Fingerprint Based Bank Locker Security System

ABSTRACT

Many of the systems available today, allowing the entrance only to those persons who know a specific code, own a card or have determined physic marks. The more complex is the system, the most difficult is to be attacked, although it will be more expensive and will require more software and hardware resources. The password method is the cheapest and simplest technology, because it only requires elementary software resources. On the other hand, this system is easily attackable, since it is quite simple to obtain the password data from a person. The Smart Cards based systems are very useful, but used like the only identification system, are not excessively trustworthy, since cards can be easily stolen, lost or simply forgotten at home. Our system is going to use fingerprint from pc as password for authentication. The advantage of biometrics is that the information is unique for each individual and that it can identify the individual in spite of variations in the time (it does not matter if the first biometric sample was taken year ago). The advantage of our system is that, it not only collects single fingerprint from user as password. Instead our system will randomly collect few fingerprints from either hands of the user and gives authentication. Since the biometric data cannot be stolen, the system would be safe and secured.

**CHAPTER 1**

**EMBEDDED SYSTEMS**

## 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 [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. The applications software on such processors is sometimes referred to as firmware. The simplest devices consist of a single microprocessor (often called a "chip”), which may itself be packaged with other chips in a hybrid system or Application Specific Integrated Circuit (ASIC). Its input comes from a detector or sensor and its output goes to a switch or activator which (for example) may start or stop the operation of a machine or, by operating a valve, may control the flow of fuel to an engine. As the embedded system is the combination of both software and hardware.

## 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...} 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.

## EXAMPLES OF AN EMBEDDED SYSTEMS

Embedded systems are found in wide range of application areas. Originally they were used only for expensive industrial control applications, but as technology brought down the cost of dedicated processors, they began to appear in moderately expensive applications such as automobiles, communication and office equipments and television Today's embedded systems are so inexpensive that they are used in almost every electronic product in our life. Embedded systems are often designed for mass production.

* + 1. Automatic Teller Machines
    2. Cellular telephone and telephone switches
    3. Computer network equipment
    4. Computer printers
    5. Disk drives
    6. Home automation products
    7. Handheld calculators
    8. Household appliances
    9. Medical equipment
    10. Measurement equipment
    11. Multifunction wrist watches
    12. Multifunction printers

## HISTORY OF AN EMBEDDED SYSTEMS

The first recognizably modern embedded systems was the Apollo Guidance Computer, developed by “Charles Stark Draper” at the MIT Instrumentation Laboratory. At the project inception, the Apollo guidance computer was considered the riskiest item in the Apollo project as it employed the then newly developed monolithic integrated circuits to reduce the size and weight. An early mass produced embedded system was the Autonetics D-17 guidance computer for the Minuteman missile, released in 1961.It was built from transistor logic and had a hard disk for main memory .When the Minuteman II went into production in 1966, the D-17 was replaced with a new computer that was the first high volume use of integrated circuits. This program alone reduced prices on quad and gate ICs from $1000/ each to $3/ each permitting their use in commercial products.

Since these early applications in the 1960s, embedded systems have come down in price and there has been a dramatic rise in processing power and functionality. The first microprocessor for example, the Intel 4004, was designed for calculators and other small systems but still required many external memory and support chips. In 1978 National Engineering Manufactures Association released a “standard” for programmable microcontrollers, including almost any computer based controllers, such as single board computers, numerical and event based controllers. Embedded Systems are designed to some specific task, rather than be a general-purpose computer for multitasks. Some also have real-time performances constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirement, allowing the system hardware to be simplified to reduce cost.

## FEATURES OF AN EMBEDDED SYSTEM

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

## CHARACTERSTICS OF AN EMBEDDED SYSTEM

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 custom-built hardware built around a Central Processing Unit (CPU). This hardware also contains memory chips onto which the software is loaded. The software residing on the memory chip is also called the ‘firmware’. The embedded system architecture can be represented as a layered architecture as shown in Fig.

The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system. For small appliances such as remote control units, air conditioners, toys etc., there is no need foran operating system and you can write only the software specific to that application.

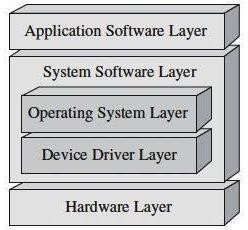


Fig 1.1 Layered Architecture of Embedded System

For applications involving complex processing, it is advisable to have an operating system. In such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip. Once the software is transferred to the memory chip, the software will continue to run *for* a long time you don’t need to reload new software.

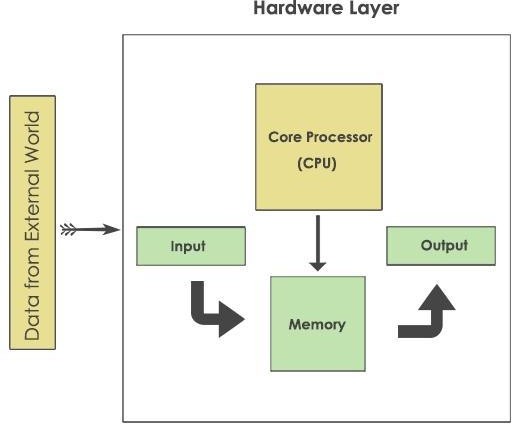


Fig 1.2 Block diagram of Embedded systems

Now, let us see the details of the various building blocks of the hardware of an embedded system. As shown in Fig. the building blocks are;

* Central Processing Unit (CPU)
* Memory (Read-only Memory and Random Access Memory)
* Input Devices
* Output devices
* Communication interface
* Application-specific circuitry

## Central Processing Unit (CPU)

The Central Processing Unit (processor, in short) can be any of the following: microcontroller,microprocessor or Digital Signal Processor (DSP). A micro-controller is a low-cost processor. Its main attraction is that on the chip itself, there will be many other

components such as memory, serial communication interface, analog-to digital converter etc. So, for small applications, a micro-controller is the best choice as the number of external components required will be very less. On the other hand, microprocessors are more powerful, but you need to use many external components with them. D5P is used mainly for applications in which signal processing is involved such as audio and video processing.

## Memory

The memory is categorized as Random Access 11emory (RAM) and Read Only Memory (ROM). The contents of the RAM will be erased if power is switched off to the chip, whereas ROM retains the contents even if the power is switched off. So, the firmware is stored in the ROM. When power is switched on, the processor reads the ROM; the program is program is executed.

## Input devices

Unlike the desktops, the input devices to an embedded system have very limited capability. There will be no keyboard or a mouse, and hence interacting with the embedded system is no easy task. Many embedded systems will have a small keypad-you press one key to give a specific command. A keypad may be used to input only the digits. Many embedded systems used in process control do not have any input device *for* user interaction; they take inputs *from* sensors or transducers 1’fnd produce electrical signals that are in turn fed to other systems.

## Output devices

The output devices of the embedded systems also have very limited capability. Some embedded systems will have a *few* Light Emitting Diodes (LEDs) *to* indicate the health status of the system modules, or *for* visual indication of alarms. A small Liquid Crystal Display (LCD) may also be used to display *some* important parameters.

## Communication interfaces

The embedded systems may need to, interact with other embedded systems at they may have to transmit data to a desktop. To facilitate this, the embedded systems are provided with one or a *few* communication interfaces such as RS232, RS422, RS485, Universal Serial Bus (USB), IEEE 1394, Ethernet etc.

## Application-specific circuitry

Sensors, transducers, special processing and control circuitry may be required fat an embedded system, depending on its application. This circuitry interacts with the processor to carry out the necessary work. The entire hardware has to be given power supply either through the 230 volts main supply or through a battery. The hardware has to design in such a way that the power consumption is minimized.

## DEBUGGING OF AN EMBEDDED SYSTEMS

Embedded debugging may be performed at different levels, depending on the facilities available. From simplest to most sophisticate they can be roughly grouped into the following areas:

* Interactive resident debugging using the simple shell provided by the embedded operating system( e.g. Forth and Basic )
* External debugging using logging or serial port output to trace operating using either a monitor in flash or using a debug server like the Remedy Debugger which even works for heterogeneous multi core systems.
* An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a JTAG or NEXUS interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.
* An in- circuit emulator (ICE) replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.

A complete emulator provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified and allowing debugging on a normal PC.

CHAPTER 02

PROPOSED SYSTEM

**Introduction**

Theft is one of the major problem in todays world places like in offices and other public places should not be secured so that issues to make secure our documents and precious things so we have decided to make this type of security system that will be more usable to all the people . This system assures the perfect use on the fingerprints for door opening and closing. Through the project we can provide high security to users. The fingerprint most of the banks have lockers such that one key is with the user and the bank has a master key. They also have password which the user has to tell the bank before going in the locker room, now if the user loses the key then, it is a big security risk. there are many thieves around us that they can easily or forcefully break our lockers so we can lost our property so to overcome this problem we are creating this type of security system Many of the bank lockers do not guarantee full safety of the user. In the fingerprint bank locker system we can easily add more than 1 fingerprint in the system so we can add our family member fingerprint as a nominee. And we can insert our multi hand fingerprint if we are facing accident and if we wound or a cut in our finger so we can use our nominee fingerprint or other multi hand fingerprint. If we are away from our house and we required urgent document or property so our family members can also use our lockers. this is a very a unique idea instead to keep keys or to protect that keys. Biometric devices are highly secured security identification and authentication device. Such devices use automated methods of verifying and recognising the identity of a living person based on a physiological behavioural characteristic. These characteristics include fingerprints, facial images, iris and voice recognisition.

