***TABLE OF CONTENTS***

Contents Page no.

1. Short or Executive Summary. 1
2. Introduction. 2
3. List of Components. 3
4. Arduino UNO. 4
5. GT511c3 Fingerprint Module. 12
6. Solenoid Lock. 14
7. 16x2 Display. 16
8. Resistors. 23
9. Darlington TIP 121 TIP Transistors. 27
10. 1N4007 Diode. 29
11. 9v Battery. 30
12. Connecting solenoid Lock with Arduino. 31
13. Connecting GT511c3 with Arduino. 32
14. Connecting 16x2 LCD with Arduino. 34
15. Arduino IDE. 36
16. Source Code. 37
17. Flow Chart. 39
18. Future Scope. 40
19. References. 41

LIST OF FIGURES

1. Arduino Board.
2. Arduino Configuration.
3. Communication with Arduino UNO through computers.
4. Arduino Power Ports.
5. Arduino USB overcurrent Protection.
6. GT511c3 Fingerprint Module.
7. Solenoid Lock.
8. 16x2 LCD.
9. LCD Pin configuration.
10. LCD Rows and Columns.
11. Some Important Pins of LCD.
12. Connecting LCD with Arduino(4 bit config.)
13. Resistors.
14. Resistors schematic and working.
15. Transistors.
16. 1N 4007 Diode.
17. 9v Battery.
18. Schematic diagram of connecting solenoid with Arduino.
19. Schematic diagram of connecting GT511c3 with Arduino.
20. Schematic diagram of connecting 16x2 LCD with Arduino.
21. Arduino IDE diagram.
22. Flowchart.
23. ESP8266 and Bluetooth with Arduino.

***EXECUTIVE SUMMARY***

Security has been playing a key role in many of our places like offices, institutions, libraries,

laboratories etc. in order to keep our data confidentially so that no other unauthorized person

could have an access on them.

Nowadays, at every point of time, we need security systems for protection of valuable data

and even money. This paper presents a fingerprint based door opening system which

provides security which can be used for many banks, institutes and various organizations

etc..,. There are other methods of verifying authentication through password, RFID but this

method is most efficient and reliable. To provide perfect security to the bank lockers and to

make the work easier, this project is taking help of two different technologies viz.

EMBEDDED SYSTEMS and BIOMETRICS. Unauthorized access is prohibited by

designing a lock that stores the fingerprints of one or more authorized users. Fingerprint is

sensed by sensor and is validated for authentication. If the fingerprint matches, the door will

be opened automatically otherwise it will remain locked.

***INTRODUCTION***

Security is of primary concern and in this busy, competitive world, human cannot find ways

to provide security to his confidential belongings manually.

Instead, he finds an alternative which can provide a full fledged security as well as atomized.

In the ubiquitous network society, where individuals can easily access their information

anytime and anywhere, people are also faced with the risk that others can easily access the

same information anytime and anywhere. Because of this risk, personal identification

technology, which can distinguish between registered legitimate users and imposters, is now

generating interest. Generally passwords, identification cards and PIN verification

techniques are being used but the disadvantage is that the passwords could be hacked and a

card may be stolen or lost. The most secured system is fingerprint recognition because a

fingerprint of one person never matches the other. Biometrics studies commonly include

fingerprint, face, iris, voice, signature, and hand geometry recognition and verification.

Many other modalities are in various stages of development and assessment. Among these

available biometric traits fingerprint proves to be one of the best traits providing good

mismatch ratio, high accurate in terms of security and also reliable.

Fingerprints are the oldest and most widely used form of biometric identification and robust

security system. The use of fingerprint for identification has been employed in law

enforcement for about a century. A much broader application of fingerprint is for personal

authentication, for instance to access a computer, a network, an ATM machine, a car or a

home. Electronic lock using fingerprint recognition system is a process of verifying the

fingerprint image to open the electronic lock. This project highlights the development of

fingerprint verification.Verification is completed by comparing the data of authorized

fingerprint image with incoming fingerprint image. Then the information of incoming

fingerprint image will undergo the comparison process to compare with authorized

fingerprint image.

***LIST OF COMPONENTS.***

* Arduino UNO.
* GT511c3 Fingerprint Module.
* Solenoid Lock.
* 16x2 LCD Display.
* Resistors.
* 120 ohm
* 220 ohm
* 330 ohm
* Darlington Transistor (TIP 121).
* IN 4007 Diode.
* 9v battery.
* Jumper Wire.

***ARDUINO UNO***

**What is Arduino UNO ?**

The **Arduino Uno** is a microcontroller board based on the ATmega328. Arduino is an open-

source, prototyping platform and its simplicity makes it ideal for  hobbyists to use as well as

professionals. The Arduino Uno has 14 digital input/output pins (of which 6 can be used as

PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power

jack, an ICSP header, and a reset button. It contains everything needed to support the

microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-

DC adapter or battery to get started.

****

"Uno" means one in Italian and is named to mark the upcoming release of Arduino

1.0. The Arduino Uno and version 1.0 will be the reference versions of Arduno,

moving forward. The Uno is the latest in a series of USB Arduino boards, and the

reference model for the Arduino platform.

The Arduino Uno differs from all preceding

boards in that it does not use the FTDI USB-

to-serial driver chip. Instead, it features the

Atmega8U2 microcontroller chip

programmed as a USB-to-serial converter.

The Arduino platform has become quite popular with people just starting out with

electronics, and for good reason. Unlike most previous programmable circuit boards, the

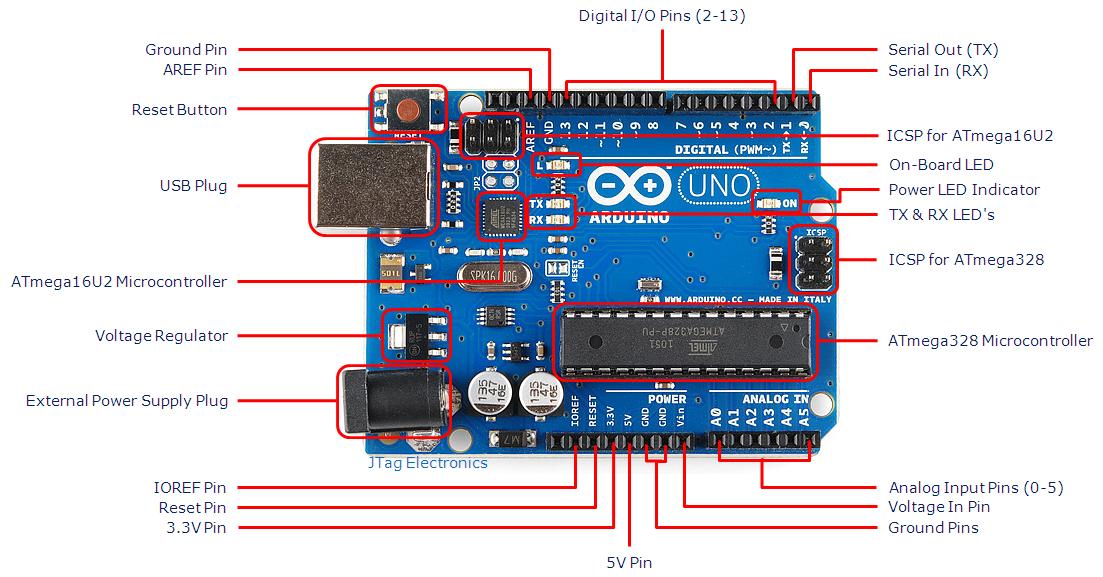
Arduino does not need a separate piece of hardware (called a programmer) in order to load

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 out the functions of the micro-controller into a

more accessible package.



### **Power (USB / Barrel Jack)**

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can

be powered from a USB cable coming from your computer or a wall power supply ([like this](https://www.sparkfun.com/products/8269))

that is terminated in a barrel jack.

The USB connection is also how you will load code onto your Arduino board. The recommended voltage for most Arduino models is between 6 and 12 Volts.

### **Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)**

The pins on your Arduino are the places where you connect wires to construct a circuit

(probably in conjuction with a [breadboard](https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/) and some [wire](https://learn.sparkfun.com/tutorials/working-with-wire). They usually have black plastic

‘headers’ that allow you to just plug a wire right into the board. The Arduino has several

different kinds of pins, each of which is labeled on the board and used for different functions.

* **GND** : Short for ‘Ground’. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
* **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
* **Analog** : The area of pins under the ‘Analog In’ label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a [temperature sensor](https://www.sparkfun.com/products/10988)) and convert it into a digital value that we can read.
* **Digital** : Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
* **PWM** : You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM), these pins as being able to simulate analog output (like fading an LED in and out).
* **AREF** : Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

### **Reset Button**

Just like the original Nintendo, the Arduino has a reset button **.** Pushing it will

temporarily connect the reset pin to ground and restart any code that is loaded on the

Arduino. This can be very useful if your code doesn’t repeat, but you want to test it multiple

times. Unlike the original Nintendo however, blowing on the Arduino doesn’t usually fix any

problems.

