**AUTOMATIC BORDER CROSSING DETECTION AND NAVIGATION FOR FISHING VESSELS**

A project report submitted in parallel fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**SUBMITTED BY**

**G. NAGA JYOTHI A. ANIL B. NANDHA KISHORE SAI**

**19B81A0452 20B85A0401 20B85A0402**

**A. NAVEENA CH. VAMSI**

**19B81A0406 19B81A0488**

Under the guidance of

Dr. K. LAKSHMI NARAYANA, M.Tech.(Ph.D.)

****ASSOCIATE PROFESSOR

**DEPARTMENT OF**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**SIR C.R. REDDY COLLEGE OF ENGINEERING**

**(AFFILIATED TO JNTUK)**

**ELURU, A.P, INDIA**

**A.Y.2022-2023**

**DEPARTMENT OF**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**SIR C R REDDY COLLEGE OF ENGINEERING**

**(AFFILIATED TO JNTUK)**

****

**CERTIFICATE**

This is to certify the project report entitled “**AUTOMATIC BORDER CROSSING DETECTION AND NAVIGATION FOR FISHING VESSELS**” is a bonafide project work was done by the following students:

**G. NAGA JYOTHI A. ANIL B. NANDHA KISHORE SAI**

**19B81A0452 20B85A0401 20B85A0402**

**A. NAVEENA CH. VAMSI**

**19B81A0406 19B81A0488**

During the **A.Y 2022-23** As a fulfillment of the academic requirements for the award of **Bachelor of Technology** degree **in Department of ELECTRONICS OF COMMUNICATION ENGINEERING** from **JNTUK.**

**PROJECT GUIDE HEAD OF THE DEPARTMENT**

**Dr. K. LAKSHMINARAYANA Dr. T. VENKATESWARARAO.**

**M.Tech. (Ph.D) M. Tech, Ph. D(ANU), Ph.D. JNTUK**

**DEPARTMENT OF ECE**

**SIR C.R.R ENGG**

**DECLARATION**

We, the students of IV/IV B. Tech, ELECTRONICS AND COMUNICATION ENGINEERING DEPARTMENT, Sir C.R. Reddy College of Engineering, have successfully completed the project work entitled “**AUTOMATIC BORDER CROSSING DETECTION AND NAVIGATION FOR FISSING VESSELS**” which has been carried out by us under the guidance of **Dr. K. LAKSHMI NARAYANA**, M.Tech, (Ph.D.), Associate professor of the department. We declare that the above work is not submitted at any university for award of any degree.

**PROJECT MEMBERS:**

G. NAGA JYOTHI 19B81A0452

A. ANIL 20B85A0401

B. NANDHA KISHORE SAI 20B85A0402

A. NAVEENA 19B81A0406

CH. VAMSI 19B81A0432

**ACKNOWLEDGEMENT**

The austerity and satisfaction that one gets on completing a project cannot be fulfilled without mentioning the people who made it possible with gratitude. We are very much grateful to the Almighty who helped us all the way throughout the project work and who has molded us into what we are today.

We wish to express our sincere thanks to various personalities who were responsible for the successful completion of this project.

We thanks to our Principal, **Prof. Dr. K. VENKATESWARA RAO** for providing the necessary infrastructure required for our project.

We are grateful to **Dr. T. VENKATESWARA RAO**, M.Tech., Ph.D., Head of Electronics and Communication Engineering Department, for providing the necessary facilities and valuable guidance for complete the project in specified time and valuable guidance.

Last but not least we also express our sincere thanks to all the faculty members of the department for their help and cooperation in complete this project.

**PROJECT MEMBERS:**

G. NAGA JYOTHI 19B81A0452

A. ANIL 20B85A0401

B. NANDHA KISHORE SAI 20B85A0402

A. NAVEENA 19B81A0406

CH. VAMSI 19B81A0432

**AUTOMATIC BORDER CROSSING DETECTION AND NAVIGATION OF BOAT**

**Abstract:**

Fishermen's livelihoods are so insecure that if they cross the country's border accidently, it is treated as a serious offence. The severity of the situation seems to worsen when impoverished fisherman are shot and their boats are taken. The fundamental cause of this problem is that marine borders between countries are difficult to identify. The term "maritime border alert system" refers to a system that assists fishermen by informing them of the country's border while protecting them and their boats. Technology such as global navigation satellite system (GPS), universal mobile telecommunications network (GSM) are used to achieve this purpose. The proposed framework includes a low-cost maritime boundary crossing warning system for fishermen.

**Chapter.1**

**Introduction**

The main aim of this paper is to save the lives of the people who unintentionally or unknowingly cross their country border. Recently 12 Indian fishermen were arrested by Sri Lankan navy for illegal entry and also they allegedly shot dead a 21 year old fisherman and injured some others when they opened fire on fishermen. Not only in Srilankan border there is a threat to the lives of the fishermen at Gujarat. Gujarati fishermen those who go for fishing near the Pakistan borders are under threat. Recently 25 fishermen were abducted by Pakistan soldiers and many were injured.

**Chapter.2**

**Literature survey**

* **ARUNVIJAY et al “DESIGN OF BORDER ALERT SYSTEM FORFISHERMEN USING GPS”, MARCH 2014**

In this method the author uses the pre-determined values of latitude and longitude points of the maritime border shown in the fig 2.1; this is stored in the microcontroller. When the boat approaches the border, boats position (latitude and longitude) is measured using GPS and compared with the stored value, if it exceeds then the boats seems to be crossed and alert message is sent to the fisherman.The advantage is accuracy range is high with the use of GPS.but the drawback is the memory required saving each point of latitude and longitude is more.

* **VIGNESH M et al, “GPS BASED BORDER ALERT SYSTEMFOR FISHERMEN WITH BOAT SPEEDOMETER”, MARCH 2015**

In this method the author foresee the use of GPS tracking system. The boats position is measured using GPS and the speed of the boats motor is controlled in case of emergency. The alert message is sent to the user (fisherman) .The advantage is for the purpose of identification the fisherman are using the GPS72h, equipment used for the navigation in sea and it provides the fastest and most accurate method for mariner navigate, measure speed, and determines location and this system enables increased levels of safety and efficiency. The disadvantage here is that border alert is intimated only to the fishermen but not to the control station.

* **MICHALSKI et al “DESIGN FOR BORDER SAFETY SYSTEM”, MARCH 2013**

In this method the boats position is measured using GPS and as the boat approaches the border limit the ignition to the motor is cut which means the motor is stopped it cannot further move forward and only by taking reverse gear the boat can be restarted. As the boat reaches the restricted zone the boat stops and only when you reverse the motor it can start again. It ensures maximum safety to the fishermen well in advance. But due to wave currents, reversing the motor may mislead the boat.

**EXISTING SYSTEM**

Conflicts exist between the two republics due to their shared coastline border. It's not a fight for land or for individuals, but it's a fight for fish, which are the lifeblood of these coastal fishing communities. Fishermen face tremendous risk when they travel by boat to catch fish. Fishermen are being stopped from fishing, harassed, and arrested by the navy of the neighbouring country.

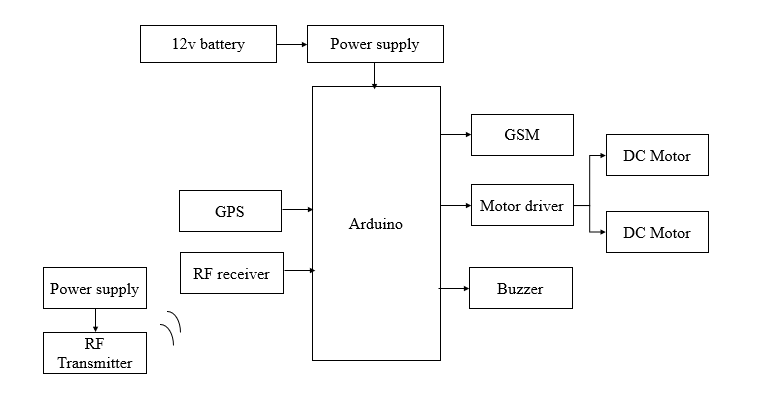
**Drawbacks of existing method:**

* Life threat for fisherman

**PROPOSED SYSTEM**

The proposed system Arduino is inexpensive and has open-source hardware and software. Arduino is used in a system connected with GPS, GSM and a Buzzer. The receiver end of Arduino is connected with GPS and the transmitter end is connected with GSM of Arduino Uno. By using embedded C, a predefined location is set and if this is crossed, an SMS alert will be sent to a predefined number and also the boat turns into reverse direction. Here we are implementing a robot which we can control the RF transmitter to Receiver, receiver is connected to the robot from transmitter we can control the robot. Whenever the robot is near to the border it will automatically stop and gives the alerts through Buzzer and SMS.

**Block diagram:**



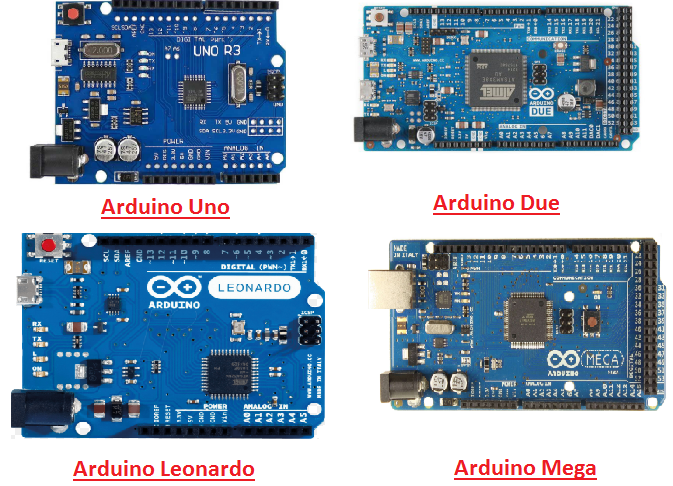
**Chapter.3**

**HARDWARE REQUIREMENTS**

**Arduino:**

Arduino Uno is a very valuable addition in the electronics that consists of USB interface, 14 digital I/O pins, 6 analog pins, and Atmega328 microcontroller. It also supports serial communication using Tx and Rx pins.