**Literature survey**

R. Ramani (2012) et al. defined a bank locker security system which is based on RFID and GSM technology which can be implemented in banks, secure offices and homes. In this system, only authorized people can get money from a safe deposit box. We have implemented a bank locker security system based on RFID and GSM technology which includes a door locking system with RFID and GSM that can activate, authenticate and validate the user and unlockthe door in real time for safe access to the safe deposit box. The main advantage of using passive RFID and GSM is that it provides more security than other systems. This system consists of a microcontroller, RFID reader, GSM modem, keyboard and LCD, in this system the RFID reader reads the identification number of the passive tag and sends it tothe microcontroller if the identification number is valid, then the microcontroller sends the SMS request to the mobile phone number of the authenticated person, the original password to open and l safe deposit box if the person sends the password to the microcontroller, which is responsible for the the keypad will verify passwords entered and received by an authenticated mobile If these two passwords match, the locker will open; otherwise it will remain in a locked position. This system is more secure than other systems as it requires two passwords for verification. This system also creates a logbook that contains the record and output of each user along with basic user information. Raghu Ram.Gangi (2013) and others have given a proposal for fingerprint verification of the security system of automatic teller machines using biometrics with hybridization. The fingerprint function is chosen for its availability, reliability and high precision. The fingerprintbased biometric system can easily be implemented to secure the ATM. In this system, the operation of these ATMs is that upon accessing the ATM, the customer inserts the fingerprint module to withdraw the money, then the machine wants the fingerprint of the user using the machine, check / identify and give accurate with biometric fingerprints results if it's valid or not.In thismanner we can try to control and secure the criminal circle of the ATMs and lockers. Sanal Malhotra (2014) has given a proposal for a bank locker security system with Odour identification, Security Questions using RFID and GSM technology which can be utilized in banks, companies and at personal secured places. Only the original account holder is able to use his locker. This system uses Odor identification, Security question technique, RFID and GSM technology which makes it more secure than any other system. The system has the capability of providing more security as 4 steps are used for verification. RFID tag can be verified using RFID technology, then valid person has to answer the security question by using Security question software techniques and it should be same as that of already stored then the valid person gets a message in his/her mobile using GSM technology and needs to type password from his/her mobile and keypad of locker, both passwords should match to open the door of the locker, and then odor identification will be done, the odor pattern should match with the odour pattern stored in the microcontroller.

These are some of the existing Smart Security designs that have been implemented- (a) GSM Based Security System PIR sensor detects motion by sensing the difference in infrared or radiant heat levels emitted by surrounding objects. The output of the PIR sensor goes high when it detects any motion. The range of a typical PIR sensor is around 6 meters or about 30 feet. When the PIR sensor detects any motion, the output of the sensor is high. This is detected by the Arduino. Then it communicates with the GSM module via serial communication to make a call to the preprogrammed mobile number. An important point to be noted about PIR sensors is that the output will be high when it detects motion. (b) IR based security alarm systemIR based security alarm circuit can detect any movement and trigger the alarm. This circuit is very useful in homes, banks, shops, restricted areas where an alert alarm is needed on any movement. This circuit is based on IR sensor where an IR beam is continuously falling on a photodiode, and whenever this Infrared beam breaks, by any kind of movement, alarm is triggered. In this IR based security alarm circuit, we have placed IR LED in front of photodiode, so that IR light can directly falls on photodiode. Whenever someone moves through this beam, IR rays stops falling on photodiode and Buzzer start beeping. Internet of things has been governing the electronics with cloud services influencing the ever increasing electronics product segment. Security and safety has always become a basic necessity for urban population. The paper proposes a security system based on Open source cloud server “things speak .com” and a low cost esp8266 Wi-Fi module. The project includes a PIR module which constantly monitoring the Home or Work space to be monitored .When the PIR module detects a intruder it sends a signal to the Atmega 328p microcontroller and the controller is connected to a Esp8266 wifi module and also to a alarm system. The System transmits an alert signal to the Open source cloud which provides a alert signal on the users mobile phone. The system employs a second esp8266 module which is programmed to act as a web server, which allows the user to activate or deactivate the security system by means of any device with internet. The system also employs a thumb print reader rs305 which controls the opening and the closing of a safety locker door. Thus the system uses esp8266 WiFi module and atmega328p to control the security system from the user’s mobile phone by means of any device with a potential internet connection.

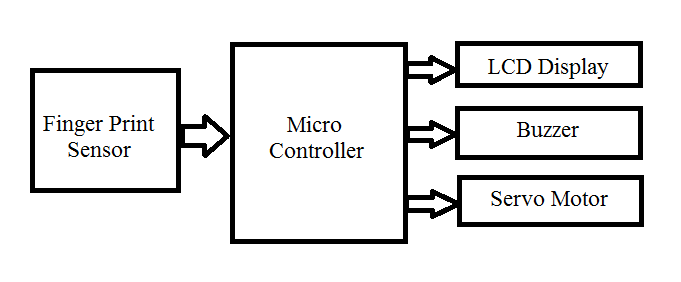
**Existing System**

In the real world, peoples are more concerned about their safety for their valuable things like jewellery, money, important documents etc. So the bank lockers are the safest place to store them. The arrival of fast growing technologies makes users to have high security systems with electronic identification options. These identification technologies include Bank Lockers and ATM as well as other intelligent cards, user IDs and password based systems, and so on. But, unfortunately these are not protected due to hacker attacks, thefts, and forgotten passwords. In spite of all these faults or failure and malfunctions or crash these systems are still existing; however, the biometric or fingerprint authentication based identification is the most efficient and reliable solution for stringent security.

**Proposed System**

Finger print processing includes two parts: finger print enrolment and finger print matching (the matching can be 1:1 or 1:N). When enrolling, user needs to enter the finger image two times. The system will process the finger images, generate a template of the finger images based on processing results and store the template. When matching, user enters the finger through optical sensor and system will generate a template of the finger image and compare it with templates present in the finger library. For 1:1 matching, system will compare the actual subject finger image with specific template designated in the module; for 1:N matching, or searching, system will search the whole finger library for matching user finger image. In both circumstances, system will return the matching results as a success or failure.

**Block Diagram**

****

The fingerprint based bank locker security system is an advanced version of the traditional bank locker system which uses keys. Keys can now be easily copied and made by thieves. In addition, the keys need to be taken care of and they can also be lost through some neglect.The fingerprint based bank locker security system can solve all of these problems. The fingerprint-based bank locker system is more secure and it is easy to use and maintain. Here there is no need for key handling and hence you don't have to worry about losing keys.The system uses fingerprint recognition to read the fingerprints and first stores the registered fingerprints in the bank locker register. The next time when a person scans his/her finger, the sensor reads it and compares it to previous recordings. If a match is now found with the existing fingerprints, it sends the match signal to the microcontroller and the controller displays this data on the LCD screen. The controller also operates the driver motor to open the locker to authorized customers. The locker remains closed for unauthorized customers.

APPLICATIONS

1. Used to increase sequrity
2. Used in Banks, Hostels

ADVANTAGES

* Multiple fingerprints can be able to add
* Easy to implement

LITERATURE REVIEW

Thirumalai1, Gokul. R2, Ganasekaran. P3, Manellore Murali. M4, Jackson Jublience Joseph. L5

1Assistant Professor, Department of Electronic and Communication Engineering 2, 3, 4, 5 Students, Department of Electronics and Communication Engineering, TJS Engineering College, Peruvoyal, Chennai – 601 206, Tamil Nadu, India

Abstract:- This paper will focused on effective recognizing and controlling system for Bank locker room which is fully self-determining. In cases of robberies, its commonly happen that the banned entrance in the locker room area which can be detected by our security system. If the robbery take place the banks are not be capable to recognize the robber due to absence of the proof by using the current human operated security system. Development of various sensors has enabled systems to have preventive and corrective measures in this regard significantly. In order to deliver a concrete security solution for critically important and confidential documents and goods, we proposed an Automated Safety Vault with Double Layered Defense Mechanism. The solution comprised of an Electronic Lock driven by password verification and a Biometric authentication for users using a Fingerprint scanning and sensing tool. Both of these two layers ensured the authenticity of the user by preventing any unauthorized access to the Vault. The system was then implemented in a prototype scope for testing and validation of the proposals. The implemented system and testing data showed that the Automated Safety Vault with all its security features had successful operation. The specification of the whole system as well as the results is observed and verified.