### **Power LED Indicator**

Just beneath and to the right of the word “UNO” on your circuit board, there’s a tiny LED

next to the word ‘ON’. This LED should light up whenever you plug your Arduino into

a power source. If this light doesn’t turn on, there’s a good chance something is wrong. Time

to re-check your circuit!

### **TX RX LEDs**

TX is short for transmit, RX is short for receive. These markings appear quite a bit in

electronics to indicate the pins responsible for [serial communication](https://learn.sparkfun.com/tutorials/serial-communication). In our case, there are

two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1,

and a second time next to the TX and RX indicator LEDs . These LEDs will give us

some nice visual indications whenever our Arduino is receiving or transmitting data (like

when we’re loading a new program onto the board).

### **Main IC**

The black thing with all the metal legs is an IC, or Integrated Circuit . Think of it as the

brains of our Arduino. The main IC on the Arduino is slightly different from board type to

board type, but is usually from the ATmega line of IC’s from the ATMEL company. This

can be important, as you may need to know the IC type (along with your board type) before

loading up a new program from the Arduino software. This information can usually be found

in writing on the top side of the IC. If you want to know more about the difference between

various IC’s, reading the datasheets is often a good idea.

### **Voltage Regulator**

The voltage regulator  is not actually something you can (or should) interact with on the

Arduino. But it is potentially useful to know that it is there and what it’s for. The voltage

regulator does exactly what it says – it controls the amount of voltage that is let into the

Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that

might harm the circuit. Of course, it has its limits, so don’t hook up your Arduino to

anything greater than 20 volts.

**Why Arduino ?**

Thanks to its simple and accessible user experience, Arduino has been used in thousands of

different projects and applications. The Arduino software is easy-to-use for beginners, yet

flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and

students use it to build low cost scientific instruments, to prove chemistry and physics

principles, or to get started with programming and robotics. Designers and architects build

interactive prototypes, musicians and artists use it for installations and to experiment with

new musical instruments. Makers, of course, use it to build many of the projects exhibited at

the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children,

hobbyists, artists, programmers - can start tinkering just following the step by step

instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical

computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and

many others offer similar functionality. All of these tools take the messy details of

microcontroller programming and wrap it up in an easy-to-use package. Arduino also

simplifies the process of working with microcontrollers, but it offers some advantage for

teachers, students, and interested amateurs over other systems:

* **Inexpensive**- Arduino boards are relatively inexpensive compared to other microcontroller

platforms. The least expensive version of the Arduino module can be assembled by hand,

and even the pre-assembled Arduino modules cost less than Rs. 1500

* **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and

Linux operating systems. Most microcontroller systems are limited to Windows.

* **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for

beginners, yet flexible enough for advanced users to take advantage of as well. For teachers,

it's conveniently based on the Processing programming environment, so students learning to

program in that environment will be familiar with how the Arduino IDE works.

* **Open source and extensible software** - The Arduino software is published as open source

tools, available for extension by experienced programmers. The language can be expanded

through C++ libraries, and people wanting to understand the technical details can make the

leap from Arduino to the AVR C programming language on which it's based. Similarly, you

can add AVR-C code directly into your Arduino programs if you want to.

* **Open source and extensible hardware** - The plans of the Arduino boards are published

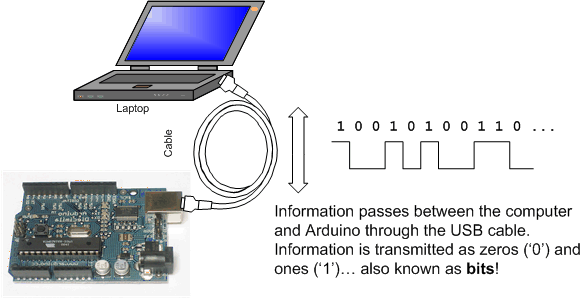
under a Creative Commons license, so experienced circuit designers can make their own

version of the module, extending it and improving it. Even relatively inexperienced users can

build the [breadboard version of the module](https://www.arduino.cc/en/Main/Standalone) in order to understand how it works and save

money.

**Communication :**

 The Arduino Uno has a number of

facilities for communicating with a computer, another Arduino, or other microcontrollers.

The ATmega328 provides UART TTL (5V) serial communication, which is available on

digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial

communication over USB and appears as a virtual com port to software on the computer.

The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed.

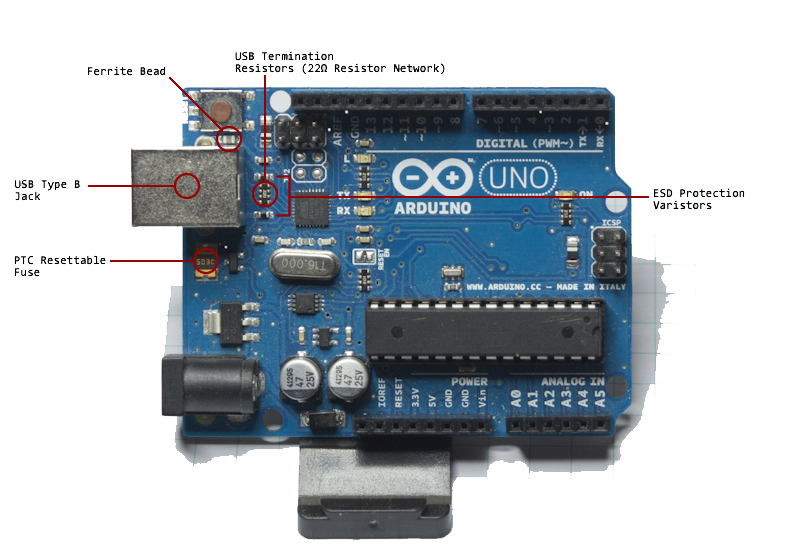
However, on Windows, a .inf file is required. The Arduino software includes a serial monitor

which allows simple textual data to be sent to and from the Arduino board. The RX and TX

LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and

USB connection to the computer (but not for serial communication on pins 0 and 1).

**Automatic (Software) Reset**

 Rather than requiring a physical press of the reset button before an upload, the Arduino Uno

is designed in a way that allows it to be reset by software running on a connected computer.

One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the

reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken

low), the reset line drops long enough to reset the chip. The Arduino software uses this

capability to allow you to upload code by simply pressing the upload button in the Arduino

environment. This means that the bootloader can have a shorter timeout, as the lowering of

DTR can be well-coordinated with the start of the upload.

This setup has other implications.

When the Uno is connected to either a computer running Mac OS X or Linux, it resets each

time a connection is made to it from software (via USB). For the following halfsecond or so,

the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e.

anything besides an upload of new code), it will intercept the first few bytes of data sent to

the board after a connection is opened. If a sketch running on the board receives one-time

configuration or other data when it first starts, make sure that the software with which it

communicates waits a second after opening the connection and before sending this data. The

Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the

trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be

able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line

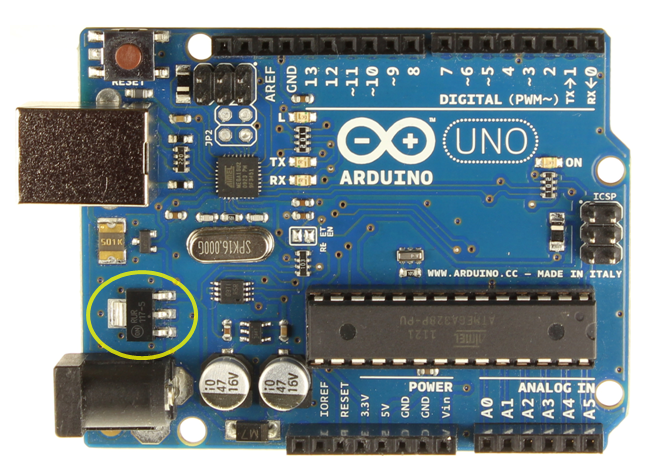
**USB Overcurrent Protection**

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from

shorts and overcurrent. Although most computers provide their own internal protection, the

fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port,

the fuse will automatically break the connection until the short or overload is removed.



***GT511C3 FINGERPRINT MODULE***

**What is GT511C3 fingerprint module?**

The GT-511C3 FPS (fingerprint scanner) is a small embedded module that consists of an

optical sensor mounted on a small circuit board. The optical sensor scans a fingerprint and

the microcontroller and software provides the modules functionality which automatically

processes the scanned fingerprint.



**Pins 1 and 2** are 3.3V TTL pins used to

communicate with the FPS module. The

default baud rate is **9600bps** after power on.

**Pin 1** is the transmit pin of the UART on the FPS

(UART Tx) and transmits a logic high of up to a

maximum of 3.3V.