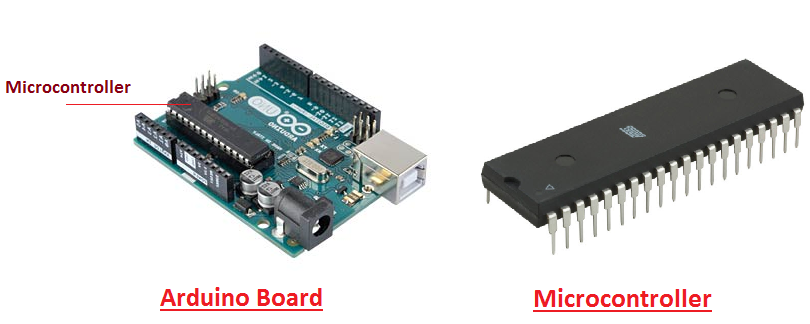
There are many versions of Arduino boards introduced in the market like Arduino Uno, Arduino Due, Arduino Leonardo, Arduino Mega, however, most common versions are Arduino Uno and Arduino Mega. If you are planning to create a project relating to digital electronics, embedded system, robotics, or IoT, then using Arduino Uno would be the best, easy and most economical option.



It is an open-source platform, means the boards and software are readily available and anyone can modify and optimize the boards for better functionality.

The software used for Arduino devices is called IDE (Integrated Development Environment) which is free to use and required some basic skills to learn it. It can be programmed using C and C++ language.

Some people get confused between **Microcontroller and Arduino**. While former is just an on system 40 pin chip that comes with a built-in microprocessor and later is a board that comes with the microcontroller in the base of the board, bootloader and allows easy access to input-output pins and makes uploading or burning of the program very easy.

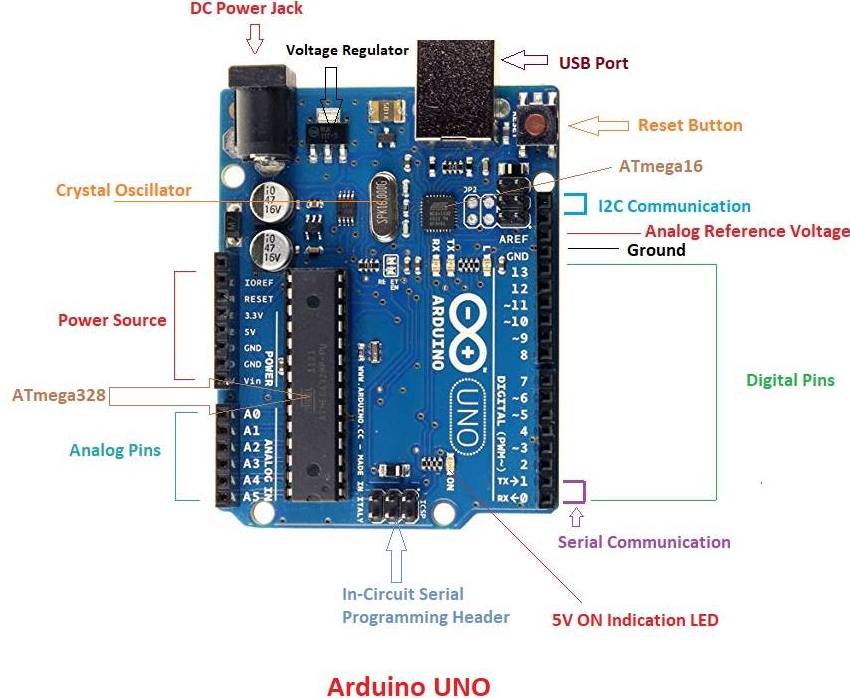


While learning microcontroller requires some expertise and skills.

Nevertheless, we can say every Arduino is basically a [microcontroller](https://www.theengineeringprojects.com/2018/03/introduction-to-microcontrollers.html) but not every microcontroller is an Arduino.

**Introduction to Arduino**

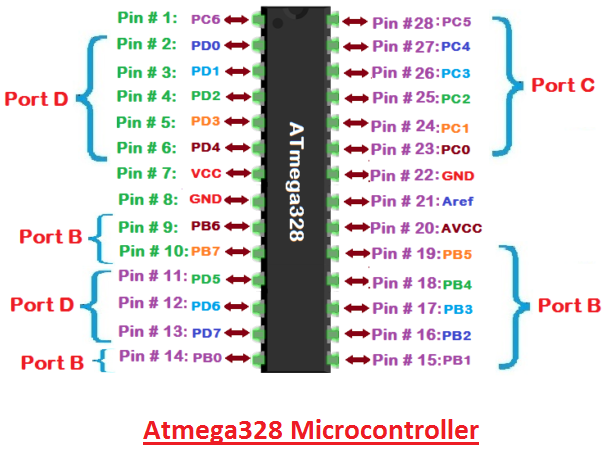
* **Arduino Uno** is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328.
* First Arduino project was started in Interaction Design Institute Ivrea in 2003 by David Cuartielles and Massimo Banzi with the intention of providing a cheap and flexible way to students and professional for controlling a number of devices in the real world.
* The current version of Arduino Uno comes with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output.
* It allows the designers to control and sense the external electronic devices in the real world

****

* This board comes with all the features required to run the controller and can be directly connected to the computer through USB cable that is used to transfer the code to the controller using IDE (Integrated Development Environment) software, mainly developed to program Arduino. IDE is equally compatible with Windows, MAC or Linux Systems, however, Windows is preferable to use. Programming languages like C and C++ are used in IDE.
* Apart from USB, battery or AC to DC adopter can also be used to power the board.
* Arduino Uno boards are quite similar to other boards in Arduino family in terms of use and functionality, however, Uno boards don’t come with FTDI USB to Serial driver chip.
* There are many versions of Uno boards available, however, Arduino Nano V3 and Arduino Uno are the most official versions that come with Atmega328 8-bit AVR Atmel microcontroller where RAM memory is 32KB.
* When nature and functionality of the task go complex, Mirco SD card can be added in the boards to make them store more information.

**Features of Arduino**

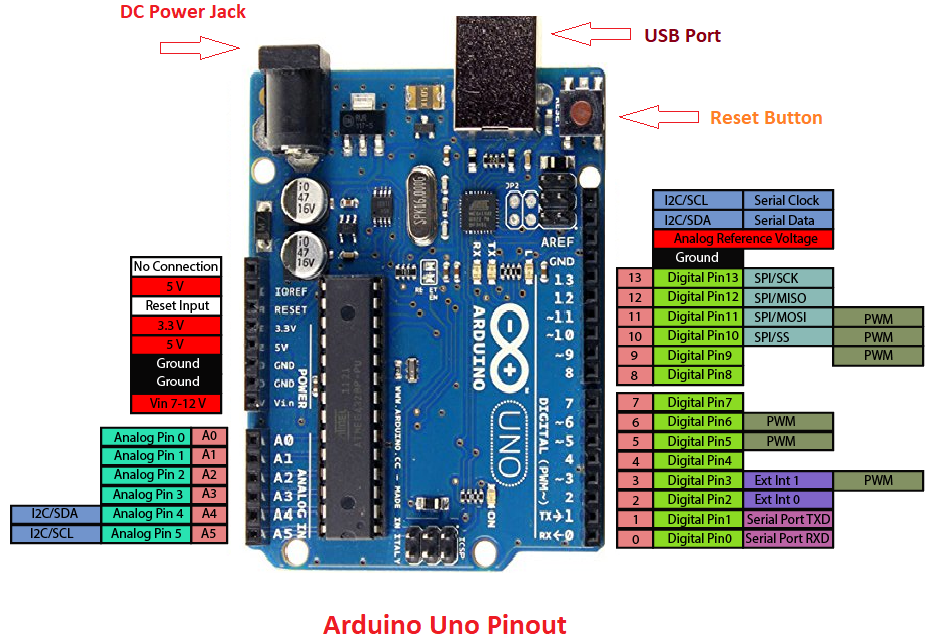
* Arduino Uno comes with USB interface i.e. USB port is added on the board to develop serial communication with the computer.
* [Atmega328](https://www.theengineeringprojects.com/2017/08/introduction-to-atmega328.html) microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle.

****

* It is an open source platform where anyone can modify and optimize the board based on the number of instructions and task they want to achieve.
* This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.
* Reset pin is added in the board that reset the whole board and takes the running program in the initial stage. This pin is useful when board hangs up in the middle of the running program; pushing this pin will clear everything up in the program and starts the program right from the beginning.
* There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins. There is no hard and fast interface required to connect the devices to the board. Simply plug the external device into the pins of the board that are laid out on the board in the form of the header.
* The 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to the high range using analogReference() function and AREF pin.
* 13KB of flash memory is used to store the number of instructions in the form of code.
* Only 5 V is required to turn the board on, which can be achieved directly using USB port or external adopter, however, it can support external power source up to 12 V which can be regulated and limit to 5 V or 3.3 V based on the requirement of the project.

**Arduino Pinout**

* Arduino Uno is based on AVR microcontroller called Atmega328. This controller comes with 2KB SRAM, 32KB of flash memory, 1KB of EEPROM. Arduino Board comes with 14 digital pins and 6 analog pins. ON-chip ADC is used to sample these pins. A 16 MHz frequency crystal oscillator is equipped on the board. Following figure shows the pinout of the Arduino Uno Board

****

**Pin Description:**

There are several I/O digital and analog pins placed on the board which operates at 5V. These pins come with standard operating ratings ranging between 20mA to 40mA. Internal pull-up resistors are used in the board that limits the current exceeding from the given operating conditions. However, too much increase in current makes these resisters useless and damages the device.

**LED.** Arduino Uno comes with built-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON and LOW will turn it OFF.

**Vin.** It is the input voltage provided to the Arduino Board. It is different than 5 V supplied through a USB port. This pin is used to supply voltage. If a voltage is provided through power jack, it can be accessed through this pin.

**5V.** This board comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using three ways i.e. USB, Vin pin of the board or DC power jack.

USB supports voltage around 5V while Vin and Power Jack support a voltage ranges between 7V to 20V. It is recommended to operate the board on 5V. It is important to note that, if a voltage is supplied through 5V or 3.3V pins, they result in bypassing the voltage regulation that can damage the board if voltage surpasses from its limit.

**GND.** These are ground pins. More than one ground pins are provided on the board which can be used as per requirement.

**Reset.** This pin is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.

**IOREF.** This pin is very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then select the proper power source.

**PWM.** PWM is provided by 3, 5, 6,9,10, 11pins. These pins are configured to provide 8-bit output PWM.

**SPI.** It is known as Serial Peripheral Interface. Four pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI communication with the help of SPI library.