R.Ramani , S. Selvaraju , S.Valarmathy, P.Niranjan , “Bank Locker Security System based on RFID and GSM Technology ’’, International Journal of Computer Applications (0975 – 8887) Volume 57– No.18, November 2012 The main goal of this paper is to design and implement a bank locker security system based on RFID and GSM technology which can be organized in bank, secured offices and homes. In this system only authentic person can be recovered money from bank locker. We have implemented a bank locker security system based on RFID and GSM technology containing door locking system using RFID and GSM which can activate, authenticate, and validate the user and unlock the door in real time for bank locker secure access. The main advantage of using passive RFID and GSM is more secure than other systems. This system consists of microcontroller, RFID reader, GSM modem, keyboard, and LCD, in this system The RFID reader reads the id number from passive tag and send to the microcontroller, if the id number is valid then microcontroller send the SMS request to the authenticated person mobile number, for the original password to open the bank locker, if the person send the password to the microcontroller, which will verify the passwords entered by the key board and received from authenticated mobile phone. if these two passwords are matched the locker will be opened otherwise it will be remain in locked position, This system is more secure than other systems because two passwords required for verification. This system also creates a log containing check-in and check-out of each user along with basic information of user.

Abhilasha A Sayar1 , Dr. Sunil N Pawar2 , “Review of Bank Locker System Using Embedded System” , International Journal of Advanced Research in Computer and Communication Engineering .,Vol. 5, Issue 2, February 2016 . In this review paper various bank locker security system designs are explained. In today’s high-speed world, security plays significant role. People are now more concern of their belongings like valuable documents, jewelry, and many more material. The safest place to keep all such valuable is bank. With the advancement in technology there are many system designed to keep bank lockers safe.

**CHAPTER 3**

**HARDWARE CONSTRAINTS**

**NodeMCU**

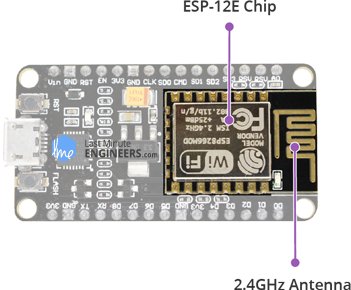
The Internet of Things (IoT) has been a trending field in the world of technology. It has changed the way we work. Physical objects and the digital world are connected now more than ever. Keeping this in mind, [Espressif Systems](https://www.espressif.com/) (A Shanghai-based Semiconductor Company) has released an adorable, bite-sized WiFi enabled microcontroller – **ESP8266**, at an unbelievable price! For less than $3, it can monitor and control things from anywhere in the world – **perfect for just about any IoT project**.

## ESP-12E Module

The development board equips the ESP-12E module containing ESP8266 chip having **Tensilica Xtensa® 32-bit LX106 RISC microprocessor** which operates at **80 to 160 MHz** adjustable clock frequency and supports **RTOS**.

**ESP-12E Chip**

* Tensilica Xtensa® 32-bit LX106
* 80 to 160 MHz Clock Freq.
* 128kB internal RAM
* 4MB external flash
* 802.11b/g/n Wi-Fi transceiver



There’s also **128 KB RAM and 4MB of Flash memory** (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays.

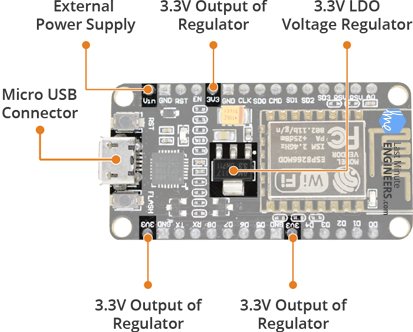
The ESP8266 Integrates **802.11b/g/n HT40 Wi-Fi transceiver**, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU even more versatile.

## Power Requirement

As the operating voltage range of ESP8266 is **3V to 3.6V**, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as **80mA during RF transmissions**. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

**Power Requirement**

* Operating Voltage: 2.5V to 3.6V
* On-board 3.3V 600mA regulator
* 80mA Operating Current
* 20 µA during Sleep Mode



**Power to the ESP8266 NodeMCU**is supplied via the **on-board MicroB USB connector**. Alternatively, if you have a regulated 5V voltage source, the **VIN pin** can be used to directly supply the ESP8266 and its peripherals.

Warning:

The ESP8266 requires a 3.3V power supply and 3.3V logic levels for communication. The GPIO pins are not 5V-tolerant! If you want to interface the board with 5V (or higher) components, you’ll need to do some level shifting.

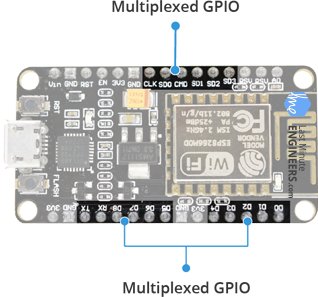
## Peripherals and I/O

The ESP8266 NodeMCU has total **17 GPIO pins** broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

* **ADC channel** – A 10-bit ADC channel.
* **UART interface** – UART interface is used to load code serially.
* **PWM outputs** – PWM pins for dimming LEDs or controlling motors.
* **SPI, I2C & I2S interface** – SPI and I2C interface to hook up all sorts of sensors and peripherals.
* **I2S interface** – I2S interface if you want to add sound to your project.

**Multiplexed I/Os**

* 1 ADC channels
* 2 UART interfaces
* 4 PWM outputs
* SPI, I2C & I2S interface



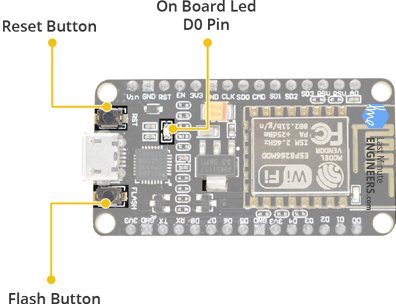
Thanks to the ESP8266’s **pin multiplexing feature** (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as PWM/UART/SPI.

## On-board Switches & LED Indicator

The ESP8266 NodeMCU features two buttons. One marked as **RST** located on the top left corner is the Reset button, used of course to reset the ESP8266 chip. The other **FLASH** button on the bottom left corner is the download button used while upgrading firmware.

**Switches & Indicators**

* RST – Reset the ESP8266 chip
* FLASH – Download new programs
* Blue LED – User Programmable



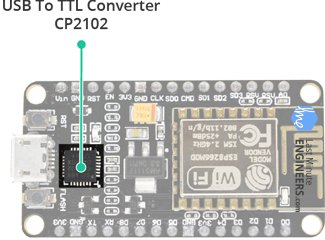
The board also has a**LED indicator** which is user programmable and is connected to the D0 pin of the board.

## Serial Communication

The board includes CP2102 USB-to-UART Bridge Controller from [Silicon Labs](http://www.silabs.com/), which converts USB signal to serial and allows your computer to program and communicate with the ESP8266 chip.

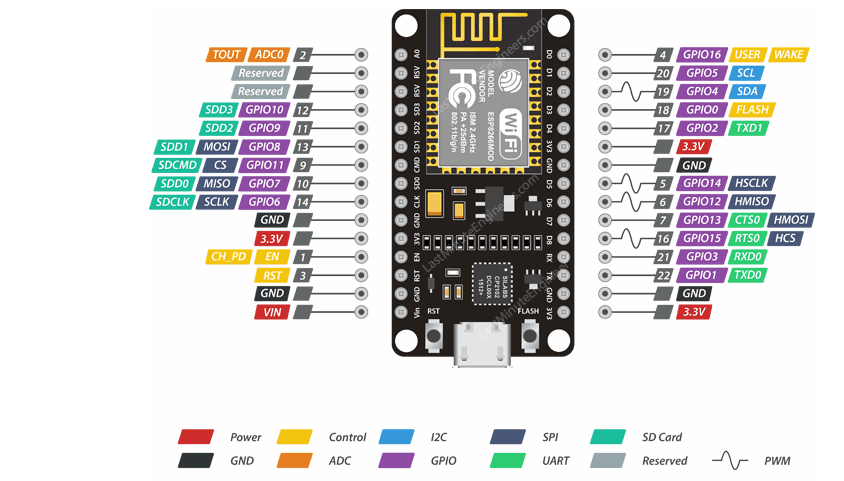
**Serial Communication**

* CP2102 USB-to-UART converter
* 4.5 Mbps communication speed
* Flow Control support



## ESP8266 NodeMCU Pinout

The ESP8266 NodeMCU has total 30 pins that interface it to the outside world. The connections are as follows:



Power Pins There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP8266 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

GND is a ground pin of ESP8266 NodeMCU development board.

I2C Pins are used to hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO Pins ESP8266 NodeMCU has 17 GPIO pins which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins ESP8266 NodeMCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports fluid control. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

SPI Pins ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

* 4 timing modes of the SPI format transfer
* Up to 80 MHz and the divided clocks of 80 MHz
* Up to 64-Byte FIFO

SDIO Pins ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μs to 10000 μs, i.e., between 100 Hz and 1 kHz.