**Pin 2** is the receive pin of the UART on the FPS

(UART Rx) and can receive a logic high level of

up to 3.3V. The voltage level sent to this pin from

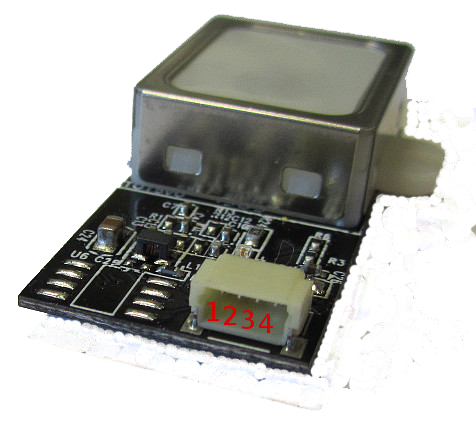
a microcontroller needs to be reduced when

working with 5V microcontrollers.

The module can be used in projects for identifying

users, for example allowing access through a door

by scanning a finger instead of using a key.



**Pin 3** is the common GND or 0V pin of the FPS module.

**Pin 4** is the 5V input to the FPS module used to power it. This value can be between 4.5V

and 6V.

An **electromagnetic lock**, **magnetic lock**,

or **maglock** is a [locking device](https://en.wikipedia.org/wiki/Lock_(security_device)) that consists of

an [electromagnet](https://en.wikipedia.org/wiki/Electromagnet) and an armature plate. There are

two main types of electric locking devices.

Locking devices can be either "fail safe" or "fail

secure". A fail-secure locking device remains

locked when power is lost. Fail-safe locking

devices are unlocked when de-energized. Direct

pull electromagnetic locks are inherently fail-safe.

***SOLENOID LOCK***



Typically the electromagnet portion of the lock is attached to the door frame and a mating

armature plate is attached to the door. The two components are in contact when the door is

closed. When the electromagnet is energized, a [current](https://en.wikipedia.org/wiki/Electric_current) passing through the electromagnet

creates a magnetic flux that causes the armature plate to attract to the electromagnet, creating

a locking action. Because the mating area of the electromagnet and armature is relatively

large, the force created by the magnetic flux is strong enough to keep the door locked even

under stress.

The most basic type of electronic lock is a [magnetic lock](https://en.wikipedia.org/wiki/Magnetic_lock) (informally called a "mag lock").

A large electro-magnet is mounted on the door frame and a corresponding armature is

mounted on the door. When the magnet is powered and the door is closed, the armature is

held fast to the magnet. Mag locks are simple to install and are very attack-resistant. One

drawback is that improperly installed or maintained mag locks can fall on people and also

that one must unlock the mag lock to both enter and to leave. This has caused fire

marshals to impose [strict rules](https://en.wikipedia.org/wiki/Fire_code) on the use of mag locks and access control practice in general.

# **Technical specs**

* 12VDC (you can use 9-12 DC volts, but lower voltage results in weaker/slower operation)
* Draws 650mA at 12V, 500 mA at 9V when activated
* Designed for 1-10 seconds long activation time
* Max Dimensions: 41.85mm / 1.64" x 53.57mm / 2.1" x 27.59mm / 11.08"
* Dimensions: 23.57mm / 0.92" x 67.47mm / 2.65" x 27.59mm / 11.08"
* Wire length: 222.25mm / 8.75"
* Weight: 147.71g

LCD (Liquid Crystal

Display) screen is an

electronic display module

and find a wide range of

applications. A 16x2 LCD

display is very basic

module and is very

commonly used in various

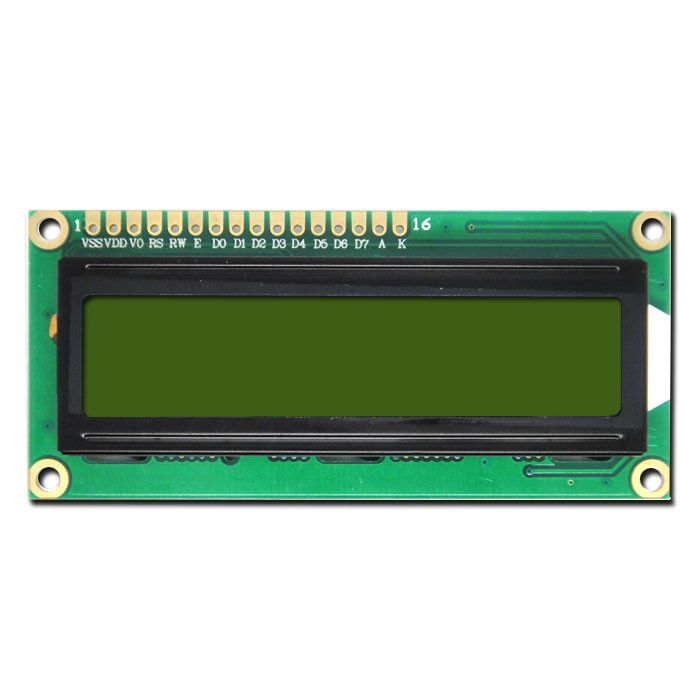
devices and circuits. These

modules are preferred

over [seven segments](http://www.engineersgarage.com/content/seven-segment-display) and

other multi segment [LED](http://www.engineersgarage.com/content/led)s.

***16X2 LCD***



The reasons being: LCDs are economical; easily programmable; have no limitation of

displaying special & even [custom characters](http://www.engineersgarage.com/microcontroller/8051projects/create-custom-characters-LCD-AT89C51) (unlike in seven segments), [animations](http://www.engineersgarage.com/microcontroller/8051projects/display-custom-animations-LCD-AT89C51) and so

on. Many products we see in our daily life have LCD's with them. They are used to show

status of the product or provide interface for inputting or selecting some process. Washing

machine, microwave,air conditioners and mat cleaners are few examples of products that

have character or graphical LCD's installed in them.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this

LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely,

Command and Data.

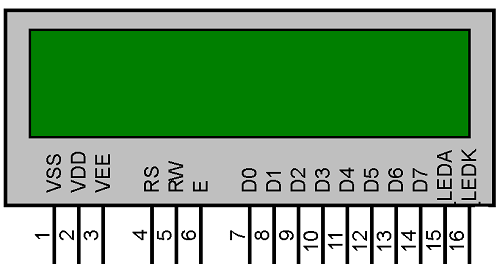
The command register stores the command instructions given to the LCD. A command is an

instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting

the cursor position, controlling display etc. The data register stores the data to be displayed on

the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to

learn more about internal structure of a [LCD](http://www.engineersgarage.com/insight/how-lcd-works).



LCD display will have 16 Pins and the programming

approach is also the same like other display. These

pin are used to configure the LCD display and work

properly and show animation or text efficiently and

neatly.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Pin No.** | **Pin Name** | **Pin Type** | **Pin Description** | **Pin Connection** |
| 1 | Pin 1 | Ground | Source Pin | This is a ground pin of LCD | Connected to the ground of the MCU/ Power source |
| 2 | Pin 2 | VCC | Source Pin | This is the supply voltage pin of LCD | Connected to the supply pin of Power source |
| 3 | Pin 3 | V0/VEE | Control Pin | Adjusts the contrast of the LCD. | Connected to a variable POT that can source 0-5V |
| 4 | Pin 4 | Register Select | Control Pin | Toggles between Command/Data Register | Connected to a MCU pin and gets either 0 or 1.  0 -> Command Mode  1-> Data Mode | |
| 5 | Pin 5 | Read/Write | Control Pin | Toggles the LCD between Read/Write Operation | Connected to a MCU pin and gets either 0 or 1.  0 -> Write Operation  1-> Read Operation | |
| 6 | Pin 6 | Enable | Control Pin | Must be held high to perform Read/Write Operation | Connected to MCU and always held high. | |
| 7 | Pin 7-14 | Data Bits (0-7) | Data/Command Pin | Pins used to send Command or data to the LCD. | In 4-Wire Mode  Only 4 pins (0-3) is connected to MCU  In 8-Wire Mode  All 8 pins(0-7) are connected to MCU | |
| 8 | Pin 15 | LED Positive | LED Pin | Normal LED like operation to illuminate the LCD | Connected to +5V | |
| 9 | Pin 16 | LED Negative | LED Pin | Normal LED like operation to illuminate the LCD connected with GND. | Connected to ground | |

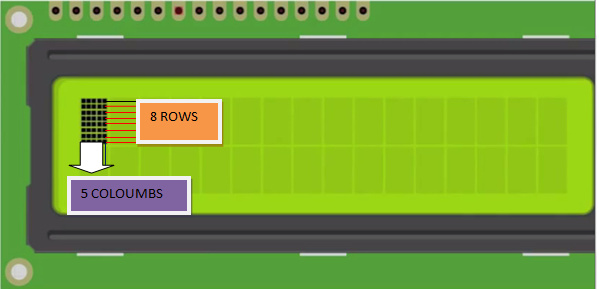
Character LCD's have a controller

build in to them named HD44780.