**AREF.** It is called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.

**TWI.** It is called Two-wire Interface. TWI communication is accessed through Wire Library. A4 and A5 pins are used for this purpose.

**Serial Communication.** Serial communication is carried out through two pins called Pin 0 (Rx) and Pin 1 (Tx).

Rx pin is used to receive data while Tx pin is used to transmit data.

**External Interrupts.** Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

### Arduino Uno Technical Specifications

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet) – 8 bit AVR family microcontroller |
| Operating Voltage | 5V |
| Recommended Input Voltage | 7-12V |
| Input Voltage Limits | 6-20V |
| Analog Input Pins | 6 (A0 – A5) |
| Digital I/O Pins | 14 (Out of which 6 provide PWM output) |
| DC Current on I/O Pins | 40 mA |
| DC Current on 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (0.5 KB is used for Bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Frequency (Clock Speed) | 16 MHz |

**Communication and Programming:**

Arduino Uno comes with an ability of interfacing with other other Arduino boards, microcontrollers and computer. The Atmega328 placed on the board provides serial communication using pins like Rx and Tx.

The Atmega16U2 incorporated on the board provides a pathway for serial communication using USB com drivers. Serial monitor is provided on the IDE software which is used to send or receive text data from the board. If LEDs placed on the Rx and Tx pins will flash, they indicate the transmission of data.

Arduino Uno is programmed using Arduino Software which a cross-platform application called IDE is written in Java. The AVR microcontroller Atmega328 laid out on the base comes with built-in boot loader that sets you free from using a separate burner to upload the program on the board.

****

**Applications:**

Arduino Uno comes with a wide range of applications. A larger number of people are using Arduino boards for developing sensors and instruments that are used in scientific research. Following are some main applications of the board.

* [Embedded System](https://www.theengineeringprojects.com/2016/10/what-is-embedded-systems.html)
* Security and Defense System
* Digital Electronics and Robotics
* Parking Lot Counter
* Weighing Machines
* Traffic Light Count Down Timer
* Medical Instrument
* Emergency Light for Railways
* Home Automation
* Industrial Automation

There are a lot of other microcontrollers available in the market that are more powerful and cheap as compared to Arduino board. So, why you prefer Arduino Uno?

Actually, Arduino comes with a big community that is developing and sharing the knowledge with a wide range of audience. Quick support is available pertaining to technical aspects of any electronic project. When you decide Arduino board over other controllers, you don’t need to arrange extra peripherals and devices as most of the functions are readily available on the board that makes your project economical in nature and free from a lot of technical expertise.

**GPS:**

Global Positioning System (GPS) is a satellite-based system that uses satellites and ground stations to measure and compute its position on Earth.

GPS is also known as Navigation System with Time and Ranging (NAVSTAR) GPS.

GPS receiver needs to receive data from at least 4 satellites for accuracy purpose. GPS receiver does not transmit any information to the satellites.

This GPS receiver is used in many applications like smartphones, Cabs, Fleet management etc.



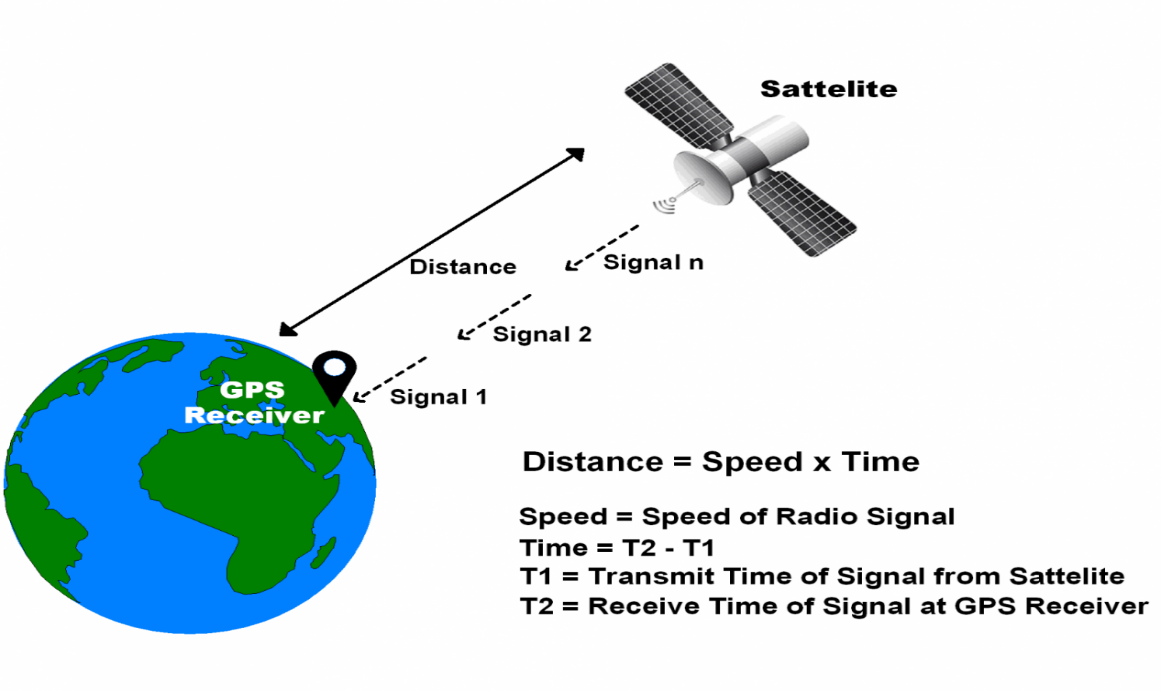
# **How GPS Works**

GPS receiver uses a constellation of satellites and ground stations to calculate accurate location wherever it is located.

These GPS satellites transmit information signal over radio frequency (1.1 to 1.5 GHz) to the receiver. With the help of this received information, a ground station or GPS module can compute its position and time.

# **How GPS Receiver Calculates its Position and Time**

GPS receiver receives information signals from GPS satellites and calculates its distance from satellites. This is done by measuring the time required for the signal to travel from satellite to the receiver.



**GPS Distance Calculation**

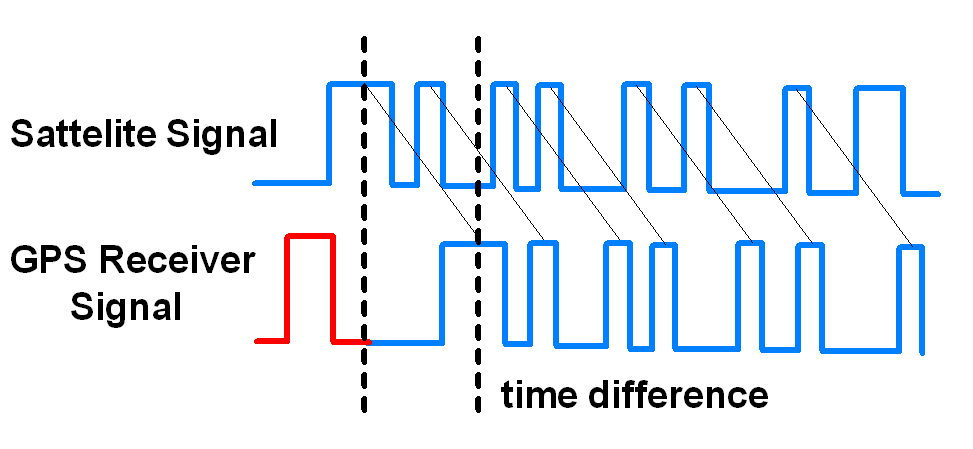
\textbf{Distance = Speed x Time}

Where,

Speed = Speed of Radio signal which is approximately equal to the speed of light i.e.3*10^{8}

Time = Time required for a signal to travel from the satellite to the receiver.

By subtracting the sent time from the received time, we can determine the travel time.



**GPS Signal Time Difference**

To determine distance, both the satellite and GPS receiver generate the same pseudocode signal at the same time.

The satellite transmits the pseudocode; which is received by the GPS receiver.

These two signals are compared and the difference between the signals is the travel time.

Now, if the receiver knows the distance from 3 or more satellites and their location (which is sent by the satellites), then it can calculate its location by using [Trilateration](http://electronics.howstuffworks.com/gadgets/travel/gps1.htm) method.

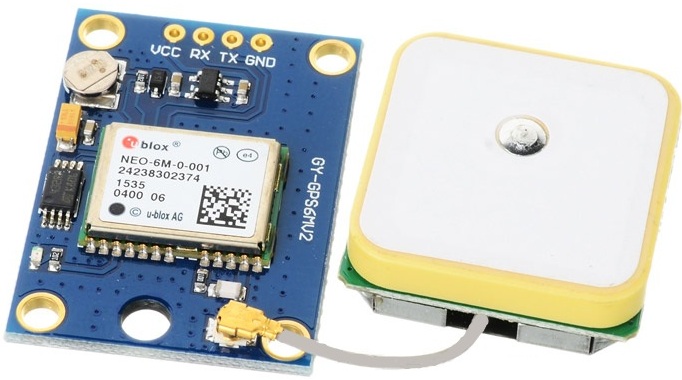
# [Trilateration](http://electronics.howstuffworks.com/gadgets/travel/gps1.htm): Trilateration is a mathematical technique used by a [global positioning system](https://www.lifewire.com/global-positioning-system-gps-1683311) (GPS) device to determine user position, speed, and elevation. By constantly receiving and analyzing radio signals from multiple GPS satellites and applying the geometry of circles, spheres, and triangles, a GPS device can calculate the precise distance or range to each satellite being tracked.

## How Trilateration Works

Trilateration is a sophisticated version of triangulation, though it does not use the measurement of angles in its calculations. Data from a single satellite provides a general location of a point within a large circular area on the Earth's surface. Adding data from a second satellite allows the GPS to narrow the specific location of that point down to a region where the two areas of satellite data overlap. Adding data from a third satellite provides an accurate position of the point on the Earth's surface.

All the GPS devices require three satellites for an accurate calculation of position. Data from a fourth satellite—or even more than four satellites—further enhance the precision of the point's location, and also allows factors such as elevation or, in the case of aircraft, altitude to also be calculated. GPS receivers routinely track four to seven satellites simultaneously and use trilateration to analyze the information.

