

A
PROJECT ON
FINGERPRINT BASED DOOR UNLOCK SYSTEM



SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF
THE DIPLOMA IN COMPUTER ENGINEERING

SUBMITTED BY

N.SANJAY	19090-CM-249
D.RAGINI	19090-CM-205
CH.SAICHARAN	19090-CM-252
SADIYA NAAZ	19090-CM-234
T.MANIKANTA	19090-CM-247
M.VENNELA	19090-CM-218
B.UDAY KIRAN	19090-CM-228
MD.SHAHNAWAZ AHMED	19090-CM-225

UNDER THE ESTEEMED GUIDANCE OF
K.SWETHA_{M.TECH}
DEPARTMENT OF COMPUTER ENGINEERING
VEMUGANTI MANOHAR RAO POLYTECHNIC
(Sponsored By Manohara Education Society, Warangal)
Rampur, Warangal, TS-506151
2019-2022

VEMUGANTI MANOHAR RAO POLYTECHNIC

RAMPUR,WARANGAL

2019-2022



CERTIFICATE

This is to certify this report certified “FINGERPRINT BASED DOOR UNLOCK SYSTEM” was carried out by

N.SANJAY	19090-CM-249
D.RAGINI	19090-CM-205
CH.SAICHARAN	19090-CM-252
SADIYA NAAZ	19090-CM-234
T.MANIKANTA	19090-CM-247
M.VENNELA	19090-CM-218
B.UDAY KIRAN	19090-CM-228

MD.SHAH NAWAZ AHMED19090-CM-225

Is partial fulfillment of the requirement for the DIPLOMA IN COMPUTER ENGINEERING, by State Board of Technical Education and Training-Telangana, Hyderabad. Under of esteemed admonishment and has been successfully completed, it was gratified to the extent of his/her knowledge and experience

PROJECT GUIDE

K.SWETHA, M.Tech

EXTERNAL EXAMINER

HEAD OF THE DEPARTMENT

G.RAJAMOULI

PRINCIPAL

V.PRADEEP KUMAR

ACKNOWLEDGEMENT

We are grateful to Sri V. PRADEEP KUMAR Principal of VMR Polytechnic for having given me the opportunity to work on this project. I thank him for making all the facilities available to me.

We would like to thank Sri G. RAJAMOULI Head of the Department, for his continuous encouragement, which lead to the successful completion of my project.

We are thankful to my project guide MRS. K.SWETHA for her immense help during my project who actually encouraged and supported me in completion of my project.

We also thank the faculty of department of Computer Engineering for their support in completion of my project.

We are thankful to my project guide MR. B.KISHAN RAJgaru for his immense help during my project who actually encouraged and supported me in completion of my project.

DOOR UNLOCK SYSTEM USING FINGERPRINT

ABSTRACT :

Our project is based on **IOT (INTERNET OF THINGS)** **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, but this method is most efficient and reliable. To provide perfect security 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.

MATERIALS USED:

- ✓ Arduino chip.
- ✓ Biometric sensor.
- ✓ Solenoid lock.
- ✓ 12V 2AMP DC adapter.
- ✓ Tip transistor.
- ✓ Jumper wires.

Digital Keypad Security Door Lock using Arduino:



Password based Digital Keypad Security Door Lock using Arduino

Often times, we need to secure a room at our home or office (perhaps a secret dexter's laboratory) so that no one can access the room without our permission and ensure protection against theft or loss of our important accessories and assets. There are so many types of security systems present today but behind the scene, for authentication they all relay on fingerprint, retina scanner, iris scanner, face id, tongue scanner, RFID reader, password, pin, patterns, etc. Off all the solutions the low-cost one is to use a password or pin-based system. So, in this project, I have built an **Arduino Keypad Door Lock** which can be mounted to any of your existing doors to secure them with a digital password. Previously, we have also built other interesting door locks which are listed below.

Disadvantages:

The main disadvantage of using keypad locks is Keypad locks can be hacked if the wireless network they are working over has not been set up and secured properly. Keypad locks that do not use any wireless signals cannot be hacked but are usually not a lot safer than standard locks as an experienced locksmith can pick them

1:) Forgetful: You may be the one to forget your keys now and then, and it can be easy to forget your PIN code for the lock and when you're in a rush to get into the room or building or it is night time and dark, you don't want to be changing the code in the middle of the night or when it's raining!