We actually talk with thiscontroller

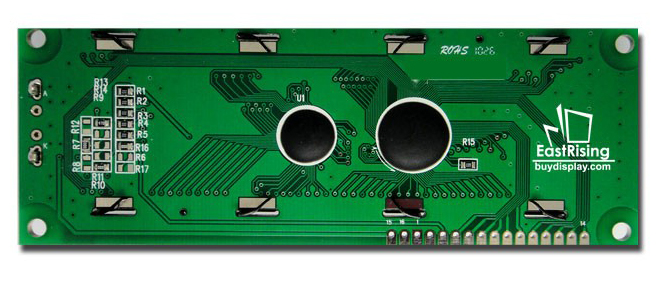
in order to display character on the

LCD screen.

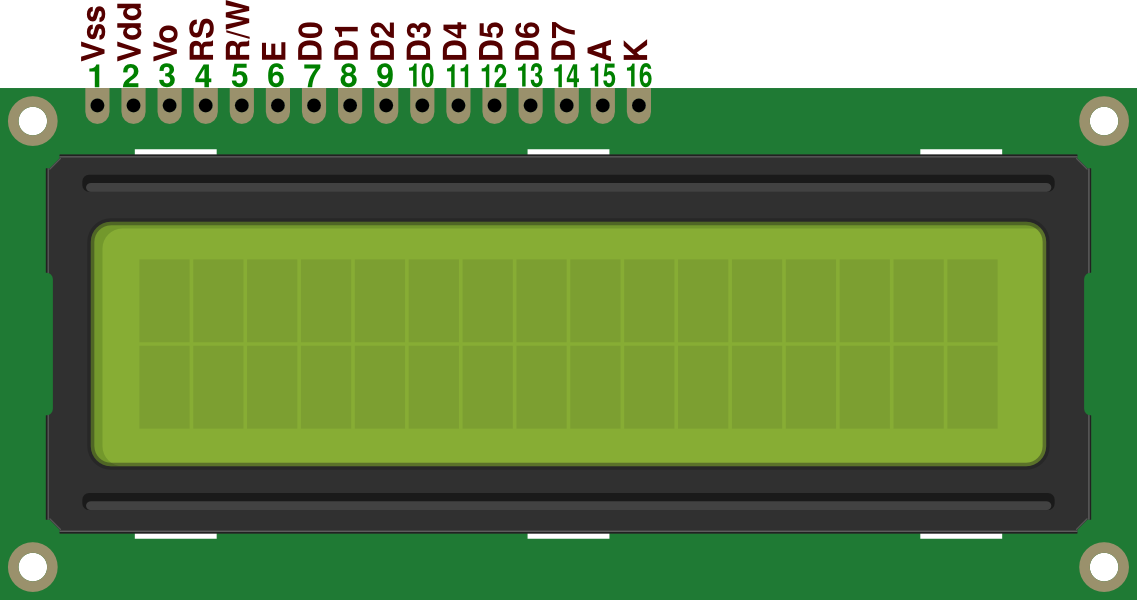


HD44780 must be properly handled and initialized before sending any data to it. HD44780

has some registers which are initialized and  manipulated for character displaying on the LCD.These registers are selected by the pins of character LCD.

****

**SOME IMPORTANT PINS OF LCD DISPLAY.**

****

**Rs(Register select)**  
Register select selects the HD44780 controller registers. It switches between Command and

data register.

* Command Register
* Data Register

**Command Register**  
When we send commands to lcd these commands go to Command register and are processed

their. Commands with their full description are given in the picture below. When Rs=0

command register is selected.

**Data Register**  
When we send Data to lcd it goes to data register and is processed their. When Rs=1 data

register is selected.

   
**Rw(Read - Write)**  
Rw pin is used to read and write data to HD44780 data and command registers. When Rw=1

we can read data from lcd. When Rw=0 we can write to lcd.

**En(Enable signal)**  
When we select the register Rs(Command and Data) and set Rw(read -  write) and placed the

raw value on 8-data lines, now its time to execute the instruction. By instruction i mean the

8-bit data or 8-bit command present on Data lines of lcd. For sending the final

data/command present on the data lines we use this enable pin. Usually it remains en=0 and

when we want to execute the instruction we make it high en=1 for some mills seconds. After

this we again make it ground en=0.

​**V0 (Set Lcd contrast)**

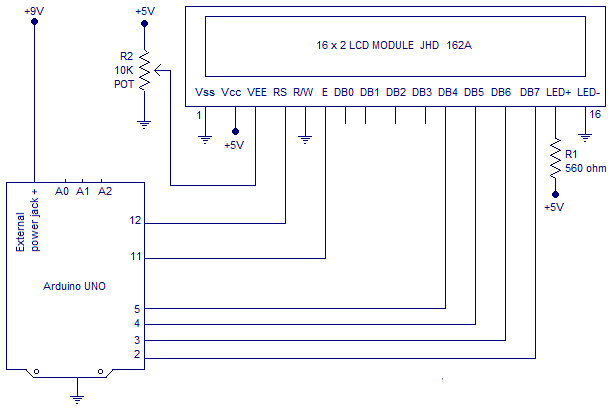
To set lcd display sharpness use this pin. Best way is to use variable resistor such as

potentiometer a variable current makes the character contrast sharp. Connect the output of

the potentiometer to this pin. Rotate the potentiometer knob forward and backward to adjust

the lcd contrast.

### **Difference between 4-bit and 8-bit Lcd Mode**



Character Lcd's can be used in 4-bit and 8-bit mode. Before you send commands and data to

your lcd. Lcd must first be initialized. This initialization is very important for lcd that are

made by Hitachi because they use HD44780 driver chip sets. Hd44780 Chip set  first has to

be initialized before using it. If you don't initialize it properly you will see nothing on your

lcd.

​  
**For 8-bit mode, this is done as follows:**

1. Wait more than 15 mill secs after power is applied.

2. Write command 0x30 to LCD and wait 5 milli seconds for the instruction to complete.

3. Write command 0x30 to LCD and wait 160 micro seconds for instruction to complete.

4. Write command 0x30 AGAIN to LCD and wait 160 micro seconds or Poll the Busy Flag.

**In 4-bit mode the high nibble is sent first before the low nibble and the En pin is toggled each time four bits is sent to the LCD. To initialize in 4-bit mode:**

1. Wait more than 15 mill secs after power is applied.

2. Write command 0x03 to LCD and wait 5 msecs for the instruction to complete.

3. Write command 0x03 to LCD and wait 160 usecs for instruction to complete.

4. Write command 0x03 AGAIN to LCD and wait 160 usecs (or poll the Busy Flag).

***RESISTORS***



A **resistor** is a [passive](https://en.wikipedia.org/wiki/Passivity_(engineering)) [two-terminal](https://en.wikipedia.org/wiki/Terminal_(electronics)) [electrical component](https://en.wikipedia.org/wiki/Electronic_component) that implements [electrical resistance](https://en.wikipedia.org/wiki/Electrical_resistance) as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, [bias](https://en.wikipedia.org/wiki/Biasing) active elements, and terminate [transmission lines](https://en.wikipedia.org/wiki/Transmission_line), among other uses. High-power resistors that can dissipate many [watts](https://en.wikipedia.org/wiki/Watt) of electrical power as heat may be used as part of motor controls, in power distribution systems, or as test loads for [generators](https://en.wikipedia.org/wiki/Electric_generator).

Fixed resistors have resistances that only change slightly with temperature, time or operating

voltage. Variable resistors can be used to adjust circuit elements (such as a volume control or

a lamp dimmer), or as sensing devices for heat, light, humidity, force, or chemical activity.

Resistors are common elements of [electrical networks](https://en.wikipedia.org/wiki/Electrical_network) and [electronic circuits](https://en.wikipedia.org/wiki/Electronic_circuit) and are

ubiquitous in [electronic equipment](https://en.wikipedia.org/wiki/Electronics). Practical resistors as discrete components can be

composed of various compounds and forms. Resistors are also implemented within [integrated circuits](https://en.wikipedia.org/wiki/Integrated_circuits).