# **GPS Module**



**GPS Receiver**

GPS receiver module gives output in standard (National Marine Electronics Association) NMEA string format. It provides output serially on Tx pin with default 9600 Baud rate.

This NMEA string output from GPS receiver contains different parameters separated by commas like longitude, latitude, altitude, time etc. Each string starts with ‘$’ and ends with carriage return/line feed sequence.

**E.g.**

$GPGGA,184237.000,1829.9639,N,07347.6174,E,1,05,2.1,607.1,M,-64.7,M,,0000\*7D

$GPGSA,A,3,15,25,18,26,12,,,,,,,,5.3,2.1,4.8\*36

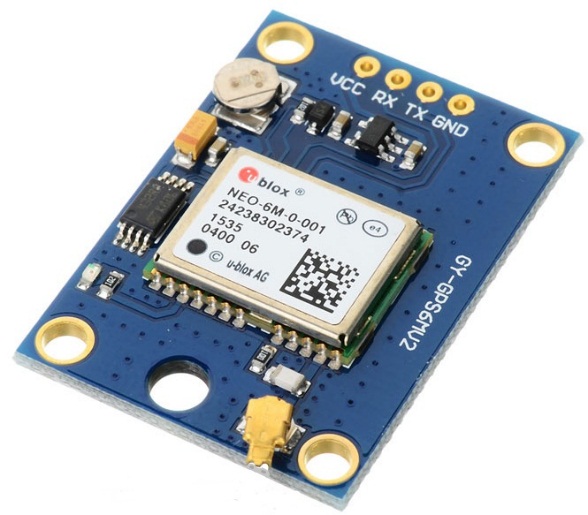
$GPGSV,3,1,11,15,47,133,46,25,44,226,45,18,37,238,45,26,34,087,40\*72

$GPGSV,3,2,11,12,27,184,45,24,02,164,26,29,58,349,,05,26,034,\*7F

$GPGSV,3,3,11,21,25,303,,02,11,071,,22,01,228,\*40

$GPRMC,184237.000,A,1829.9639,N,07347.6174,E,0.05,180.19,230514,,,A\*64

**Pin Description**



**GPS Receiver Module**

**VCC:** Power Supply 3.3 – 6 V

**GND:**Ground

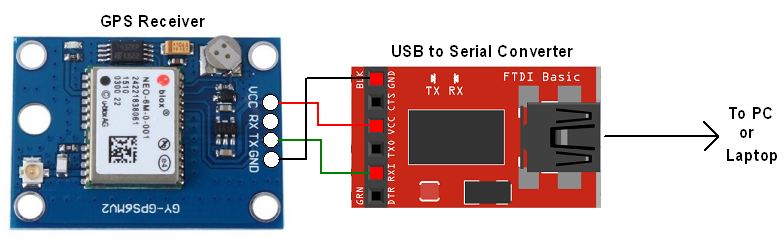
**TX:**Transmit data serially which gives information about location, time etc.

**RX:**Receive Data serially. It is required when we want to configure GPS module.

# **Check GPS module**

Before Interfacing GPS module with PIC18F4550 microcontroller, we can check the output of GPS module. From that string, we can extract information like longitude, latitude, time which is helpful to find location and timing information.

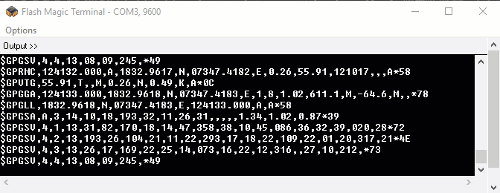
To do this, connect this GPS module to the PC via USB to Serial converter or DB9 connector. Also, it is necessary to keep antenna of GPS module on proper location.



**GPS Serial Interface**

1. Now open any serial terminal e.g. Realterm, Hyper terminal, Putty etc. on PC/laptop.
2. Open the PORT with 9600 baud rate.
3. The terminal will show data coming from GPS receiver module.

The output data from GPS receiver module displaying on a serial terminal as follows.



In the above string, the NMEA string starting with “$GPGGA” is most popularly used. It provides us Time, Longitude, Latitude and Altitude along with directions. This information is helpful to find Time and Location.

**RF MODULE:**

In generally, the wireless systems designer has two overriding constraints: it must operate over a certain distance and transfer a certain amount of information within a data rate. The RF modules are very small in dimension and have a wide operating voltage range i.e. 3V to 12V.

Basically the RF modules are 433 MHz RF transmitter and receiver modules. The transmitter draws no power when transmitting logic zero while fully suppressing the carrier frequency thus consume significantly low power in battery operation. When logic one is sent carrier is fully on to about 4.5mA with a 3volts power supply. The data is sent serially from the transmitter which is received by the tuned receiver. Transmitter and the receiver are duly interfaced to two microcontrollers for data transfer.

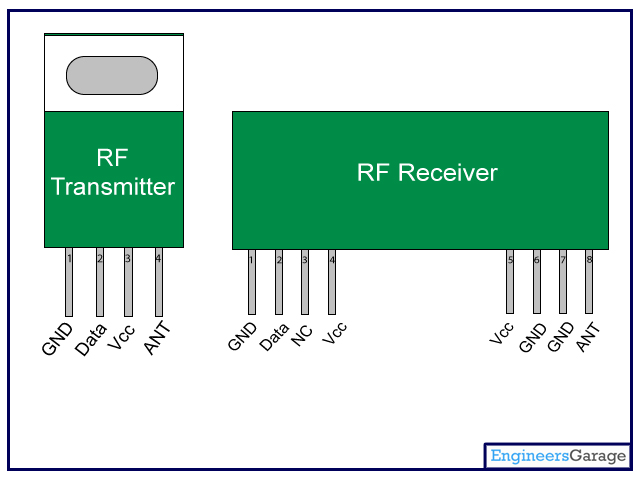
The RF module, as the name suggests, operates at Radio Frequency. The corresponding frequency range varies between 30 kHz & 300 GHz. In this RF system, the digital data is represented as variations in the amplitude of carrier wave. This kind of modulation is known as Amplitude Shift Keying (ASK).

Transmission through RF is better than IR (infrared) because of many reasons. Firstly, signals through RF can travel through larger distances making it suitable for long range applications. Also, while IR mostly operates in line-of-sight mode, RF signals can travel even when there is an obstruction between transmitter & receiver. Next, RF transmission is more strong and reliable than IR transmission. RF communication uses a specific frequency unlike IR signals which are affected by other IR emitting sources.

This **RF module** comprises of an **RF Transmitter** and an **RF Receiver**. The transmitter/receiver (Tx/Rx) pair operates at a frequency of **434 MHz**. An RF transmitter receives serial data and transmits it wirelessly through RF through its antenna connected at pin4. The transmission occurs at the rate of 1Kbps - 10Kbps.The transmitted data is received by an RF receiver operating at the same frequency as that of the transmitter.

The RF module is often used alongwith a pair of encoder/decoder. The encoder is used for encoding parallel data for transmission feed while reception is decoded by a decoder. [HT12E](http://www.engineersgarage.com/content/ht12e)-[HT12D](http://www.engineersgarage.com/content/ht12d), HT640-HT648, etc. are some commonly used encoder/decoder pair ICs.

**Pin Diagram:**



**Pin Description:**

**RF Transmitter**

|  |  |  |
| --- | --- | --- |
| **Pin No** | **Function** | **Name** |
| 1 | Ground (0V) | Ground |
| 2 | Serial data input pin | Data |
| 3 | Supply voltage; 5V | Vcc |
| 4 | Antenna output pin | ANT |

**RF Receiver**

|  |  |  |
| --- | --- | --- |
| **Pin No** | **Function** | **Name** |
| 1 | Ground (0V) | Ground |
| 2 | Serial data output pin | Data |
| 3 | Linear output pin; not connected | NC |
| 4 | Supply voltage; 5V | Vcc |
| 5 | Supply voltage; 5V | Vcc |
| 6 | Ground (0V) | Ground |
| 7 | Ground (0V) | Ground |
| 8 | Antenna input pin | ANT |

Features of RF Module:

* Receiver frequency 433MHz
* Receiver typical frequency 105Dbm
* Receiver supply current 3.5mA
* Low power consumption
* Receiver operating voltage 5v
* Transmitter frequency range 433.92MHz
* Transmitter supply voltage 3v~6v
* Transmitter output power 4v~12v

**GSM**

GSM is a mobile communication modem; it is stands for global system for mobile communication (GSM). The idea of GSM was developed at Bell Laboratories in 1970.  It is widely used mobile communication system in the world. GSM is an open and digital cellular technology used for transmitting mobile voice and data services operates at the 850MHz, 900MHz, 1800MHz and 1900MHz frequency bands.

GSM system was developed as a digital system using time division multiple access (TDMA) technique for communication purpose. A GSM digitizes and reduces the data, then sends it down through a channel with two different streams of client data, each in its own particular time slot. The digital system has an ability to carry 64 kbps to 120 Mbps of data rates.

There are various cell sizes in a GSM system such as macro, micro, pico and umbrella cells. Each cell varies as per the implementation domain. There are five different cell sizes in a GSM network macro, micro, pico and umbrella cells. The coverage area of each cell varies according to the implementation environment.

### Time Division Multiple Access

TDMA technique relies on assigning different time slots to each user on the same frequency. It can easily adapt to data transmission and voice communication and can carry 64kbps to 120Mbps of data rate.

### GSM Architecture

A GSM network consists of the following components:

* **A Mobile Station:**  It is the mobile phone which consists of the transceiver, the display and the processor and is controlled by a SIM card operating over the network.
* **Base Station Subsystem:** It acts as an interface between the mobile station and the network subsystem. It consists of the Base Transceiver Station which contains the radio transceivers and handles the protocols for communication with mobiles. It also consists of the Base Station Controller which controls the Base Transceiver station and acts as a interface between the mobile station and mobile switching centre.
* **Network Subsystem:** It provides the basic network connection to the mobile stations. The basic part of the Network Subsystem is the Mobile Service Switching Centre which provides access to different networks like ISDN, PSTN etc. It also consists of the Home Location Register and the Visitor Location Register which provides the call routing and roaming capabilities of GSM. It also contains the Equipment Identity Register which maintains an account of all the mobile equipments wherein each mobile is identified by its own IMEI number. IMEI stands for International Mobile Equipment Identity.