Control Pins are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

* EN pin – The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
* RST pin – RST pin is used to reset the ESP8266 chip.
* WAKE pin – Wake pin is used to wake the chip from deep-sleep.

## ESP8266 Development Platforms

Now, let’s move on to the interesting stuff!

There are a variety of development platforms that can be equipped to program the ESP8266. You can go with [Espruino](https://www.espruino.com/) – JavaScript SDK and firmware closely emulating Node.js, or use [Mongoose OS](https://mongoose-os.com/) – An operating system for IoT devices (recommended platform by Espressif Systems and Google Cloud IoT) or use a software development kit (SDK) provided by Espressif or one of the platforms listed on [WiKiPedia](https://en.wikipedia.org/wiki/ESP8266" \l "SDKs).

Fortunately, the amazing ESP8266 community took the IDE selection a step further by creating an Arduino add-on. If you’re just getting started programming the ESP8266, this is the environment we recommend beginning with, and the one we’ll document in this tutorial.

This ESP8266 add-on for Arduino is based on the amazing work by [Ivan Grokhotkov](https://github.com/igrr) and the rest of the ESP8266 community. Check out the [ESP8266 Arduino GitHub repository](https://github.com/esp8266/Arduino) for more information.

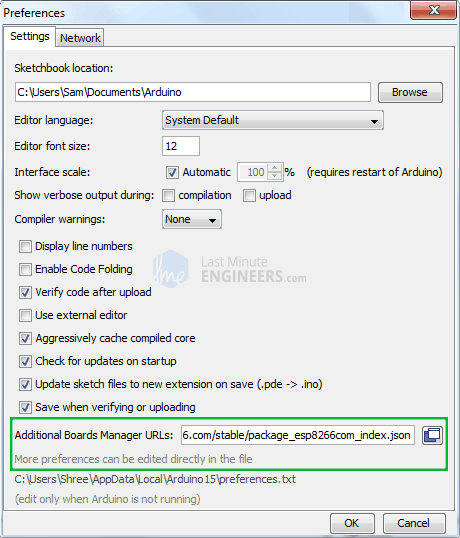
## Installing the ESP8266 Core on Windows OS

Let’s proceed with installing ESP8266 Arduino core.

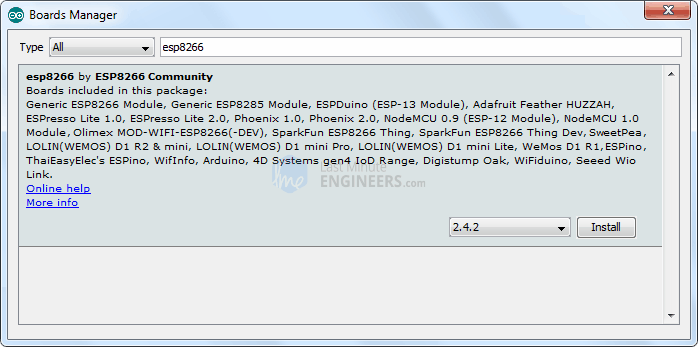
The first thing is having latest Arduino IDE (Arduino 1.6.4 or higher) installed on your PC. If don’t have it, we recommend upgrading now.

To begin, we’ll need to update the board manager with a custom URL. Open up Arduino IDE and go to File > Preferences. Then, copy below URL into the Additional Board Manager URLs text box situated on the bottom of the window:

http://arduino.esp8266.com/stable/package\_esp8266com\_index.json



Hit OK. Then navigate to the Board Manager by going to Tools > Boards > Boards Manager. There should be a couple new entries in addition to the standard Arduino boards. Filter your search by typing esp8266. Click on that entry and select Install.



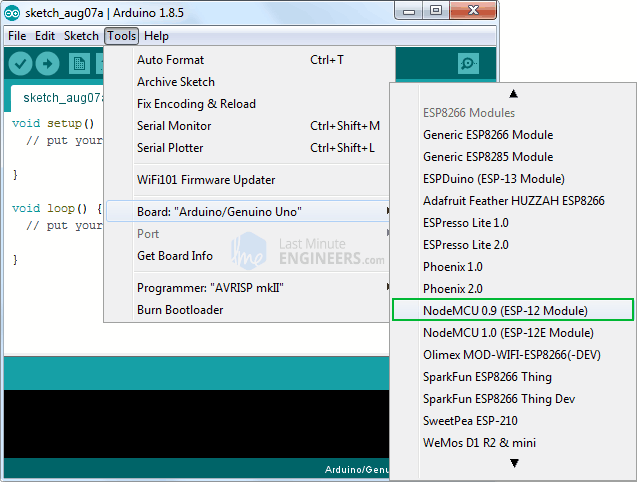
The board definitions and tools for the ESP8266 include a whole new set of gcc, g++, and other reasonably large, compiled binaries, so it may take a few minutes to download and install (the archived file is ~110MB). Once the installation has completed, a small *INSTALLED* text will appear next to the entry. You can now close the Board Manager.

Arduino Example: Blink

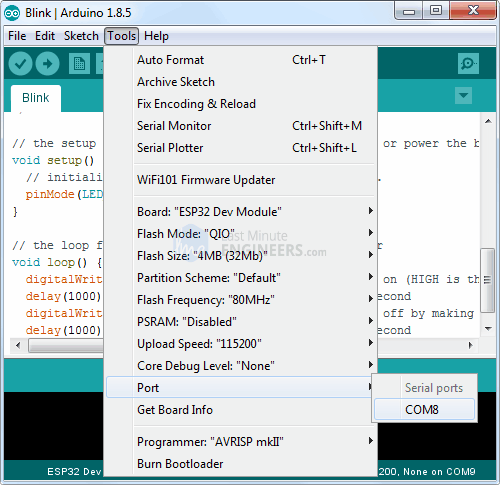
To make sure ESP8266 Arduino core and the NodeMCU are properly set up, we’ll upload the simplest sketch of all – The Blink!

We will use the on-board LED for this test. As mentioned earlier in this tutorial, D0 pin of the board is connected to on-board Blue LED & is user programmable. Perfect

Before we get to uploading sketch & playing with LED, we need to make sure that the board is selected properly in Arduino IDE. Open Arduino IDE and select NodeMCU 0.9 (ESP-12 Module) option under your Arduino IDE > Tools > Board menu.



Now, plug your ESP8266 NodeMCU into your computer via micro-B USB cable. Once the board is plugged in, it should be assigned a unique COM port. On Windows machines, this will be something like COM#, and on Mac/Linux computers it will come in the form of /dev/tty.usbserial-XXXXXX. Select this serial port under the Arduino IDE > Tools > Port menu. Also select the Upload Speed : 115200



Warning:

More attention needs to be given to selecting board, choosing COM port and selecting Upload speed. You may get espcomm\_upload\_mem error while uploading new sketches, if failed to do so.

Once you are done, try the example sketch below.

void setup()

{

pinMode(D0, OUTPUT);

}

void loop()

{

digitalWrite(D0, HIGH);

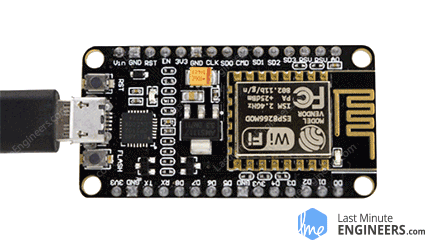
delay(500);

digitalWrite(D0, LOW);

delay(500);

}

Once the code is uploaded, LED will start blinking. You may need to tap the RST button to get your ESP8266 to begin running the sketch.



## POWER SUPPLY

## REGULATED POWER SUPPLY

The DC power supply is practically converted to each and every stage in an electronic system. Thus a common requirement for all these phases will be the DC power supply. All low power system can be run with a battery. But, for a long time operating devices, batteries could prove to be costly and complicated. The best method used is in the form of an unregulated power supply –a combination of a transformer, rectifier and a filter. The diagram is shown below.

All devices will have a certain power supply limit and the electronic circuits inside these devices must be able to supply a constant DC voltage within this limit. This DC supply is regulated and limited in terms of voltage and current. But the supply provided from mains may be fluctuating and could easily break down the electronic equipment, if not properly limited. This work of converting an unregulated alternating

current (AC) or voltage to a limited Direct current (DC) or voltage to make the output constant regardless of the fluctuations in input, is done by a regulated power supply circuit.

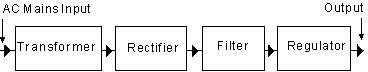


Fig 3.5 Block diagram of Regulated power supply

All the active and passive electronic devices will have a certain DC operating point (Q-point or Quiescent point), and this point must be achieved by the source of DC power. A step down transformer is used to reduce the voltage level to the devices needs. In India, a 1 Ø supply is available at 230 volts. The output of the transformer is a pulsating sinusoidal AC voltage, which is converted to pulsating DC with the help of a rectifier.