The electrical function of a resistor is specified by its resistance: common commercial

resistors are manufactured over a range of more than nine [orders of magnitude](https://en.wikipedia.org/wiki/Orders_of_magnitude). The nominal

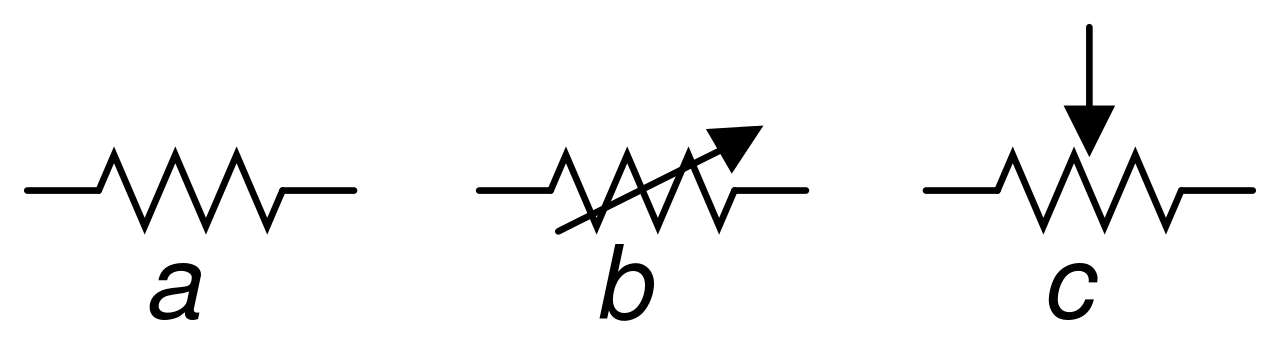
value of the resistance falls within the [manufacturing tolerance](https://en.wikipedia.org/wiki/Engineering_tolerance#Electrical_component_tolerance), indicated on the component.

The unit of resistance is called the Ohm, which is usually shortened to Ω the Greek letter

Omega. Because an Ohm is a low value of resistance (it doesn't resist much at all), we also

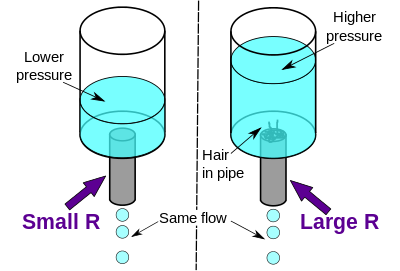
give the values of **resistors** in kΩ (1000 Ω) and MΩ (1000,000 Ω).

**Electronic Sysmbol and Notation.**

****

(a) resistor (b) rheostat (variable resistor) (c) potentiometer

**Theory of Operation**

****

The [hydraulic analogy](https://en.wikipedia.org/wiki/Hydraulic_analogy) compares

electric current flowing through

circuits to water flowing through

pipes. When a pipe (left) is

clogged with hair (right), it takes

a larger pressure to achieve the

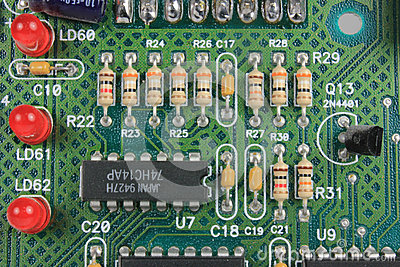
same flow of water.

Pushing electric current through a

large resistance is like pushing

water through a pipe clogged with hair: It requires a larger push ([voltage](https://en.wikipedia.org/wiki/Voltage)) to drive the same

flow ([electric current](https://en.wikipedia.org/wiki/Electric_current)).This phenomenon can be best defined by the ohm’s law of electricity.



Resistor is one of most

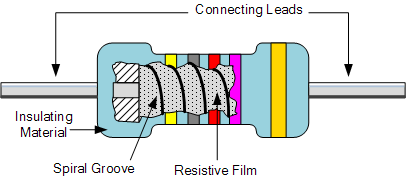
commomly used electronics

component.It can found in all

types of circuit like

controlles,remotes ,etc.

**OHM’s law :**

****

The behaviour of an ideal resistor is dictated by the relationship specified by [Ohm's law](https://en.wikipedia.org/wiki/Ohm%27s_law):

{\displaystyle V=I\cdot R.}

Ohm's law states that the voltage (V) across a resistor is proportional to the current (I), where

the constant of proportionality is the resistance (R). For example, if a 300 [ohm](https://en.wikipedia.org/wiki/Ohm) resistor is

attached across the terminals of a 12 volt battery, then a current of 12 / 300 =

0.04 [amperes](https://en.wikipedia.org/wiki/Ampere) flows through that resistor.

Practical resistors also have some [inductance](https://en.wikipedia.org/wiki/Inductance) and [capacitance](https://en.wikipedia.org/wiki/Capacitance) which affect the relation

between voltage and current in [alternating current](https://en.wikipedia.org/wiki/Alternating_current) circuits.

The [ohm](https://en.wikipedia.org/wiki/Ohm_(unit)) (symbol: [Ω](https://en.wikipedia.org/wiki/%CE%A9)) is the [SI](https://en.wikipedia.org/wiki/International_System_of_Units) unit of [electrical resistance](https://en.wikipedia.org/wiki/Electrical_resistance), named after [Georg Simon Ohm](https://en.wikipedia.org/wiki/Georg_Simon_Ohm).

An ohm is equivalent to a [volt](https://en.wikipedia.org/wiki/Volt) per [ampere](https://en.wikipedia.org/wiki/Ampere). Since resistors are specified and manufactured

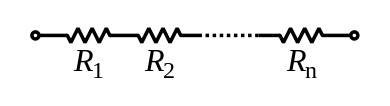
over a very large range of values, the derived units of milliohm (1 mΩ = 10−3 Ω), kilohm (1

kΩ = 103 Ω), and megohm (1 MΩ = 106 Ω) are also in common usage.

**Series and Parallel Resistors.**

The total resistance of resistors connected in series is the sum of their individual resistance

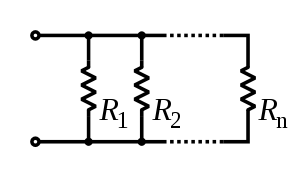
values.

[](https://en.wikipedia.org/wiki/File:Resistors_in_series.svg)

{\displaystyle R\_{\mathrm {eq} }=R\_{1}+R\_{2}+\cdots +R\_{n}.}

The total resistance of resistors connected in parallel is the reciprocal of the sum of the

reciprocals of the individual resistors.

[](https://en.wikipedia.org/wiki/File:Resistors_in_parallel.svg)

{\displaystyle {\frac {1}{R\_{\mathrm {eq} }}}={\frac {1}{R\_{1}}}+{\frac {1}{R\_{2}}}+\cdots +{\frac {1}{R\_{n}}}.}

For example, a 10 ohm resistor connected in parallel with a 5 ohm resistor and a 15 ohm

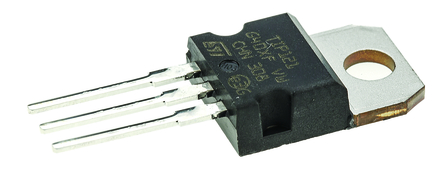
resistor produces 1/1/10 + 1/5 + 1/15 ohms of resistance, or 30/11 = 2.727 ohms.

A resistor network that is a combination of parallel and series connections can be broken up

into smaller parts that are either one or the other. Some complex networks of resistors cannot

be resolved in this manner, requiring more sophisticated circuit analysis. Generally,

the  [matrix methods](https://en.wikipedia.org/wiki/Equivalent_impedance_transforms#2-terminal.2C_n-element.2C_3-element-kind_networks) can be used to solve such problems.



***TIP 121 TRANSISTOR***

In [electronics](https://en.wikipedia.org/wiki/Electronics), the **Darlington transistor** (often called a **Darlington pair**) is a compound

structure consisting of two [bipolar transistors](https://en.wikipedia.org/wiki/Bipolar_transistor) (either integrated or separated devices)

connected in such a way that the current amplified by the first transistor is amplified further

by the second one.[[1]](https://en.wikipedia.org/wiki/Darlington_transistor#cite_note-TAoE-1) This configuration gives a much higher [current](https://en.wikipedia.org/wiki/Electric_current) [gain](https://en.wikipedia.org/wiki/Gain_(electronics)) than each transistor

taken separately and, in the case of integrated devices, can take less space than two

individual transistors because they can use a *shared* collector. Integrated Darlington pairs

come packaged singly in transistor-like packages or as an array of devices (usually eight) in

an [integrated circuit](https://en.wikipedia.org/wiki/Integrated_circuit).