2:) Keep the PIN code safe and the lock clean: Only tell the code to people who you trust, as you don't want a code to your property to be local news. When the lock has been used a few too many times, the coating may start to come away or mucky fingerprints may start to occur on the buttons. Keep the lock maintained and clean to stop unwanted people finding out the code!

3:) Power Failure: Some digital door locks are powered by electricity, if your house or building has a power failure, then the door lock will not work which restricts you from entering the building. Buying a mechanical or battery powered lock will not affect you if there is a power failure.

4:) Limit the PIN Code Length: Some digital door locks have a PIN code length up to 10 digits - this is not what you want! Digital Door Locks will be much more secure if they are only 4 digits long. Purchase a quality lock that you can change the PIN code on, don't buy locks that are provided with a PIN code because people can find out the code.

PROPOSED SYSTEM:

These days office/corporate environment security is a major threat faced by every individual when away from home or at the home. When it comes to security systems, it is one of the primary concerns in this busy competitive world, where human cannot find ways to provide security to his/her confidential belongings manually. Instead, he/she finds an alternative solution which provides better, reliable and atomized security. This is an era where everything is connected through network, where anyone can get hold of information from anywhere around the world. Thus, chances of one's info being hacked are a serious issue. Due to these risks it's very important to have some kind of personal identification system to access one's own information. Now a days, personal identification is becoming an important issue all around. Among mainstream personal identification methods, we mostly see password and identification cards techniques. But it is easy to hack password now and identification cards may get lost, thus making these methods quite unreliable.

There are certain situations which are very annoying like when a person locks himself out of his house or office or he leaves his key inside or sometimes when a thief just breaks the lock and steals everything. These kinds of situations always trouble people who use manual door lock with keys. Although in some places people use smart cards, there might arise a situation when someone loses the card or keeps the card inside. Then in other scenarios there are caretakers for locking houses or offices and keeping the keys safe. But then again there are times when a person in charge of the keys might not be available or has gone to some emergency routine, which can cause unwanted delay for people who need the key straightaway. These are some of the hassles that people might face when using keys or smart cards. That is when our system, fingerprint-based door lock system comes into play. Our design is implemented to provide better securities as users don't need to remember passwords and don't need any sort of keys or cards that often get lost. If someone's fingerprint is authorized in the system, he/she would not face any sort of delays to enter a room. Fingerprint recognition is one of the most secure systems because a fingerprint of one person never matches with others. Therefore, unauthorized access can be restricted by designing a lock that stores the fingerprints of one or more authorized users and unlock the system when a match is found. Bio-metrics authorization proves to be one of the best traits because the skin on our palms and soles exhibits a flow like pattern of ridges on each fingertip which is unique and immutable. This makes fingerprint a unique identification for everyone. The popularity and reliability on fingerprint scanner can be easily guessed from its use in recent hand-held devices like mobile phones and laptops.

PROJECT BASED ON : IOT

INTRODUCTION:

The Internet of Things (IoT) is the network of physical objects - devices, instruments,

vehicles, buildings and other items embedded with electronics, circuits, software, sensors and network connectivity that enable these objects to collect and exchange data. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency and accuracy. The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first internet- connected appliance, able to report its inventory and whether newly loaded drinks were cold. Kevin Ashton (born 1968) is a British technology pioneer who is known for inventing the term “the Internet of Things” to describe a system where the Internet is connected to the physical world via ubiquitous sensors.

IoT is able to interact without human intervention. Some preliminary IoT applications have been already developed in healthcare, transportation, and automotive industries.

IoT technologies are at their infant stages; however, many new developments have occurred in the integration of objects with sensors in the Internet. The development of IoT involves many issues such as infrastructure, communications, interfaces, protocols, and standards.

CONCEPT OF IOT

Kevin Ashton firstly proposed the concept of IoT in 1999, and he referred the IoT as uniquely identifiable connected objects with Radio frequency Identification (RFID) technology.

IoT was generally defined as "dynamic global network infrastructure with self-configuring capabilities based on standards and communication protocols".

