment) is the program used to write code, and comes in the form of a downloadable file on the Arduino website. The **Arduino board** is the physical board that stores and performs the code uploaded to it. Both the software package and the board are referred to as **"Arduino."**

Integrated Development Environment tool is referred to as an IDE by using this tool entire programming for proposed system is done. The required Baud rate for serial communication between an Arduino Bluetooth module and a smart phone is 9600 bits per second. The sample Arduino commands were discussed. When the serial data is available or not explained from the adjacent instruction "Serial.available( )> 0" means that receive data serially from smart phone via Bluetooth and "Serial.println( )" instruction is described that to send data serially from Arduino controller board to smart phone. State variable is to get the value from smart phone to store the values and then compare with different condition and execute the required task as per the program code logics.

To begin, download the Arduino IDE from the Arduino website. Make sure to select the right version for your Operating System (OS). For a full getting started guide for each OS, please refer to the Arduino guide. Once the arduino.zip file has been downloaded, extract the file to a folder somewhere on your computer. There is no install simply open the folder and double click the .exe.

**Source code**

**Results**

**Conclusion:**

The system designed provides safety of the riders, in case of accidents it will notify the registered contact and the location of the accident provides a timely safety measure. This also detects the consumption of alcohol and prevents drink and drive cases. This also ensures the person wears the helmet mandatorily.

1. Syan Tapadar, Arnab Kumar Saha, Dr Himadri Nath Saha, Shinjini Ray, Robin Karlose Accident and Alcohol detection in Bluetooth enabled Smart Helmets for motorbikes.
2. Nitin Agarwal, Anshul Kumar Singh, Pushpender Pratap Singh, Rajesh Sahani, SMART HELMET, International Research Journal of Engineering and Technology, Volume 2, issue 2, May 2015.
3. N. Fathima, A. Ahammed, R. Banu, B.D. Parameshachari, and N.M. Naik, Optimized neighbor discovery in Internet of Things (IoT), In Proc. of International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), pp. 1-5, 2017.

We can implement various bioelectric sensors on the helmet to measure various activities and we can view the statistics of the rider. We can use voice commands to control the basic bike functionalities. Now the rider can leave the helmet on the two-wheeler while parking, without any special actions or security measures. We can use solar energy on two- wheelers for charging the electric vehicles and for mobile devices. In the future self – driving motorbikes can be developed with artificial intelligence and the rider will be safe and no accidents will occur.

References:

1. Bindu Sebastian Priyanka Kp, Hridhya Kuttikrishanan, Smart Helmet International Journal of Technology & Advanced Engineering, Volume5, Issue:12, December 2015.
2. Professor Chitte P.P., Salunke Akshay S., Thorat Aniruddha, N Bhosale, Smart Helmet & Intelligent Bike System, International Research Journal of Engineering and Technology (IRJET) Volume: 03 Issue: 05, May2016.
3. Jianyun Ni; Jing Luo; "Microcontroller-based engineering education innovation, " Educational and Information Technology (ICEIT), 2010 International Conference on, vol.3, no., pp. V3-109-V3-112, 17-19 Sept. 2010.
4. S. Chandran, S. Chandrashekhar, E. Elizabeth N, Konnect: An Internet of Things (IoT) based Smart Helmet for Accident Detection and Notification, India Conference (INDICON), 2016 IEEE Annual.
5. Jennifer William, Kaustubh Padwal, Nexon Samuel, Akshay Bawkar, Smita Rukhande intelligent Helmet International Journals of Scientific& Engineering Research, volume 7, issue 3, March-2016.
6. Shoeb Ahmed Shabbeer, Merin Melleet Smart helmet for accident detection and notification 2nd IEEE international conference on computational systems and information technology 2017.