This output is given to a filter circuit which reduces the AC ripples, and passes the DC components. But here are certain disadvantages in using an unregulated power supply.

Regulated power supply is an electronic circuit that is designed to provide a constant dc voltage of predetermined value across load terminals irrespective of ac mains fluctuations or load variations.

A regulated power supply essentially consists of an ordinary power supply and a voltage regulating device, as illustrated in the figure. The output from an ordinary power supply is fed to the voltage regulating device that provides the final output. The output voltage remains constant irrespective of variations in the ac input voltage or variations in output (or load) current.

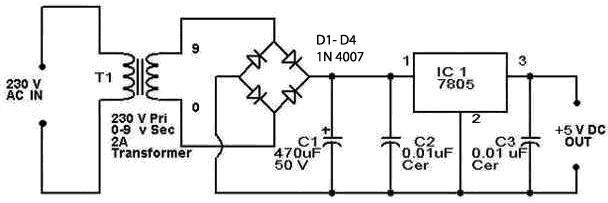


Fig 3.6 Circuit diagram of Regulated power supply

## TRANSFORMER

A transformer can be defined as a static device which helps in the transformation of electric power in one circuit to electric power of the same frequency in another circuit. The voltage can be raised or lowered in a circuit, but with a proportional increase or decrease in the current ratings.

## TRANSFORMER WORKINGPRINCIPLE

The main principle of operation of a transformer is mutual inductance between two circuits which is linked by a common magnetic flux. A basic transformer consists of two coils that are electrically separate and inductive, but are magnetically linked through a path of reluctance. The working principle of the transformer can be understood from the figure below.

As shown below the electrical transformer has primary and secondary windings. The core laminations are joined in the form of strips in between the strips you can see that there are some narrow gaps right through the cross-section of the core. These staggered joints are said to be ‘imbricated’. Both the coils have high mutual inductance.A mutual electro-motive force is induced in the transformer from the alternating flux that is set up in the laminated core, due to the coil that is connected to a source of alternating voltage.

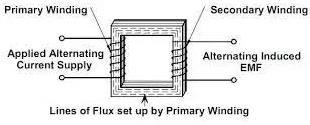


Fig 3.7 Working principle of Transformer

Most of the alternating flux developed by this coil is linked with the other coil and thus produces the mutual induced electro-motive force. The so produced electro-motive force can be explained with the help of Faraday’s laws of Electromagnetic Induction as

e=M\*dI/dt

If the second coil circuit is closed, a current flows in it and thus electrical energy is transferred magnetically from the first to the second coil.The alternating current supply is given to the first coil and hence it can be called as the primary winding. The energy is drawn out from the second coil and thus can be called as the secondary winding.

In short, a transformer carries the operations shown below:

* + - * Transfer of electric power from one circuit to another.
      * Transfer of electric power without any change in frequency.
      * Transfer with the principle of electromagnetic induction.
      * The two electrical circuits are linked by mutual induction.

## CLASSIFICATION OF TRANSFORMER

* + - * Step-Up Transformer
      * Step-Down Transformer

## Step-Down Transformer

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels. An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1. Step down transformers can be considered nothing more than a voltage ratio device.

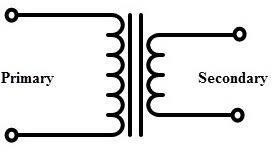


Fig 3.8 Step Down Transformer

## Step-Up Transformer

A step up transformer has more turns of wire on the secondary coil, which makes a larger induced voltage in the secondary coil. It is called a step up transformer because the voltage output is larger than the voltage input. Step-up transformer 110v 220v design is one whose secondary voltage is greater than its primary voltage. This kind of transformer "steps up" the voltage applied to it. For instance, a step up transformer is needed to use a 220v product in a country with a 110v supply. A step up transformer 110v 220v converts alternating current (AC) from one voltage to another voltage. It has no moving parts and works on a magnetic induction principle; it can be designed to "step- up" or "step-down" voltage. So a step up transformer increases the voltage and a step down transformer decreases the voltage.

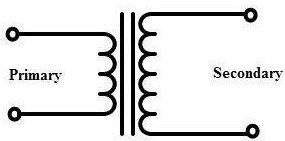


Fig 3.9 Step Up Transformer

## APPLICATIONS OF TRANSFORMER

* + - * The [transformer](http://www.polytechnichub.com/transformer/) used for impedance matching.
      * The transformer used for isolate two circuits electrically.
      * It is used to increase or decrease the alternating voltages in electric power applications.
      * The transformer used in voltmeter, ammeters, protective relay etc.
      * The transformer used for step up low voltage in case of measurement.
      * The transformer used for step down high voltage for safety.
      * The transformer used in **rectifier**.
      * It is used in voltage regulators, voltage stabilizers, power supplies etc.

## RECTIFIER CIRCUIT

A rectifier is an electrical device that converts alternating current (AC), which periodically reverses direction, to direct current (DC), which flows in only one direction. The process is known as rectification.

## HALF WAVE RECTIFIER

A simple Half Wave Rectifier is nothing more than a single pn junction diode connected in series to the load resistor. As you know a diode is to electric current like a one-way valve is to water, it allows electric current to flow in only one direction.

This property of the diode is very useful in creating simple rectifiers which are used to convert AC to DC. If you look at the Half Wave Rectifier diagram, we are giving an alternating current as input.

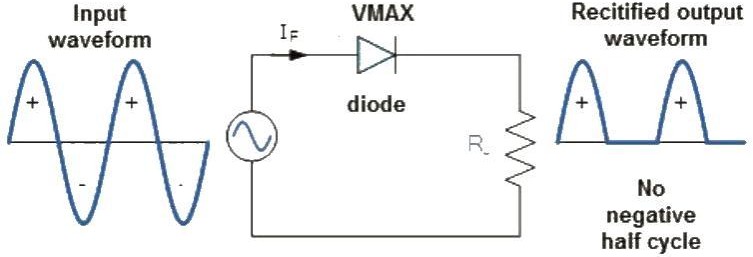


Fig 3.10 Half wave Rectifier

Input voltage is given to a step-down transformer and the resulting reduced output of the transformer is given to the diode ‘D’ and load resistor RL. The output voltage is measured across load resistor RL.

## FULL WAVE RECTIFIER

Full wave rectifier rectifies the full cycle in the waveform i.e. it rectifies both the positive and negative cycles in the waveform. This Full wave rectifier has an advantage over the half wave i.e. it has average output higher than that of half wave rectifier. The number of AC components in the output is less than that of the input.

The full wave rectifier can be further divided mainly into following types.

* + - * Centre Tapped Full Wave Rectifier
      * Full Wave Bridge Rectifier

## 

## CENTRE TAPPED FULL WAVE RECTIFIER

Centre tap is the contact made at the middle of the winding of the transformer. In the centre tapped full wave rectifier two diodes were used. These are connected to the centre tapped secondary winding of the transformer. Above circuit diagram shows the centre tapped full wave rectifier. It has two diodes. The positive terminal of two diodes is connected to the two ends of the transformer. Centre tap divides the total secondary voltage into equal parts.

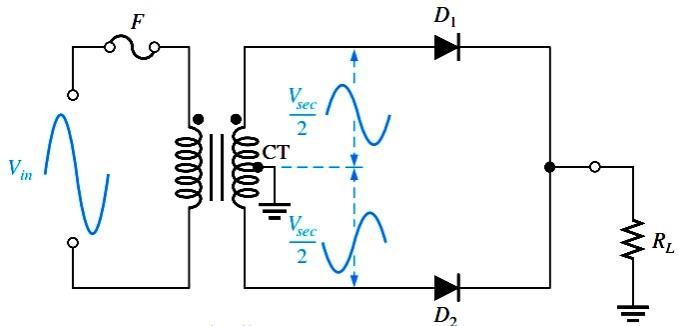


Fig 3.11 Centre Tapped Full Wave Rectifier

## FULL WAVE BRIDGE RECTIFIER

Bridge is a type of electrical circuit. Bridge rectifier is a type of rectifier in which diodes were arranged in the form of a bridge. This provides full wave rectification and is of low cost. So it is used in many applications.

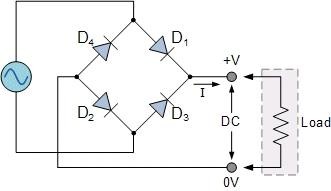


Fig 3.12 Full Wave Bridge Rectifier

## Working of Bridge Rectifier

The working of a bridge rectifier is simple. The circuit diagram of bridge rectifier is given above. The secondary winding of the transformer is connected to the two diametrically opposite points of the bridge at points 1 and 3. Assume that a load is connected at the output. The load RLoad is connected to bridge through points 2 and 4.

During first half cycle of the AC input, the upper portion of the transformer secondary winding is positive with respect to the lower portion. Thus during the first half cycle diodes D1 and D4 are forward biased. Current flows through path 1-2, enter into the load RL. It returns back flowing through path 4-3. During this half input cycle, the diodes D2 and D3 are reverse biased. Hence there is no current flow through the path 2-3 and 1-4.