The Darlington configuration was invented by [Bell Laboratories](https://en.wikipedia.org/wiki/Bell_Laboratories) engineer Sidney

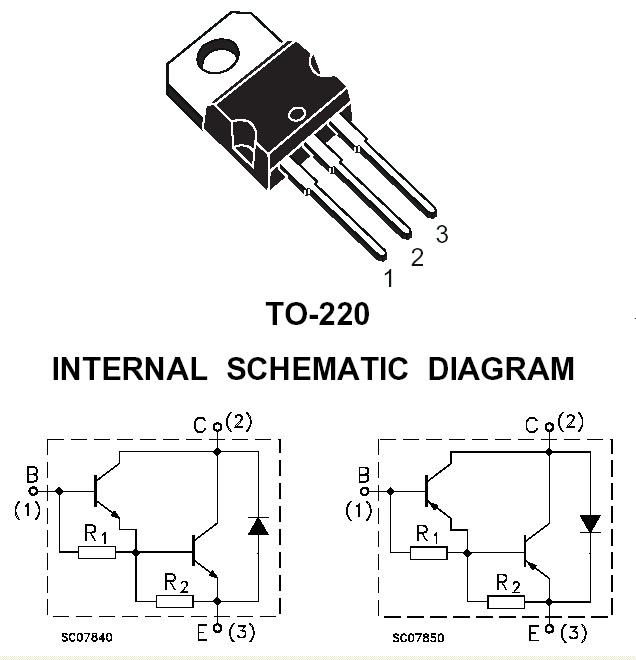
Darlington  in 1953. He [patented](https://en.wikipedia.org/wiki/Patent) the invention of having two or three transistors on a single

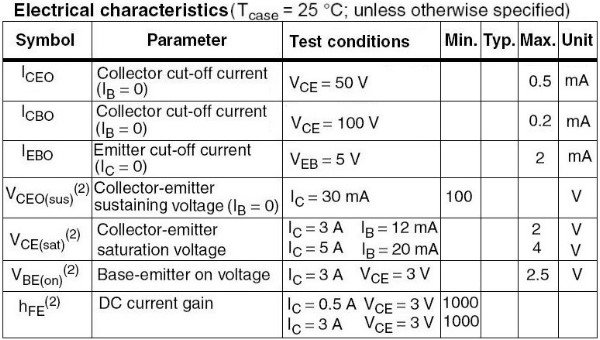
chip sharing a collector.

A similar configuration but with transistors of opposite type (one NPN and one PNP) is

the [Sziklai pair](https://en.wikipedia.org/wiki/Sziklai_pair), sometimes called the "complementary Darlington".

**Schematic diagram of Transistor.**

****

****

The **1N4001 series** (or **1N4000 series**) is a

family of popular 1 A ([ampere](https://en.wikipedia.org/wiki/Ampere)) general-

purpose [silicon](https://en.wikipedia.org/wiki/Silicon) [rectifier](https://en.wikipedia.org/wiki/Rectifier) [diodes](https://en.wikipedia.org/wiki/Diode) commonly

used in [AC adapters](https://en.wikipedia.org/wiki/AC_adapter#AC_adapter) for common

household appliances. Blocking Voltage

varies from 50 to 1000 volts. This

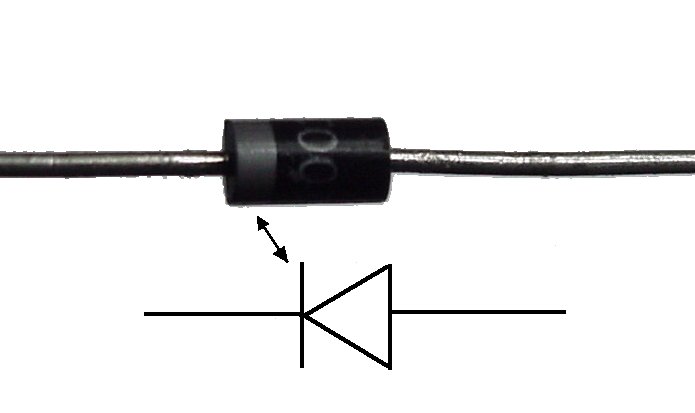
diode series is available in [DO-41](https://en.wikipedia.org/wiki/DO-41) axial

package, [SMA](https://en.wikipedia.org/wiki/DO-214) and [MELF](https://en.wikipedia.org/wiki/Metal_electrode_leadless_face) surface mount

packages.

***1N 4007 DIODE***





The **1N5400 series** is a similarly

popular series for higher-current

3 A applications. These diodes are

typically available in the larger

DO-201AD axial package to

dissipate heat better.

These are fairly low-speed rectifier diodes, being inefficient for square waves of more than 15 kHz. These devices are widely used and recommended.

The series was [second sourced](https://en.wikipedia.org/wiki/Second_source) by many manufacturers. The 1N4000 series were in the Motorola *Semiconductor Data Manual* in 1965, as replacements for 1N2609 through 1N2617.The 1N5400 series were announced in *Electrical Design News* in 1968, along with the now lesser known 1.5 A 1N5391 series.

The table below shows the maximum repetitive reverse blocking voltages of each of the members of the 1N4000 and 1N5400 series.

***9v BATTERY***



The **nine-volt battery**, or **9-volt battery**, in its

most common form was introduced for the

early [transistor radios](https://en.wikipedia.org/wiki/Transistor_radio). It has a rectangular

prism shape with rounded edges and a

polarized snap connector at the top. This type

is commonly used in Walkie Talkies

, [clocks](https://en.wikipedia.org/wiki/Clocks) and [smoke detectors](https://en.wikipedia.org/wiki/Smoke_detectors).

The nine-volt battery format is commonly available in primary carbon-zinc and alkaline

chemistry, in primary lithium iron disulfide, and in rechargeable form in nickel-cadmium,

nickel-metal hydride and lithium-ion. Mercury-oxide batteries of this format, once common,

have not been manufactured in many years due to their mercury content. [Designations](https://en.wikipedia.org/wiki/Battery_nomenclature) for

this format include *NEDA 1604* and *IEC 6F22* (for zinc-carbon) or *MN1604 6LR61* (for

alkaline). The size, regardless of chemistry, is commonly designated *PP3* - a designation

originally reserved solely for carbon-zinc - or in some countries, *E* or *E-block*.

Most nine-volt alkaline batteries are constructed of six individual 1.5V LR61 cells enclosed

in a wrapper. These cells are slightly smaller than LR8D425 [AAAA cells](https://en.wikipedia.org/wiki/AAAA_battery) and can be used in

their place for some devices, even though they are 3.5 mm shorter. Carbon-zinc types are

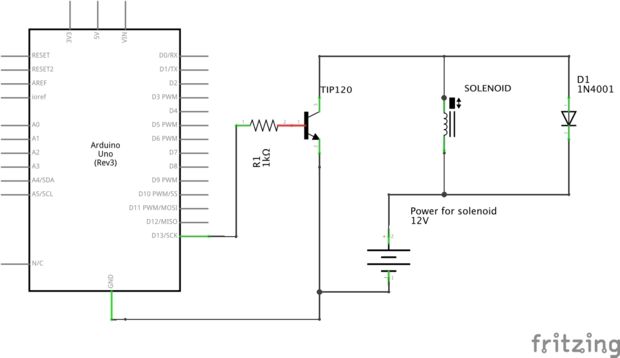
made with six flat cells in a stack, enclosed in a moisture-resistant wrapper to prevent drying.

Primary lithium types are made with 3 cells in series.

In 2007, 9-volt batteries accounted for 4% of alkaline primary battery sales in the US. In

Switzerland in 2008, 9-volt batteries totalled 2% of primary battery sales and 2% of

secondary battery sales.



***CONNECTING SOLENOID WITH ARDUINO***

We connected the output pin of the arduino to the Base leg of the transistor through a 1K

resistor.  The Collector leg of the transistor will be connected to the ground leg of the device

we are driving (our solenoid).  The Emitter leg is connected to the ground channel of our

circuit.

The "ground" leg of the solenoid is connected to the collector leg of the transistor.  The

"power" goes to the high voltage power channel (from our batteries).  I put "ground" and

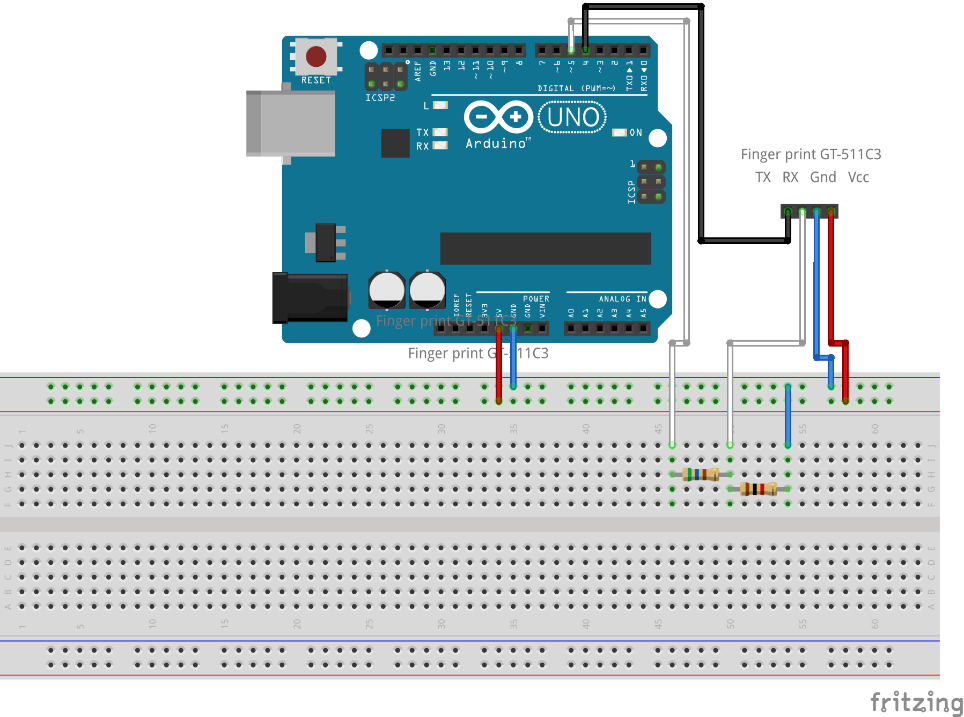
"power" in quotation marks because none of the solenoids I've ever used have been polarized

so it doesn't matter which lead is connected to ground and which goes to power.