### Features of GSM Module:

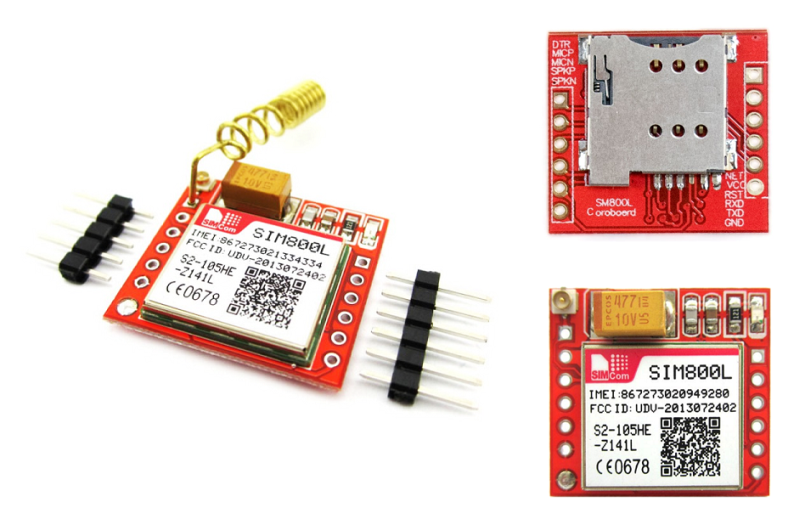
* Improved spectrum efficiency
* International roaming
* Compatibility with integrated services digital network (ISDN)
* Support for new services.
* SIM phonebook management
* Fixed dialing number (FDN)
* Real time clock with alarm management
* High-quality speech
* Uses encryption to make phone calls more secure
* Short message service (SMS)

The security strategies standardized for the GSM system make it the most secure telecommunications standard currently accessible. Although the confidentiality of a call and secrecy of the GSM subscriber is just ensured on the radio channel, this is a major step in achieving end-to- end security.

### GSM Modem

A GSM modem is a device which can be either a mobile phone or a modem device which can be used to make a computer or any other processor communicate over a network. A GSM modem requires a SIM card to be operated and operates over a network range subscribed by the network operator.  It can be connected to a computer through serial, USB or Bluetooth connection.

A GSM modem can also be a standard GSM mobile phone with the appropriate cable and software driver to connect to a serial port or USB port on your computer. GSM modem is usually preferable to a GSM mobile phone. The GSM modem has wide range of applications in transaction terminals, supply chain management, security applications, weather stations and GPRS mode remote data logging.

****

It requires a **SIM (Subscriber Identity Module)** card just like mobile phones to activate communication with the network. Also they have **IMEI** (International Mobile Equipment Identity) number similar to mobile phones for their identification. A GSM/GPRS MODEM can perform the following operations:

1.      Receive, send or delete SMS messages in a SIM.

2.      Read, add, search phonebook entries of the SIM.

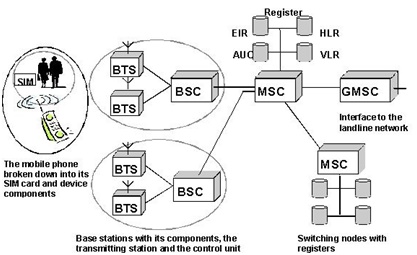
3.      Make, Receive, or reject a voice call.

The MODEM needs **AT commands**, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the **GSM and GPRS cellular network**.

### **GSM Architecture**

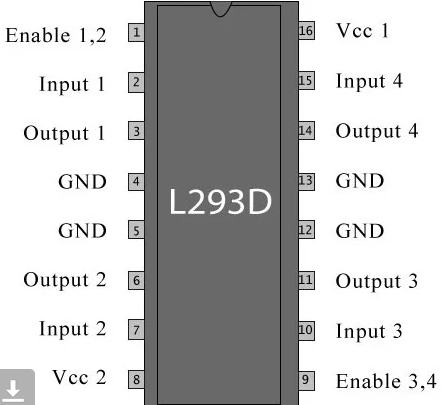
The GSM architecture is divided into Radio Subsystem, Network and Switching Subsystem and the Operation Subsystem. The radio sub system consists of the Mobile Station and Base Station Subsystem.

The mobile station is generally the mobile phone which consists of a transceiver, display and a processor. Each handheld or portable mobile station consists of a unique identity stored in a module known as SIM (Subscriber Identity Chip). It is a small microchip which is inserted in the mobile phone and contains the database regarding the mobile station.



**Motor Driver:**

A motor driver is an integrated circuit chip which is usually used to control motors in autonomous robots. Motor driver act as an interface between Arduino and the motors. The most commonly used motor driver IC’s are from the L293 series such as L293D, L293NE, etc. These ICs are designed to control 2 DC motors simultaneously. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor. We will be referring the motor driver IC as L293D only. L293D has 16 pins.



The L293D is a 16 pin IC, with eight pins, on each side, dedicated to the controlling of a motor. There are 2 INPUT pins, 2 OUTPUT pins and 1 ENABLE pin for each motor. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor.

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Enable 1,2 | This pin enables the input pin Input 1(2) and Input 2(7) |
| 2 | Input 1 | Directly controls the Output 1 pin. Controlled by digital circuits |
| 3 | Output 1 | Connected to one end of  Motor 1 |
| 4 | Ground | Ground pins are connected to ground of circuit (0V) |
| 5 | Ground | Ground pins are connected to ground of circuit (0V) |
| 6 | Output 2 | Connected to another end of  Motor 1 |
| 7 | Input 2 | Directly controls the Output 2 pin. Controlled by digital circuits |
| 8 | Vcc2 (Vs) | Connected to Voltage pin for running motors (4.5V to 36V) |
| 9 | Enable 3,4 | This pin enables the input pin Input 3(10) and Input 4(15) |
| 10 | Input 3 | Directly controls the Output 3 pin. Controlled by digital circuits |
| 11 | Output 3 | Connected to one end of Motor 2 |
| 12 | Ground | Ground pins are connected to ground of circuit (0V) |
| 13 | Ground | Ground pins are connected to ground of circuit (0V) |
| 14 | Output 4 | Connected to another end of Motor 2 |
| 15 | Input 4 | Directly controls the Output 4 pin. Controlled by digital circuits |
| 16 | Vcc2 (Vss) | Connected to +5V to enable IC function |

**Working of L293D**

There are 4 input pins for l293d, pin 2, 7 on the left and pin 15, 10 on the right as shown on the pin diagram. Left input pins will regulate the rotation of motor connected across left side and right input for motor on the right hand side. The motors are rotated on the basis of the inputs provided across the input pins as LOGIC 0 or LOGIC 1.

In simple you need to provide Logic 0 or 1 across the input pins for rotating the motor.

**L293D Logic Table.**

Let’s consider a Motor connected on left side output pins (pin 3, 6). For rotating the motor in clockwise direction the input pins has to be provided with Logic 1 and Logic 0.

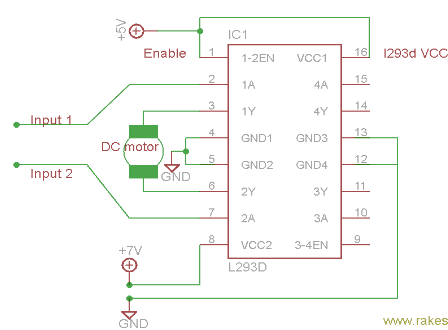
Pin 2 = Logic 1 and Pin 7 = Logic 0 | Clockwise Direction

Pin 2 = Logic 0 and Pin 7 = Logic 1 | Anticlockwise Direction

**Pin 2 = Logic 0**and**Pin 7 = Logic 0** | Idle [No rotation] [Hi-Impedance state]

**Pin 2 = Logic 1**and**Pin 7 = Logic 1** | Idle [No rotation]

**Circuit Diagram for l293d motor driver IC controller**



## Voltage Specification

VCC is the voltage that it needs for its own internal operation 5v; L293D will not use this voltage for driving the motor. For driving the motors it has a separate provision to provide motor supply VSS (V supply).  L293d will use this to drive the motor. It means if you want to operate a motor at 9V then you need to provide a Supply of 9V across VSS Motor supply.

The maximum voltage for VSS motor supply is 36V. It can supply a max current of 600mA per channel. Since it can drive motors Up to 36v hence you can drive pretty big motors with this l293d. VCC pin 16 is the voltage for its own internal Operation. The maximum voltage ranges from 5v and up to 36v.

## What is a DC Motor?

## A direct current (DC) motor is a type of electric machine that converts electrical energy into mechanical energy. DC motors take electrical power through direct current, and convert this energy into mechanical rotation.

## DC motors use magnetic fields that occur from the electrical currents generated, which powers the movement of a rotor fixed within the output shaft. The output torque and speed depends upon both the electrical input and the design of the motor.

## How DC motors work

## The term ‘DC motor’ is used to refer to any rotary electrical machine that converts direct current electrical energy into mechanical energy. DC motors can vary in size and power from small motors in toys and appliances to large mechanisms that power vehicles, pull elevators and hoists, and drive steel rolling mills.

## But how do DC motors work?

## DC motors include two key components: a **stator** and an **armature**. The stator is the stationary part of a motor, while the armature rotates. In a DC motor, the stator provides a rotating magnetic field that drives the armature to rotate.

## A simple DC motor uses a stationary set of magnets in the stator, and a coil of wire with a current running through it to generate an electromagnetic field aligned with the centre of the coil. One or more windings of insulated wire are wrapped around the core of the motor to concentrate the magnetic field.

## The windings of insulated wire are connected to a commutator (a rotary electrical switch), that applies an electrical current to the windings. The commutator allows each armature coil to be energised in turn, creating a steady rotating force (known as torque).

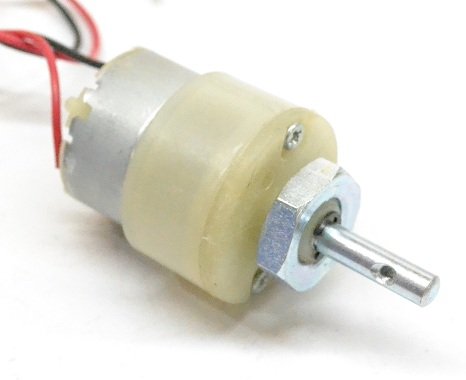
## When the coils are turned on and off in sequence, a rotating magnetic field is created that interacts with the differing fields of the stationary magnets in the stator to create torque, which causes it to rotate. These key operating principles of DC motors allow them to convert the electrical energy from direct current into mechanical energy through the rotating movement, which can then be used for the propulsion of objects.