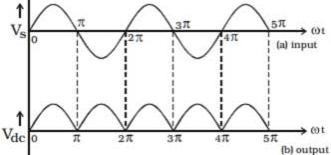


Fig 3.13 Output Wave Forms of Full Wave Bridge Rectifier

During the next cycle lower portion of the transformer is positive with respect to the upper portion. Hence during this cycle diodes D2 and D3 are forward biased. Current flows through the path 3-2 and flows back through the path 4-1.The diodes D1 and D4 are reverse biased. So there is no current flow through the path 1-2 and 3-4.Thus negative cycle is rectified and it appears across the load.

## FILTER CIRCUIT

A filter circuit is a device to remove the A.C components of the rectified output, but allows the D.C components to reach the load. A filter circuit is in general a combination of inductor (L) and Capacitor (C) called LC filter circuit. A capacitor allows

A.C only and inductor allows D.C only to pass.

## Components involved in filter Circuit

A filter circuit comprises of generally inductor and capacitor. The inductor allows DC only to pass through it and capacitor allows AC only to pass through it. Thus, a circuit formed by the combination of inductors and capacitors can effectively filter the signal according to the application.

## CAPACITOR FILTER

This filter consists of a high value capacitor placed directly across the load resistor. This capacitor then gets charged i.e. it stores energy during the conducting period and delivers this energy to the load during the non-conducting period. Through this process, the time duration during which the current flows through the load resistor gets prolonged and ripple component gets considerably reduces.

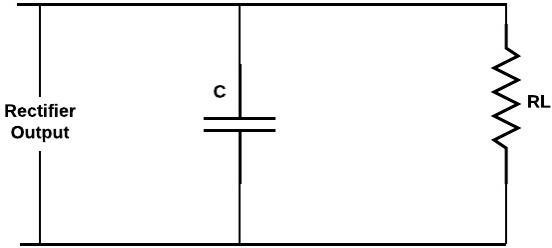


Fig 3.14 Capacitor Filter

The function of the capacitor filte may be viewed in terms of

Impedance \dfrac{1}{2\pi f C} to the ripple component of frequency f. this ripple, component of current, therefore, gets bypassed through C and only dc component flows through the load resistor RL.

## 

## VOLTAGE REGULATOR

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages.

A voltage regulator is designed to automatically ‘regulate’ voltage level. It basically steps down the input voltage to the desired level and keeps that in that same level during the supply. This makes sure that even when a load is applied the voltage doesn’t drop.

These regulators consists the three pins there are Pin1: It is used for input pin.

Pin2: This is ground pin for regulator

Pin3: It is used for output pin. Through this pin we get the output.

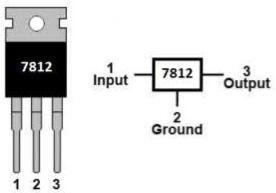


Fig 3.15 12 V Voltage Regulator

There are two types of regulator are they.

* + - * Positive Voltage Series (78xx) and
      * Negative Voltage Series (79xx)

## Positive Voltage Series (78xx)

’78’ indicate the positive series and ‘05’indicates the voltage rating. In this priect we have to used two voltage regulators there are 7802 and 7812 voltage regulators.

Thee 7805 produces the maximum 5V.’05’indicates the regulator output is 5V. And The 7812 produces the maximum 12V. ’12’indicates the regulator output is 12V.

## Negative Voltage Series (79xx)

’78’ indicate the negative series and ‘xx’ indicates the voltage rating. Suppose 7905 produces the maximum -5V.’05’indicates the regulator output is -5V.

**SERVO MOTOR**

A servomotor is a linear actuator or rotary actuator that allows for precise control of linear or angular position, acceleration, and velocity. It consists of a motor coupled to a [sensor](https://www.electrical4u.com/sensor-types-of-sensor/) for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

****

**LIQUID CRYSTAL DISPLAY (LCD)**

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because of the following reasons:

* The declining prices of LCDs.
* The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.
* Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU to keep displaying the data.
* Ease of programming for characters and graphics.

These components are “specialized” for being used with the microcontrollers, which means that they cannot be activated by standard IC circuits. They are used for writing different messages on a miniature LCD.

A model described here is for its low price and great possibilities most frequently used in practice. It is based on the HD44780 microcontroller (Hitachi) and can display messages in two lines with 16 characters each. It displays all the alphabets, Greek letters, punctuation marks, mathematical symbols etc. In addition, it is possible to display symbols that user makes up on its own. Automatic message on display (shift left and right), appearance of the pointer, backlight etc. are considered as useful characteristics.

**LCD CONNECTIONS**

Depending on how many lines are used for connection to the microcontroller, there are 8-bit and 4-bit LCD modes. The appropriate mode is determined at the beginning of the process in a phase called “initialization”. In the first case, the data are transferred through outputs D0-D7 as it has been already explained. In case of 4-bit LED mode, for the sake of saving valuable I/O pins of the microcontroller, there are only 4 higher bits (D4-D7) used for communication, while other may be left unconnected.

Consequently, each data is sent to LCD in two steps: four higher bits are sent first (that normally would be sent through lines D4-D7), four lower bits are sent afterwards. With the help of initialization, LCD will correctly connect and interpret each data received. Besides, with regards to the fact that data are rarely read from LCD (data mainly are transferred from microcontroller to LCD) one more I/O pin may be saved by simple connecting R/W pin to the Ground. Such saving has its price. EvenEven though message displaying will be normally performed, it will not be possible to read from busy flag since it is not possible to read from display.

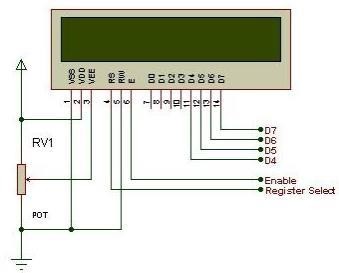


Fig 3.23 Pin Connections of LCD

**PIN CONFIGURATION OF LIQUID CRYSTAL DISPLAY**

|  |  |  |
| --- | --- | --- |
| **Pin No:** | **Pin Name:** | **Description** |
| 1 | Vss (Ground) | Ground pin connected to system ground |
| 2 | Vdd (+5 Volt) | Powers the LCD with +5V (4.7V – 5.3V) |
| 3 | VE  (Contrast V) | Decides the contrast level of display. Grounded to get maximum contrast. |
| 4 | Register Select | Connected to Microcontroller to shift between command/data register |
| 5 | Read/Write | Used to read or write data. Normally grounded to write data to LCD |
| 6 | Enable | Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgement |

|  |  |  |
| --- | --- | --- |
| 7 | Data Pin 0 | Data pins 0 to 7 forms a 8-bit data line. They can be connected to Microcontroller to send 8-bit data.  These LCD’s can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free. |
| 8 | Data Pin 1 |  |
| 9 | Data Pin 2 |
| 10 | Data Pin 3 |
| 11 | Data Pin 4 |
| 12 | Data Pin 5 |
| 13 | Data Pin 6 |
| 14 | Data Pin 7 |

|  |  |  |
| --- | --- | --- |
| 15 | LED  Positive | Backlight LED pin positive terminal |
| 16 | LED  Negative | Backlight LED pin negative terminal |

Table 3.3 Pin description table of 16\*2 LCD display

**BASIC COMMANDS OF LCD**

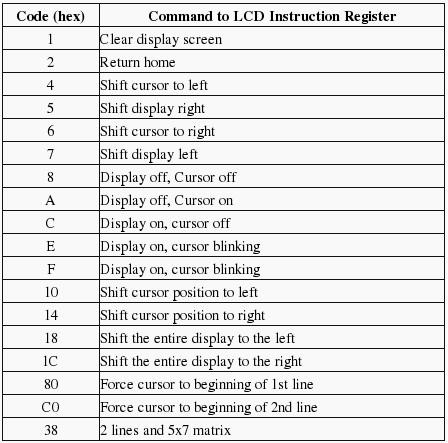


Table 3.4 Basic commands of LCD

**LCD INITIALISATION**

Once the power supply is turned on, LCD is automatically cleared. This process lasts for approximately 15mS. After that, display is ready to operate. The mode of operating is set by default. This means that:

* Display is cleared
* Mode
* DL = 1 Communication through 8-bit interface N = 0 Messages are displayed in one line
* F = 0 Character font 5 x 8 dots
* Display/Cursor on/off D = 0 Display off
* U = 0 Cursor off
* B = 0 Cursor blink off
* Character entry

ID = 1 Addresses on display are automatically incremented by 1 S = 0 Display shift off

Automatic reset is mainly performed without any problems. If for any reason power supply voltage does not reach full value in the course of 10mS, display will start perform completely unpredictably if voltage supply unit cannot meet this condition or if it is needed to provide completely safe operating, the process of initialization by which a new reset enabling display to operate normally must be applied.