The diode connects the power channel to the solenoid-ground-leg/transistor-collector-leg,

preventing the kickback voltage from damaging the circuit.  The diode is polarized and

should be oriented with white/sliver stripe on the power channel side of the connection.



### Arduino Uno / 5V Microcontroller Interface

When connecting to a microcontroller that uses 5V voltage

levels on its pins, a level converter

must be used to reduce the 5V

output from the microcontroller to

the FPS module because the FPS

module can only handle 3.3V on its

UART pins.

A voltage divider consisting of two

resistors can be used as a level

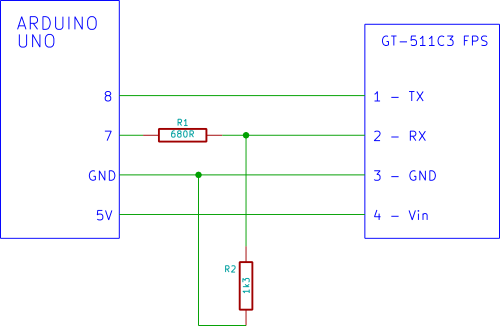
converter to reduce the 5V

incoming signal to 3.3V. The circuit

diagram below shows the

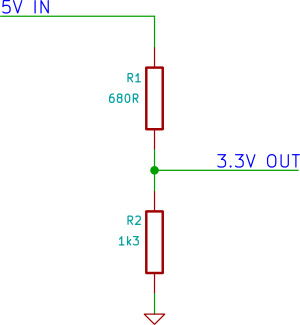
fingerprint scanner module connected to an Arduino Uno.

***CONNECTING GT511c3 WITH ARDUINO***



#### **Voltage Divider**

The circuit below shows the voltage divider from the above circuit diagram. When the Arduino drives its output pin to 5V, the voltage divider reduces this level to 3.3V so that the fingerprint module is not over-driven.

**Voltage Divider / Level circuit.**

The actual calculated voltage output from the above divider circuit is approximately 3.283V.

#### **Choosing Voltage Divider Resistors**

Other resistor combinations can be used to get a value close to 3.3V output as shown in the

table below.

|  |  |  |
| --- | --- | --- |
| R1 | R2 | CALCULATED VOLTAGE |
| 620Ω | 1k2 | 3.3V |
| 680Ω | 1k3 | 3.28V |
| 560Ω | 1k | 3.21V |
| 1k | 1k8 | 3.21V |

The higher the resistance combination of R1 and R2, the more likely that the

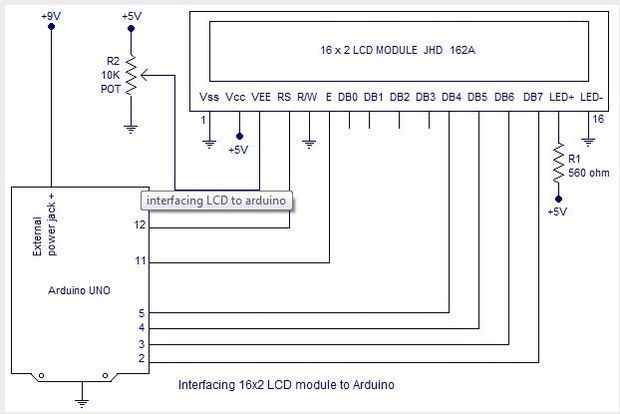
communications between the Arduino and SFM will become unreliable due to interference

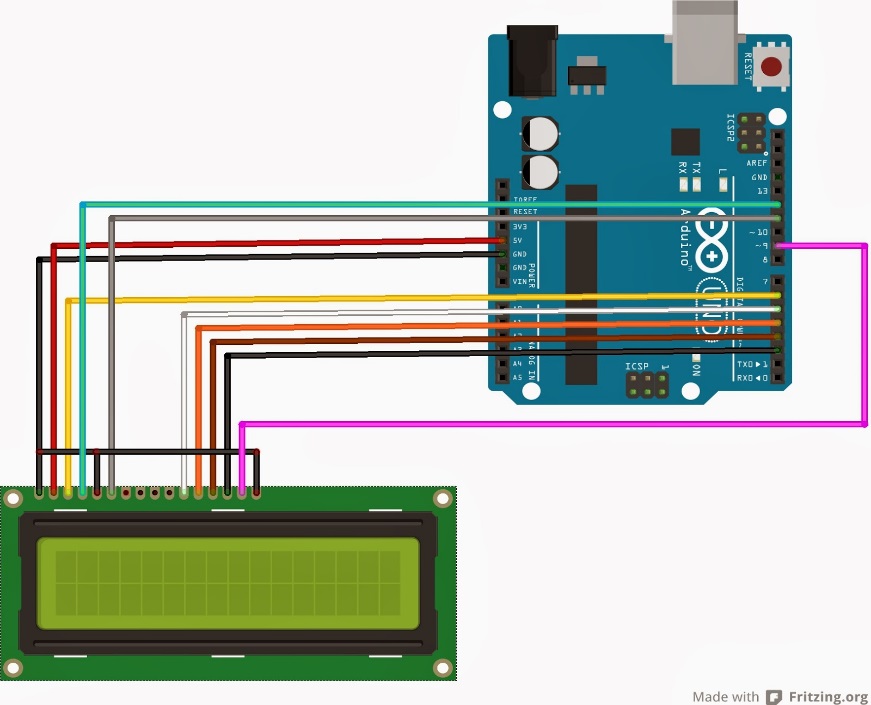
from electrical noise.

The lower the resistance combination of R1 and R2, the more the Arduino pin will be loaded

(the lower the resistance, the more current that will be drawn from the pin by the resistors).

***CONNECTING 16X2 LCD WITH ARDUINO***





Pin1(Vss):Ground pin of the LCD

module.

Pin2(Vcc):+5V supply is given to

this pin

Pin3(VEE):Contrast adjustment

pin. This is done by connecting

the ends of the pin to PWM~ pin

6 which can be act as analog too.

Pin4(RS):Register select pin.The JHD162A has two registers namely command register and

data register. Logic HIGH at RS pin selects data register and logic LOW at RS pin will select

command register. If we make the RS pin HIGH and put a data on the data lines (DB0 to

DB7) it will be recognized as a data. If we make the RS pin LOW and put a data on the data

lines, then it will be taken as a command.

Pin5(R/W): Read/Write modes. This pin is used for selecting between read and write modes.

Logic HIGH at this pin activates read mode and logic LOW at this pin activates write mode.

Pin6(E): This pin is meant for enabling the LCD module. A HIGH to LOW signal at this pin

will enable the module.

Pin7(DB0) to Pin14(DB7): These are data pins. The commands and data are put on these

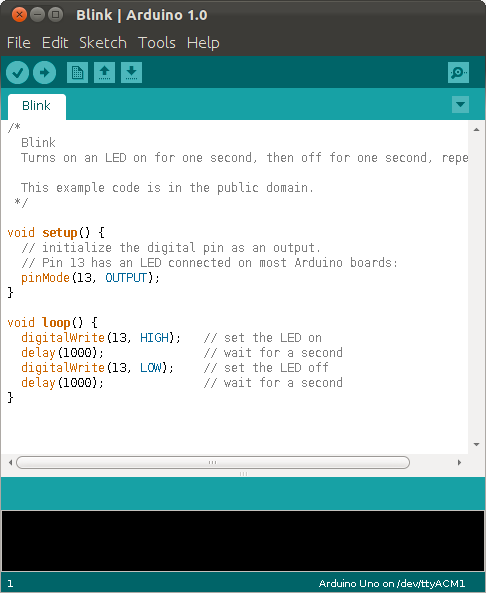
pins.

Pin15(LED+): Anode of the back light LED. When operated on 5V, a 560 ohm resistor

should be connected in series to this pin. In arduino based projects the back light LED can be

powered from the 3.3V source on the arduino board.

Pin16(LED-): Cathode of the back light LED.