## Who invented the DC motor?

## This amazing piece of electrical equipment has revolutionised our lives in many ways, but who invented the DC motor? As with all major innovations, there are many people who had a role to play through the development of similar mechanisms.

## In the US, Thomas Davenport is widely celebrated as the inventor of the first electric motor, and undoubtedly he was the first to patent a useable electric motor in 1837. Davenport, however, was not the first person to build an electric motor, with various inventors in Europe having already developed more powerful versions by the time Davenport filed his patent.

## In 1834, Moritz Jacobi had presented a motor that was three times as powerful as the one Davenport would later patent, while Sibrandus Stratingh and Christopher Becker were the first to demonstrate a practical application for an electric motor, by running a small model car in 1835.



The first practical DC motor was invented some years later in 1886 by Frank Julian Sprague, whose invention lead to the first motor powered trolley system in 1887, and the first electric elevator in 1892. Sprague’s DC motor was a hugely significant development, leading to a variety of applications which would reshape the face of industry and manufacturing.

## Types of DC Motors

## So far, this guide has broadly explained how DC motors work, the history of these mechanisms, and what they look like. While the principles are the same across variants, there are actually several different types of DC motors, which offer specific advantages and disadvantages over each other.

**Buzzer:**

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. Buzzer is an integrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys, automotive electronic equipment, telephones, timers and other electronic products for sound devices. Active buzzer 5V Rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board in combination, can complete a simple circuit design, to "plug and play.



**Buzzer Pin Configuration**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Positive | Identified by (+) symbol or longer terminal lead. Can be powered by 5V DC |
| 2 | Negative | Identified by short terminal lead. Typically connected to the ground of the circuit |

**Buzzer Features and Specifications**

* Rated Voltage: 6V DC
* Operating Voltage: 4-8V DC
* Rated current: <30mA
* Sound Type: Continuous Beep
* Resonant Frequency: ~2300 Hz
* Small and neat sealed package
* Breadboard and Perf board friendly

**How to use a Buzzer**

A **buzzer**is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on [breadboard](https://components101.com/misc/breadboard-connections-uses-guide), Perf Board and even on PCBs which makes this a widely used component in most electronic applications.

There are two types are buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beeeeeeppp.... sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep. Beep. Beep. Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customized with help of other circuits to fit easily in our application.

This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at required time and require interval.

**Applications of Buzzer**

* Alarming Circuits, where the user has to be alarmed about something
* Communication equipment’s
* Automobile electronics
* Portable equipment’s, due to its compact size

**Battery:**

A rechargeable battery is an energy storage device that can be charged again after being discharged by applying [DC](https://whatis.techtarget.com/definition/DC-direct-current) current to its terminals.

Rechargeable [batteries](https://searchmobilecomputing.techtarget.com/definition/battery) allow for multiple usages from a cell, reducing waste and generally providing a better long-term investment in terms of dollars spent for usable device time. This is true even factoring in the higher purchase price of rechargeable and the requirement for a charger.  
A rechargeable battery is generally a more sensible and sustainable replacement to one-time use batteries, which generate current through a chemical reaction in which a reactive anode is consumed. The anode in a rechargeable battery gets consumed as well but at a slower rate, allowing for many charges and discharges.

In use, rechargeable batteries are the same as conventional ones. However, after discharge the batteries are placed in a charger or, in the case of built-in batteries, an [AC](https://whatis.techtarget.com/definition/alternating-current-AC)/DC adapter is connected.

While rechargeable batteries offer better long term cost and reduce waste, they do have a few cons. Many types of rechargeable cells created for consumer devices, including AA and AAA, C and D batteries, produce a lower voltage of 1.2v in contrast to the 1.5v of alkaline batteries. Though this lower voltage doesn't prevent correct operation in properly-designed electronics, it can mean a single charge does not last as long or offer the same power in a session. This is not the case, however, with lithium polymer and [lithium ion batteries](https://searchmobilecomputing.techtarget.com/definition/Lithium-Ion-battery).

Some types of batteries such as [nickel cadmium](https://searchmobilecomputing.techtarget.com/definition/Nickel-Cadmium-battery) and [nickel-metal hydride](https://searchmobilecomputing.techtarget.com/definition/Nickel-Metal-Hydride-battery) can develop a [battery memory effect](https://whatis.techtarget.com/definition/battery-memory-effect) when only partially discharged, reducing performance of subsequent charges and thus [battery life](https://whatis.techtarget.com/definition/battery-life) in a given device.

Rechargeable batteries are used in many applications such as cars, all manner of consumer electronics and even off-grid and supplemental facility power storage.

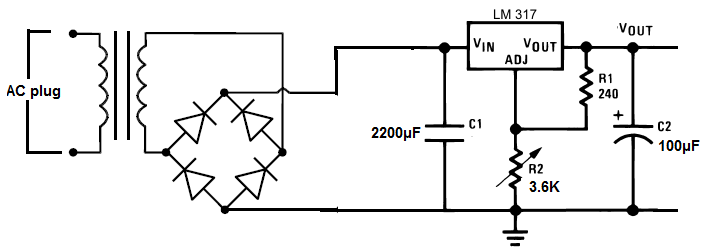
****

Power supply:

A power supply is a component that provides at least one electrical charge with power. It typically converts one type of electrical power to another, but it can also convert a different Energy form in electrical energy, such as solar, mechanical, or chemical.

A power supply provides electrical power to components. Usually the term refers to devices built into the powered component. Computer power supplies, for example, convert AC current to DC current and are generally located along with at least one fan at the back of the computer case.

Most computer power supplies also have an input voltage switch that, depending on the geographic location, can be set to 110v/115v or 220v/240v. Due to the different power voltages supplied by power outlets in different countries, this switch position is crucial.

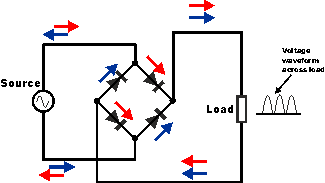


Some basic components used in the supply of power:

**Rectifier:**

A **rectifier** is an electrical device that [converts](https://en.wikipedia.org/wiki/Electric_power_conversion) [alternating current](https://en.wikipedia.org/wiki/Alternating_current) (AC), which periodically reverses direction, to [direct current](https://en.wikipedia.org/wiki/Direct_current) (DC), which flows in only one direction. The process is known as *rectification*, since it "straightens" the direction of current.

Rectifiers have many uses, but are often found to serve as components of DC power supplies and direct power transmission systems with high voltage. Rectification can be used in roles other than direct current generation for use as a power source.

****

**Circuit of rectifier**

****

**Rectifier**

**Capacitors:**

Capacitors are used to attain from the connector the immaculate and smoothest DC voltage in which the rectifier is used to obtain throbbing DC voltage which is used as part of the light of the present identity. Capacitors are used to acquire square DC from the current AC experience of the current channels so that they can be used as a touch of parallel yield.

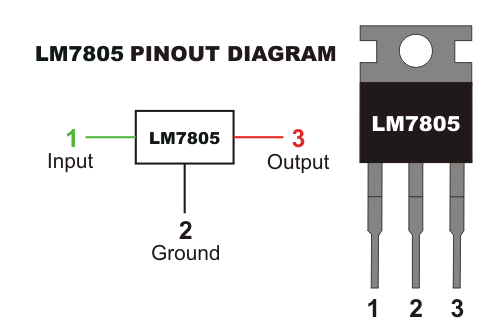


**Capacitor**

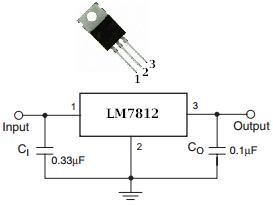
**Voltage regulators:**

The 78XX voltage controller is mainly used for voltage controllers as a whole. The XX speaks to the voltage delivered to the specific gadget by the voltage controller as the yield. 7805 will supply and control 5v yield voltage and 12v yield voltage will be created by 7812.

The voltage controllers are that their yield voltage as information requires no less than 2 volts. For example, 7805 as sources of information will require no less than 7V, and 7812, no less than 14 volts. This voltage is called Dropout Voltage, which should be given to voltage controllers.



**7805 voltage regulator with pinout**



**7812 voltage regulator with pinout**

**SOFTWARE REQUIREMENTS**

**Arduino IDE:**

**Arduino IDE**where IDE stands for Integrated Development Environment – An official software introduced by Arduino.cc, that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go.

**Introduction to Arduino IDE:**

* Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
* It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
* It is easily available for operating systems like MAC, Windows, and Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.
* A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, [Arduino Micro](https://www.theengineeringprojects.com/2018/09/introduction-to-arduino-micro.html) and many more.
* Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
* The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
* The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.
* This environment supports both C and C++ languages.

**How to install Arduino IDE:**

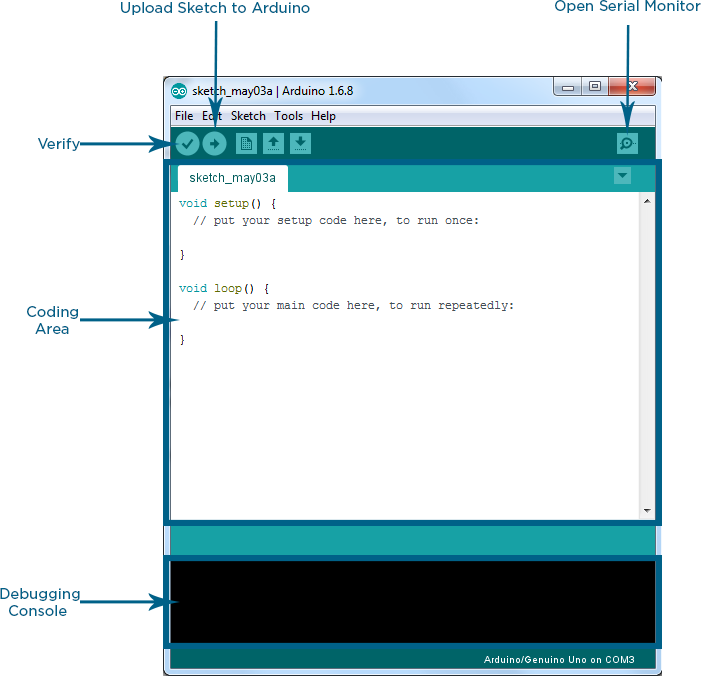
You can download the Software from [Arduino](https://www.arduino.cc/en/Main/Software) main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.