Algorithm according to the initialization is being performed depends on whether connection to the microcontroller is through 4- or 8-bit interface. All left over to be done after that is to give basic commands and of course- to display messages.

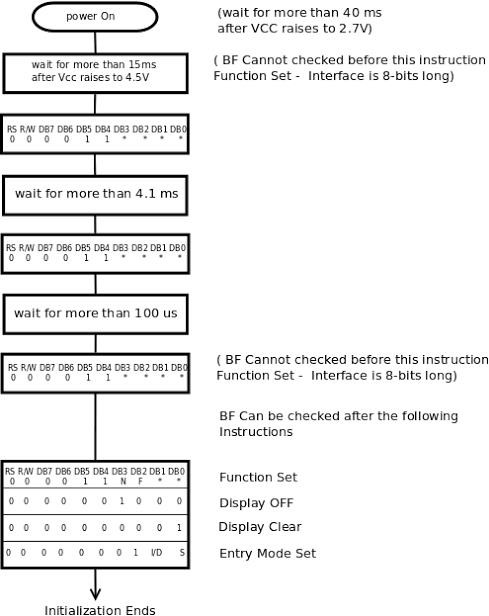


Fig 3.24 Initialization of LCD

**FEATURES OF 16×2 LCD**

* + - * Operating Voltage is 4.7V to 5.3V
      * Current consumption is 1mA without backlight
      * Alphanumeric LCD display module, meaning can display alphabets and numbers
      * Consists of two rows and each row can print 16 characters.
      * Each character is build by a 5×8 pixel box
      * Can work on both 8-bit and 4-bit mode
      * It can also display any custom generated characters
      * Available in Green and Blue Backlight.

**Buzzer**

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include **alarm devices, timers, and confirmation of user input such as a mouse click or keystroke**.



FINGER PRINT SENSOR

The fingerprint sensor is one kind of sensor which is used in a fingerprint detection device. These devices are mainly inbuilt in the fingerprint detection module and it is used for computer safety. The main features of this device mainly include accuracy, better performance, robustness based on exclusive fingerprint [biometric technology](https://www.elprocus.com/biometric-authentication-system-applications/). Both fingerprint scanner otherwise reader are an extremely safe & suitable device for safety instead of a secret word. Because the password is easy to scan and also it is hard to keep in mind.



### Features

The features of this sensor include the following.

* It includes image collection as well as chip algorithm
* The fingerprint reader can perform lesser growth and can be fixed into a range of end products
* Power use is low, excellent performance, small in size, and less cost
* [Optical technology](https://www.elprocus.com/optical-sensors-types-basics-and-applications/) which is used is professional, and exact module developed techniques
* The capabilities of[image processing](https://www.elprocus.com/image-processing-projects-for-engineering-students/) are good, and can effectively capture pictures up to 500 dpi resolution

# CHAPTER 04

# SOFTWARE CONSTRAINTS

## 

## INTRODUCTION TO ARDUINO IDE

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 programmed (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.

## 

## KEY FEATURES OF ARDUINO IDE

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.

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 onthe Arduino board.

## 

## ARDUINO DATA TYPES

Data types in C refers to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in the storage and how the bit pattern stored is interpreted.

The following table provides all the data types that you will use During Arduino programming.

## Void

The void keyword is used only in function declarations. It indicates that the function is expected to return no information to the function from which it was called.

Example:

Void Loop ( )

{

// rest of the code

}

## Boolean

A Boolean holds one of two values, true or false. Each Boolean variable occupies one byte of memory.

Example:

Boolean state= false ; // declaration of variable with type boolean and initialize it with false.

Boolean state = true ; // declaration of variable with type boolean and initialize it with false.

## Char

A data type that takes up one byte of memory that stores a character value. Character literals are written in single quotes like this: 'A' and for multiple characters, strings use double quotes: "ABC".

However, characters are stored as numbers. You can see the specific encoding in the ASCII chart. This means that it is possible to do arithmetic operations on characters, in whichthe ASCII value of the character is used. For example, 'A' + 1 has the value 66, since theASCII value of the capital letter A is 65.

Example:

Char chr\_a = ‘a’ ;//declaration of variable with type char and initialize it with character a.

Char chr\_c = 97 ;//declaration of variable with type char and initialize it with character 97

## Unsigned char

Unsigned char is an unsigned data type that occupies one byte of memory. The unsigned char data type encodes numbers from 0 to 255.

Example:

Unsigned Char chr\_y = 121 ; // declaration of variable with type Unsigned char and initialize it with character y

## Byte

A byte stores an 8-bit unsigned number, from 0 to 255. Example:

byte m = 25 ;//declaration of variable with type byte and initialize it with 25

## int

Integers are the primary data-type for number storage. int stores a 16-bit (2-byte) value. This yields a range of -32,768 to 32,767 (minimum value of -2^15 and a maximum value of (2^15) - 1).

The int size varies from board to board. On the Arduino Due, for example, an int stores a 32-bit (4-byte) value. This yields a range of -2,147,483,648 to 2,147,483,647 (minimum value of -2^31 and a maximum value of (2^31) - 1).

Example:

int counter = 32 ;// declaration of variable with type int and initialize it with 32.

## Unsigned int

Unsigned int’s (unsigned integers) are the same as int in the way that they store a 2 byte value. Instead of storing negative numbers, however, they only store positive values, yielding a useful range of 0 to 65,535 (2^16) - 1). The Due stores a 4 byte (32-bit) value, ranging from 0 to 4,294,967,295 (2^32 - 1).

Example:

Unsigned int counter= 60 ; // declaration of variable with type unsigned int and initialize it with 60.

## Word

On the Uno and other ATMEGA based boards, a word stores a 16-bit unsigned number. On the Due and Zero, it stores a 32-bit unsigned number.

Example

word w = 1000 ;//declaration of variable with type word and initialize it with 1000.

Long Long variables are extended size variables for number storage, and store 32 bits (4 bytes), from 2,147,483,648 to 2,147,483,647.

Example:

Long velocity= 102346 ;//declaration of variable with type Long and initialize it with 102346

## Unsigned long

Unsigned long variables are extended size variables for number storage and store 32 bits (4 bytes). Unlike standard longs, unsigned longs will not store negative numbers, making their range from 0 to 4,294,967,295 (2^32 - 1).

Example:

Unsigned Long velocity = 101006 ;// declaration of variable with type Unsigned Long and initialize it with 101006.

## Short

A short is a 16-bit data-type. On all Arduinos (ATMega and ARM based), a short stores a 16-bit (2-byte) value. This yields a range of -32,768 to 32,767 (minimum value of -2^15 and a maximum value of (2^15) - 1).

Example:

short val= 13 ;//declaration of variable with type short and initialize it with 13

## Float

Data type for floating-point number is a number that has a decimal point. Floating-point numbers are often used to approximate the analog and continuous values because they have greater resolution than integers.

Floating-point numbers can be as large as 3.4028235E+38 and as low as 3.4028235E+38. They are stored as 32 bits (4 bytes) of information.

Example:

float num = 1.352;//declaration of variable with type float and initialize it with

1.352.

## Double

On the Uno and other ATMEGA based boards, Double precision floating-point number occupies four bytes. That is, the double implementation is exactly the same as the float, with no gain in precision. On the Arduino Due, doubles have 8-byte (64 bit) precision

Example:

double num = 45.352 ;// declaration of variable with type double and initialize it with 45.352.

## STEPS TO UPLOAD THE PROGRAM IN 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) anda USB cable.

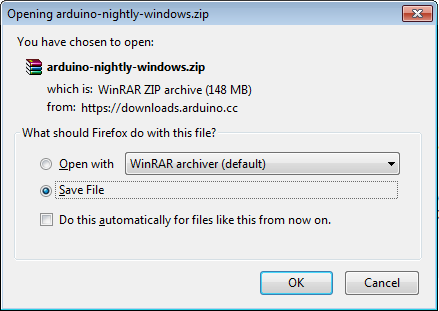
In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind youwould connect to a USB printer as shown in the following image.



Fig 4.1 USB Cable

**Step 2:** Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page 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.



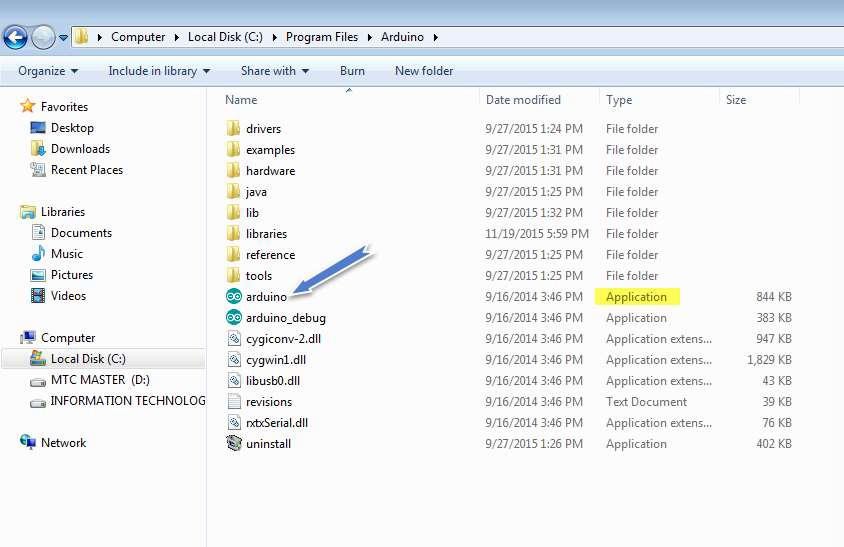
**Step 3:** Power up your board.