The Arduino project provides the

Arduino [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE), which is

a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application

written in the programming

language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It originated from

the IDE for the

languages [*Processing*](https://en.wikipedia.org/wiki/Processing_(programming_language)) and *wiring.*

A program written with the IDE

for Arduino is called

a *sketch*. Sketches are saved on

the development computer as text

files with the file extension *.ino*.

Arduino Software (IDE) pre-1.0

saved sketches with the

extension *.pde*.

***ARDUINO IDE***

The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consist of only two functions:

* *setup()*: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
* *loop()*: After *setup()* has been called, function *loop()* is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

***SOURCE CODE***

#include<LiquidCrystal.h>

#include "FPS\_GT511C3.h"

#include "SoftwareSerial.h"

int contrast=120;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

FPS\_GT511C3 fps(8,9);

int id;

int match;

void setup()

{

analogWrite(6,contrast);

lcd.begin(16,2);

lcd.clear();

lcd.print("WELCOME”");

delay(2500);

lcd.clear();

fps.Open();

fps.SetLED(true);

}

void loop()

{

if (fps.IsPressFinger())

{

fps.CaptureFinger(false);

id = fps.Identify1\_N();

if (id <200)

{

lcd.print("Verified ID : ");

lcd.println(id);

}

else

{

lcd.println("Finger not found");

}

}

else

{

lcd.println("Please press ");

lcd.setCursor(0,1);

lcd.print("ENROLLED finger");

}

delay(100);

}

void unlock()

{

match=id;

if(match==id)

{

digitalWrite(13,HIGH);

delay(6000);

}

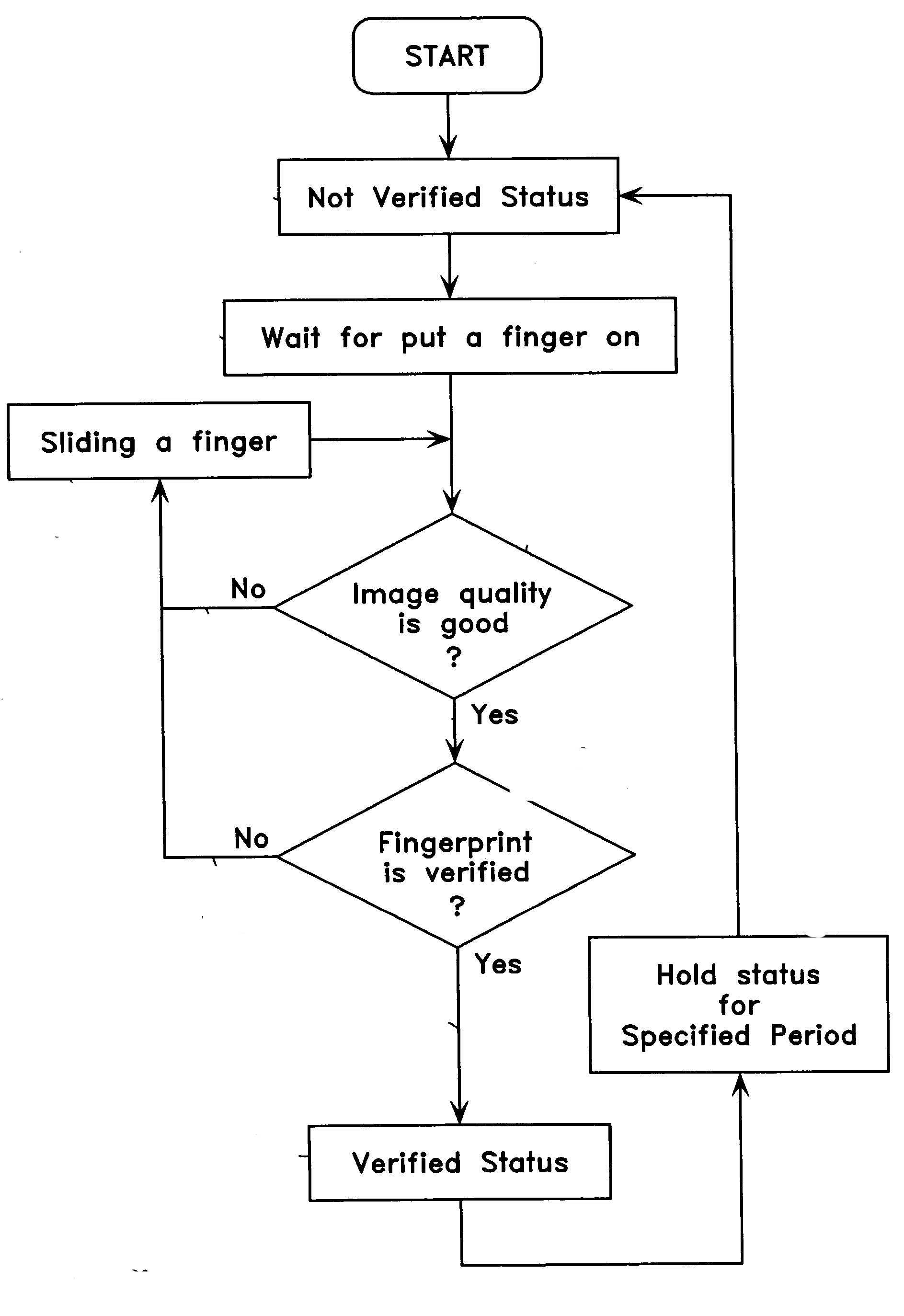
else

{

digitalWrite(13,LOW);

}

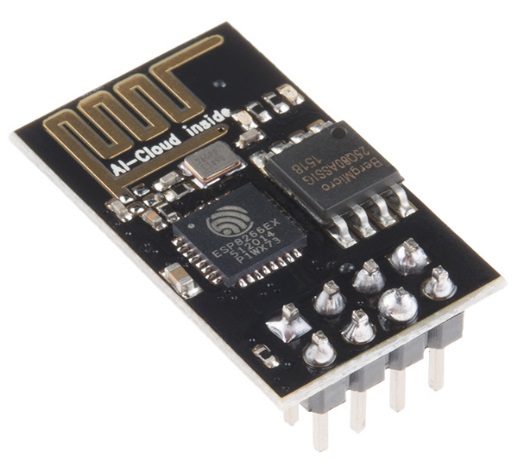
}



***FLOW CHART***

**Show verified in LCD**

**Unlock the lock**

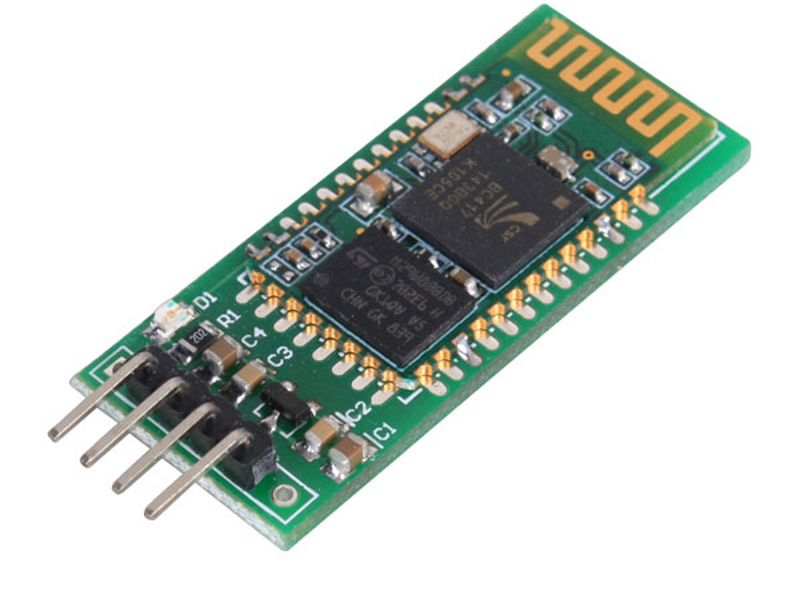


The **ESP8266** is a low-cost [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) chip with full TCP/IP stack and [MCU (microcontroller unit)](https://en.wikipedia.org/wiki/Microcontroller) capability produced by Shanghai-based Chinese manufacturer, [Espressif Systems](https://en.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1).

***FUTURE IMPLEMENTATION***

This WI-FI module can be attached with Arduino Uno based fingerprint lock system.

Then this Arduino based lock system can be controlled through internet and controlled remotely , this increase the flexibility of the life.Door canbe also unlocked without fingerprint .



Bluetooth can be attached also with this assembly to control remotely from short distance.

In theory ,it can be unlocked from the fingerprint or password of personal mobile phones.

List of websites used in this project and got references.

***REFERENCE***

* <https://startingelectronics.org>
* <http://www.instructables.com>
* <http://oppimateriaalit.jamk.fi>
* <https://www.arduino.cc>
* <https://en.wikipedia.org>
* <https://diyhacking.com>
* <https://www.onsemi.com>
* <https://www.sparkfun.com/>
* <http://www.projectsof8051.com>
* https://img1.banggood.com