* If you aim to download Windows app version, make sure you have Windows 8.1 or Windows 10, as app version is not compatible with Windows 7 or older version of this operating system.

The IDE environment is mainly distributed into three sections

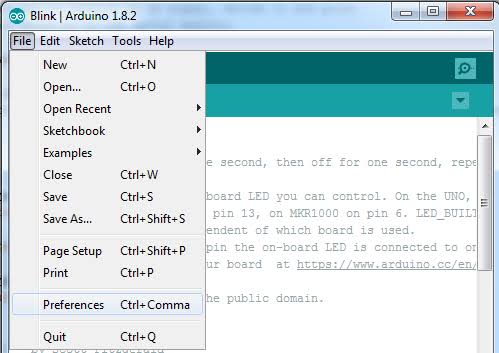
* **1. Menu Bar**
* **2. Text Editor**
* **3. Output Pane**

As you download and open the IDE software, it will appear like an image below.



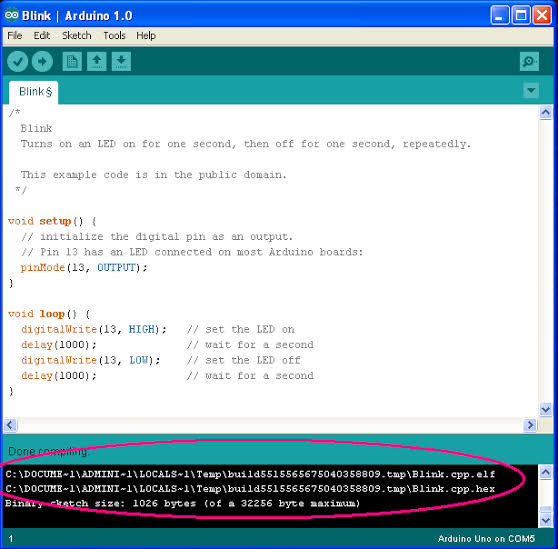
The bar appearing on the top is called **Menu Bar** that comes with five different options as follow

* **File** – You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into.

****

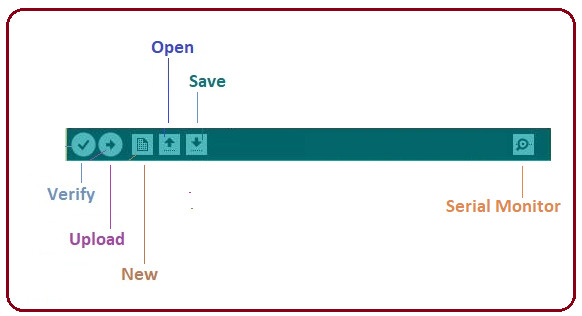
As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.

And at the end of compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.

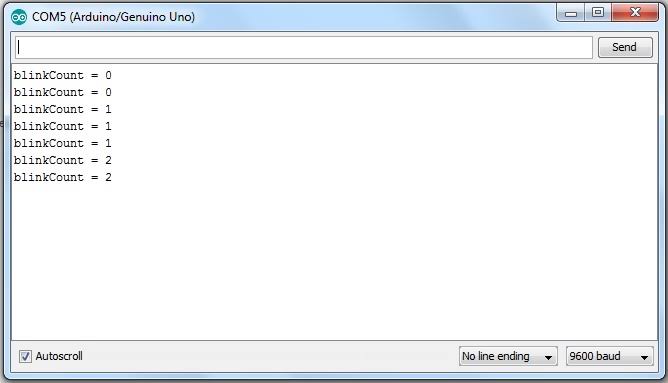
****

* **Edit** – Used for copying and pasting the code with further modification for font
* **Sketch** – For compiling and programming
* **Tools** – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
* **Help** – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

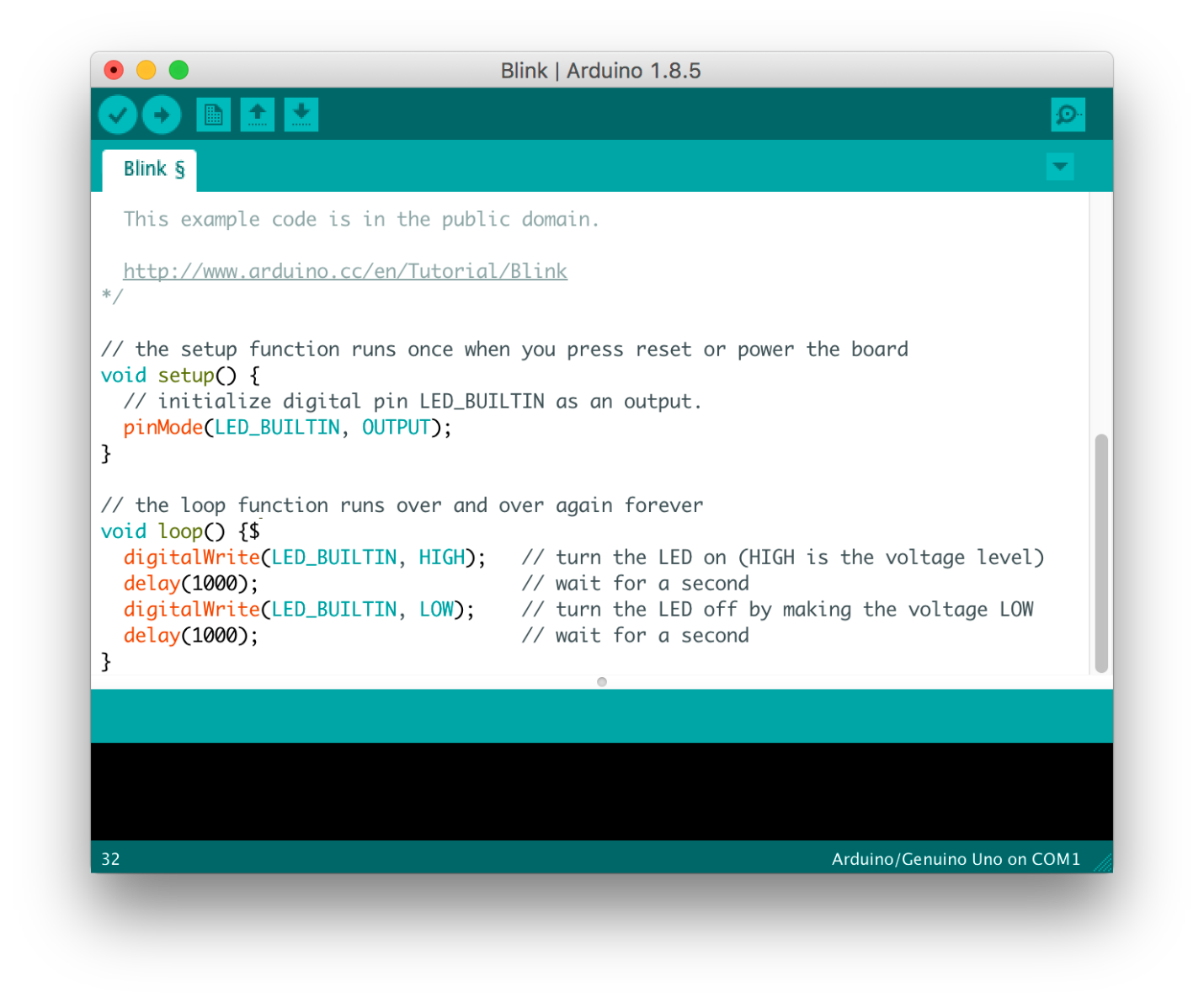
The **Six Buttons** appearing under the Menu tab are connected with the running program as follow.

****

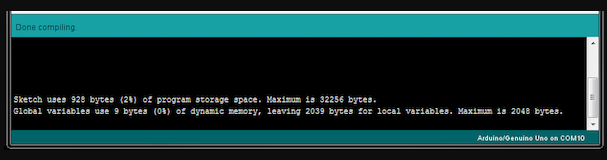
* The check mark appearing in the circular button is used to verify the code. Click this once you have written your code.
* The arrow key will upload and transfer the required code to the Arduino board.
* The dotted paper is used for creating a new file.
* The upward arrow is reserved for opening an existing Arduino project.
* The downward arrow is used to save the current running code.
* The button appearing on the top right corner is a **Serial Monitor** – A separate pop-up window that acts as an independent terminal and plays a vital role for sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.
* You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor, the output will show as the image below.



The main screen below the Menu bard is known as a simple text editor used for writing the required code.



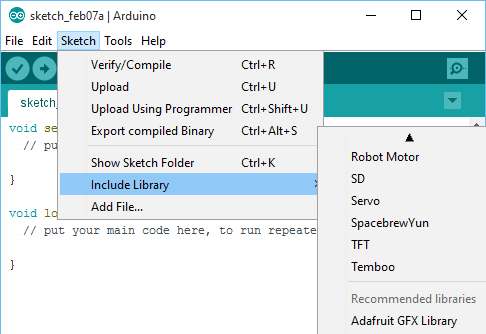
The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module.



More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller, however, there are some dedicated libraries used for calling and executing specific functions on the board.

**Libraries:**

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library.



As you click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, it will appear on the text editor as

#include <EEPROM.h>.

Most of the libraries are preinstalled and come with the Arduino software. However, you can also download them from the external sources.

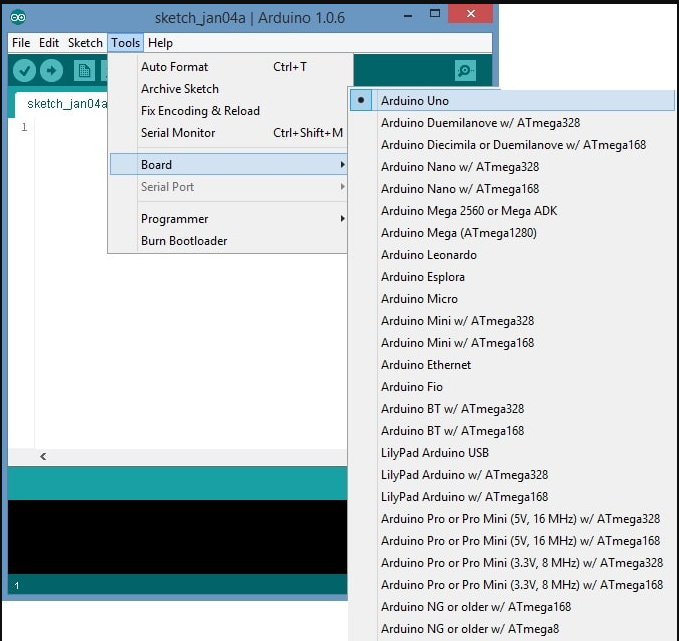
**Making pins Input and output:**

The digitalRead and [digitalWrite](https://www.theengineeringprojects.com/2018/09/how-to-use-digitalwrite-arduino-command.html) commands are used for addressing and making the Arduino pins as an input and output respectively.