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). Doubleclick the icon to start the IDE.

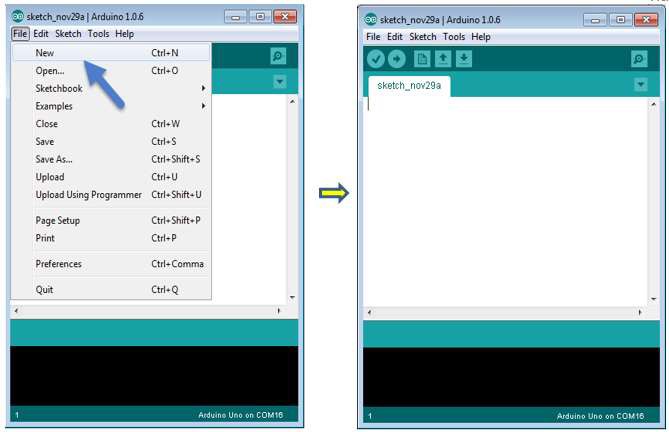


**Step 5:** Open your first project.

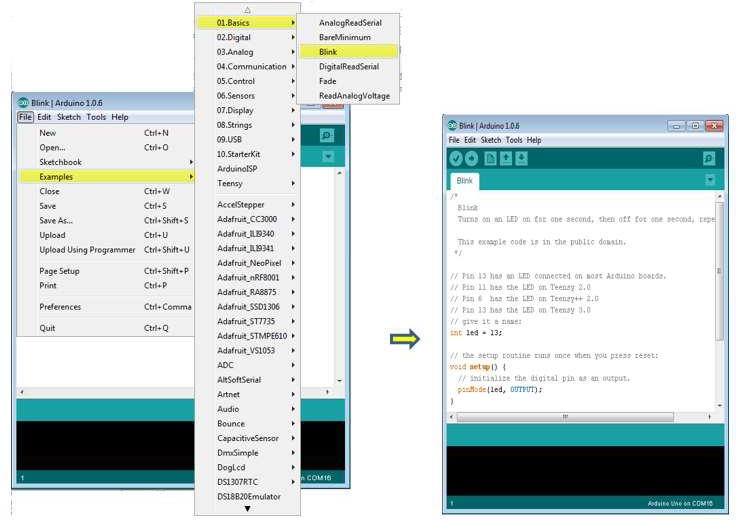
Once the software starts, you have two options: Create a new project.

Open an existing project example.

To create a new project, select File --> New.To open



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

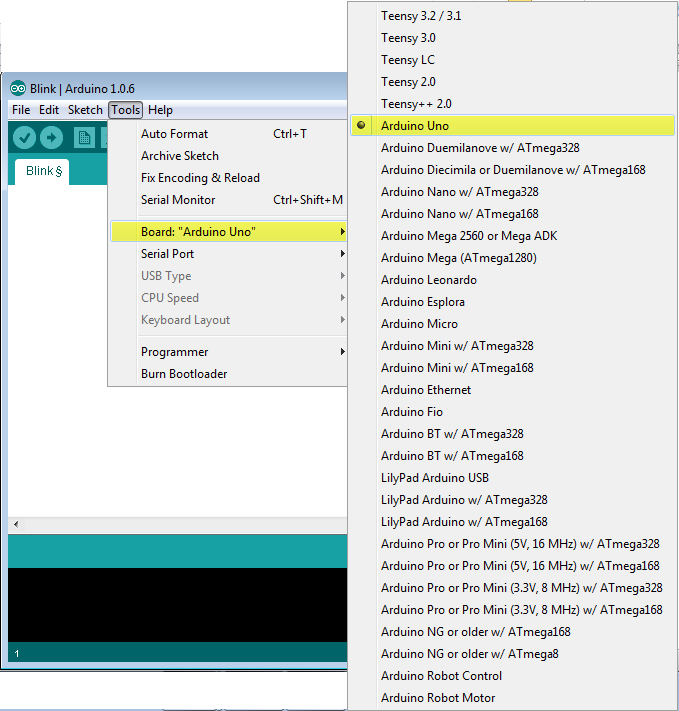


Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

**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.

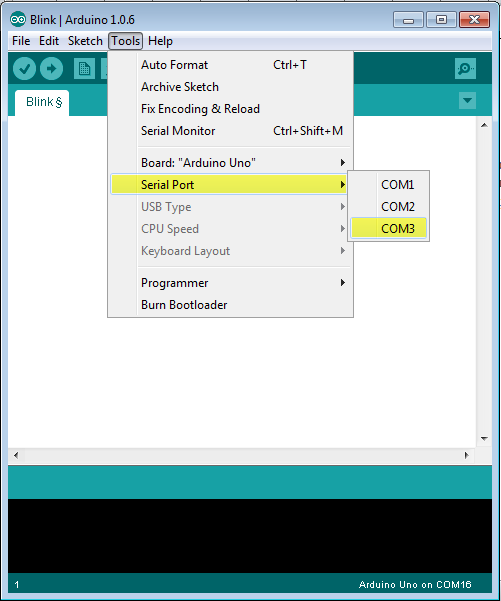
Go to Tools -> Board and select your board



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

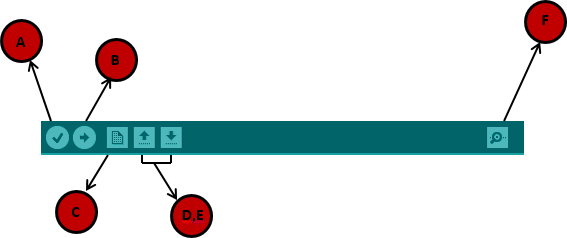
**Step 7:** Select your serial port.

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.

Note: If you have an Arduino Mini, NG, or other board, you need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software.

## ARDUINO PROGRAMMING STRUCTURE

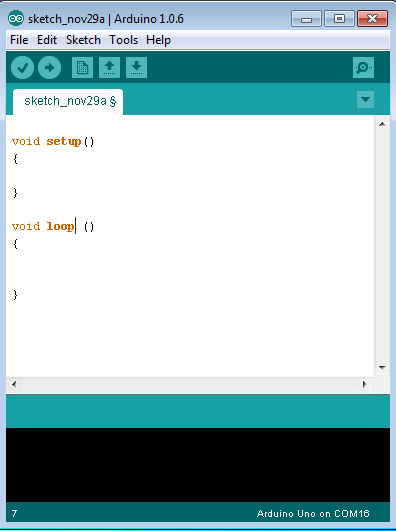
In this chapter, we will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open- source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

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 secutively, allowing your program to change and respond. Use it to actively control the Arduino board.

INPUT

OUTPUT

RETURN

CONCLUSION

Finger print and face based security is providing higher security then existing system. And GSM will also provide security if someone open the locker of authenticate user. The message will be directly gone on authenticate person mobile. He found out someone is try to open his locker. Biometric is an emerging area with many opportunities for growth. Possibly in the near future, you need not to have remembered PINs and passwords and keys in your bags or pockets will be things of the past. There is no security system that is completely out of spoofing. Every system is subject to breakable.

REFRENCES

[1] Prabhakar S Pankanti S, and Jain, A.K “Biometric recognition: Security and Privacy Concern” Security and Privacy, IEEE Volume: 1 Issue: 2.

[2] Sagar S. Palsodkar, Prof S.B Patil “Biometric and GSM Based Security for lockers “International Journal of Engineering Research and Application ISSN: 2248-9622, Vol.4, December2014.

[3] Anil k. Jain, Ling Hong, Sharath Pankanti, Ruud Bolle “An Identity-Authentication System using Fingerprints” .IEEE Vol.85 No.9 September1997.

[4] Anil k. Jain, Salil Prabhakar Ling Hong “A multichannel approach to fingerprint classification” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, April 1999.

[5] \*Dr. V. Vaidehi, K. Gayathri \*S. Vignesh “Efficient face detection and recognition using block independent component analysis and clustering IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011.

[6] Mary Lourde R and Dushyant Khosla “Fingerprint Identification in Biometric Security Systems” International Journal of Computer and Electrical Engineering, Vol. 2, No. 5, October, 2010.

[7] P. Viola, M. Jones. “Rapid object detection using a Boosted cascade of simple features”. In : IEEE Conference on Computer Vision and Pattern Recognition, pp. 511-518, 2001.