These commands are text sensitive i.e. you need to write them down the exact way they are given like digitalWrite starting with small “d” and write with capital “W”. Writing it down with Digitalwrite or digitalwrite won’t be calling or addressing any function.

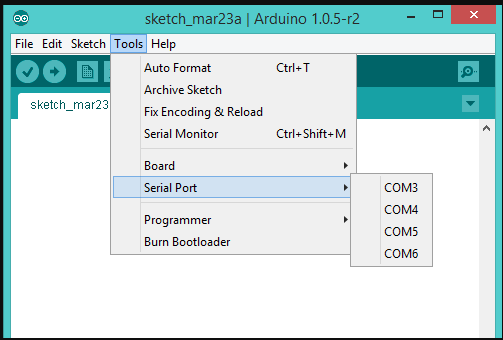
**How to select the board:**

In order to upload the sketch, you need to select the relevant board you are using and the ports for that operating system. As you click the Tools on the Menu, it will open like the figure below.



* Just go to the “Board” section and select the board you aim to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USB board. You can look for the USB serial device in the ports section of the Windows Device Manager.

Following figure shows the COM4 that I have used for my project, indicating the Arduino Uno with COM4 port at the right bottom corner of the screen.



* After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six button section or you can go to the Sketch section and press verify/compile and then upload.
* The sketch is written in the text editor and is then saved with the file extension .ino.

It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, older version may require the physical reset on the board.

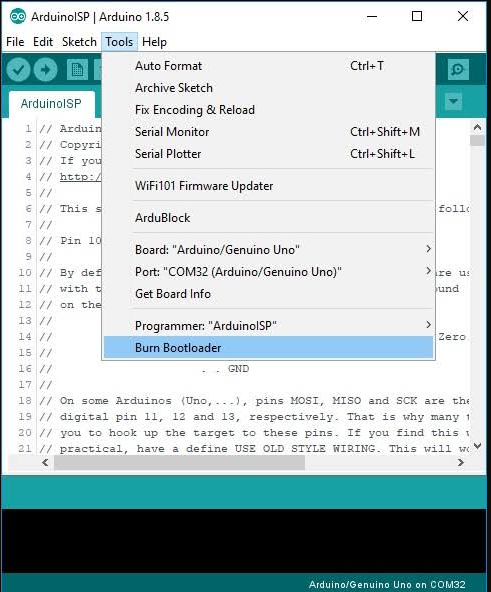
* Once you upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.

**Note**: The port selection criteria mentioned above is dedicated for Windows operating system only, you can check this [Guide](https://www.arduino.cc/en/Guide/Environment) if you are using MAC or Linux.

* The amazing thing about this software is that no prior arrangement or bulk of mess is required to install this software, you will be writing your first program within 2 minutes after the installation of the IDE environment.

**BootLoader:**

As you go to the Tools section, you will find a bootloader at the end. It is very helpful to burn the code directly into the controller, setting you free from buying the external burner to burn the required code.



When you buy the new Arduino Module, the bootloader is already installed inside the controller. However, if you intend to buy a controller and put in the Arduino module, you need to burn the bootloader again inside the controller by going to the Tools section and selecting the burn bootloader.

**MAIN CODE:**

#include <TinyGPS++.h>

TinyGPSPlus gps;

String LAT = "13.630108";

String LON = "79.408439";

#define motor\_pin1 4

#define motor\_pin2 5

#define motor\_pin3 6

#define motor\_pin4 7

#define Switch1 9

#define Switch2 10

#define Switch3 11

#define Switch4 12

#define buzzer 2

void setup()

{

Serial.begin(9600);

pinMode(Switch1, INPUT\_PULLUP);

pinMode(Switch2, INPUT\_PULLUP);

pinMode(Switch3, INPUT\_PULLUP);

pinMode(Switch4, INPUT\_PULLUP);

pinMode(motor\_pin1, OUTPUT);

pinMode(motor\_pin2, OUTPUT);

pinMode(motor\_pin3, OUTPUT);

pinMode(motor\_pin4, OUTPUT);

pinMode(buzzer, OUTPUT);

digitalWrite(motor\_pin1, LOW);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, LOW);

digitalWrite(buzzer, LOW);

}

void loop()

{

smartDelay(1000);

String LAT = String(gps.location.lat(), 6);

String LON = String(gps.location.lng(), 6);

Serial.print("LATITUDE = ");

Serial.println(LAT);

delay(1000);

Serial.print("LONGITUDE = ");

Serial.println(LON);

delay(1000);

if (millis() > 5000 && gps.charsProcessed() < 10)

Serial.println(F("No GPS data received: check wiring"));

int Switchstate1 = digitalRead(Switch1);

Serial.print("Switchstate1:");

Serial.println(Switchstate1);

int Switchstate2 = digitalRead(Switch2);

Serial.print("Switchstate2:");

Serial.println(Switchstate2);

int Switchstate3 = digitalRead(Switch3);

Serial.print("Switchstate3:");

Serial.println(Switchstate3);

int Switchstate4 = digitalRead(Switch4);

Serial.print("Switchstate4:");

Serial.println(Switchstate4);

if(Switchstate1 == 0)

{

Serial.println("forward");

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, HIGH);

digitalWrite(motor\_pin4, LOW);

if(LAT >"13.630104" || LON >"79.408432")

{

Serial.println("stop");

digitalWrite(buzzer, HIGH);

Serial.println("BACK");

digitalWrite(buzzer, HIGH);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, HIGH);

delay(3000);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, HIGH);

digitalWrite(motor\_pin4, LOW);

delay(5000);

digitalWrite(motor\_pin1, LOW);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, LOW);

send\_msg();

delay(8000);

}

}

else if (Switchstate2 == 0)

{

Serial.println("backward");

digitalWrite(motor\_pin1, LOW);

digitalWrite(motor\_pin2, HIGH);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, HIGH);

if(LAT >"13.630104" || LON >"79.408432")

{

Serial.println("stop");

digitalWrite(buzzer, HIGH);

Serial.println("BACK");

digitalWrite(buzzer, HIGH);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, HIGH);

delay(3000);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, HIGH);

digitalWrite(motor\_pin4, LOW);

delay(5000);

digitalWrite(motor\_pin1, LOW);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, LOW);

send\_msg();

delay(8000);

}

}

else if (Switchstate3 == 0)

{

Serial.println("right");

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, LOW);

delay(1000);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, HIGH);

digitalWrite(motor\_pin4, LOW);

if(LAT >"13.630104" || LON >"79.408432")

{

Serial.println("stop");

digitalWrite(buzzer, HIGH);

Serial.println("BACK");

digitalWrite(buzzer, HIGH);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, HIGH);

delay(3000);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, HIGH);

digitalWrite(motor\_pin4, LOW);

delay(5000);

digitalWrite(motor\_pin1, LOW);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, LOW);

send\_msg();

delay(8000);

}

}

else if (Switchstate4 == 0)

{

Serial.println("left");

digitalWrite(motor\_pin1, LOW);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, HIGH);

digitalWrite(motor\_pin4, LOW);

delay(1000);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, HIGH);

digitalWrite(motor\_pin4, LOW);

if(LAT >"13.630104" || LON >"79.408432")

{

Serial.println("BACK");

digitalWrite(buzzer, HIGH);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, HIGH);

delay(3000);

digitalWrite(motor\_pin1, HIGH);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, HIGH);

digitalWrite(motor\_pin4, LOW);

delay(5000);

digitalWrite(motor\_pin1, LOW);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, LOW);

send\_msg();

delay(8000);

}

}

else

{

digitalWrite(buzzer, LOW);

digitalWrite(motor\_pin1, LOW);

digitalWrite(motor\_pin2, LOW);

digitalWrite(motor\_pin3, LOW);

digitalWrite(motor\_pin4, LOW);

}

}

void send\_msg()

{

Serial.println("MESSAGE SEND");

Serial.println("AT+CMGF=1");

delay(1000);

Serial.println("AT+CMGS=\"+917286819827\"\r");

delay(1000);

Serial.println("BORDER CROSSED ALERT");

delay(1000);

Serial.print("https://maps.google.com/maps?q=");

Serial.print(gps.location.lat(), 6);

Serial.print(",");

Serial.print(gps.location.lng(), 6);

delay(1000);

Serial.println((char)26);

delay(1000);

}

static void smartDelay(unsigned long ms)

{

unsigned long start = millis();

do

{

while (Serial.available())

gps.encode(Serial.read());

} while (millis() - start < ms);

}

**Advantages:**

* Find the live location of fisherman
* Give early alerts at border crossing time
* Low cost

**Applications:**

* Boats
* Army
* Navy boats

**CONCLUSION**

Thus the fishermen can easily identify the national sea borders and therefore prevents them from entering their area. Thus saving their lives and providing good relationship with the neighboring countries. Also, the piracy of ship can be easily brought under control.

**REFERENCES**

[1] K. Suresh Kumar et. Al, “Design of low cost maritime boundary identification device using GPS system”, International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4665-4672.

[2] M Sivaramaganesh, International journal of innovative research in electrical, electronics, instrumentation and control engineering vol. 2, issue 3, March 2014

[3] http://www.thehindu.com/multimedia/dynamic/01689/TH\_ 09\_GROWING\_rev\_\_1689954g

[4] S. Mani Sunder, “Deep sea fishermen patrol system for coastal intruder positioning” Scientific Engineering and Technology (ISSN : 1581)Volume 2 Issue 3, PP : 129

[5] P.Satheesh, “Maritime Border Refuge System [MBR]”, National Conference on Emerging Trends in Computer, Communication & Instrumentation in Strength Security.