(1) Internet of Documents

(2) Internet of Commerce

(3) Internet of Applications

(4) Internet of People

(5) Internet of Things

1. The Internet of Documents - e-libraries, document-based webpages.

2. The Internet of Commerce - e-commerce, e-banking and stock trading websites.

3. The Internet of Applications - Web 2.0

4. The Internet of People - Social networks.

5. The Internet of Things - Connected devices and machines.



Internet of things common definition is defining as:

Internet of things (IOT) is a network of physical objects. The internet is not only a network of computers, but it has evolved into a network of device of all type and sizes, vehicles, smart phones, home appliances, toys, cameras, medical instruments and industrial systems, animals, people, buildings, all connected, all communicating & sharing information based on stipulated protocols in order to achieve smart reorganizations, positioning, tracing, safe and control and even personal real time online monitoring , online upgrade, process control and administration.

We Define IOT into Three Categories as Below: Internet of things is an internet of three things

(i) People to people,

(ii) People to machine /things,

(iii) Things/machine to things/machine, interacting through internet

Internet of Things is referred to the general idea of things, especially everyday objects, that are readable, recognizable, locatable, addressable through information sensing device and/or controllable via the Internet, irrespective of the communication means (whether via RFID, wireless LAN, wide area networks, or other means).

Everyday objects include not only the electronic devices we encounter or the products of higher technological development such as vehicles and equipment but things that we do not ordinarily think of as electronic at all - such as food, clothing, chair, animal, tree, water etc.

CHARACTERISTICS OF IOT:

Interconnectivity: With regard to the IoT, anything can be interconnected with the global information and communication infrastructure.

Things-related Services: The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things

Heterogeneity: The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.

Dynamic Changes : The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.

Enormous Scale : The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet, at Even more critical will be the management of the data generated and their interpretation for application purposes.

Safety : As we gain benefits from the IoT, we must not forget about safety. This includes the safety of our personal data and the safety of our physical well-being. Securing the endpoints.

Connectivity : Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data.

APPLICATIONS OF IOT:

The IoT application covers "smart" environments/spaces in domains such as : Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy.

- ◆ **Remote Control Appliances** : Switching on and off remotely appliances to avoid accidents and save energy.
- ◆ **Weather**: Displays outdoor weather conditions such as humidity, temperature, pressure, wind speed and rain levels with ability to transmit data over long distances
- ◆ **Smart Home Appliances** -Ingredients you need to buy and with all the information available on a Smartphone app. Washing machines allowing you to monitor the laundry remotely, and Kitchen ranges with interface to a Smartphone app allowing remotely adjustable temperature control and monitoring the oven's self-cleaning feature
- ◆ **Safety Monitoring** : Cameras, and home alarm systems making people feel safe in their daily life at home.
- ◆ **Intrusion Detection Systems** : Detection of window and door openings and violations to prevent intruders.
- ◆ **Energy and Water Use** : Energy and water supply consumption monitoring to obtain advice on how to save cost and resources and many more.

ADVANTAGES AND DISADVANTAGES OF IOT:

ADVANTAGES:

- 1. Communication :** Since IoT has communication between devices, in which physical devices (also known as M2M communication) are able to stay connected and hence the total transparency is available with lesser inefficiencies and greater quality. IoT can start a connection without human interference.
- 2. Automation and Control :** Without human involvement, machines are automating and controlling vast amount of information, which leads faster and timely output. IoT can gather information with the use of sensors and make its right decision.
- 3. Monitoring Saves Money and Time :** Since IOT uses smart sensors to monitor various aspects in our daily life for various applications which saves money and time.
- 4. Better Quality of Life :** IoT based applications increases comfort and better management in our daily life; thereby improving the quality of life.
- 5. Home Security :** IoT systems are used in home security which can control by the smartphones.
- 6. New Business Opportunities :** Creates new business for IoT technology, hence increases economic growth and new jobs.
- 7. Better Environment :** Saves natural resources and trees and helps in creating a smart greener and sustainable planet.
- 8. Accurate and Faster :** IoT process and deliver information accurate and faster in the smallest amount of time and minimum utilization of energy.

DISADVANTAGES:

1. Compatibility : As devices from different manufacturers will be interconnected in IoT,

presently; there is no international standard of compatibility for the tagging and monitoring equipment.

2. Complexity : The IoT is a diverse and complex network. Any failure or bugs in the

software or hardware will have serious consequences. Even power failure can cause a lot of inconvenience. In case of network failure, it needs time to restore the service to the customers.

3. Privacy/Security : IoT has involvement of multiple devices and technologies and multiple companies will be monitoring it. Since lot of data related to the context will be transmitted by the smart sensors, there is a high risk of losing private data.

4. Lesser Employment of Menial Staff : With the advent of technology, daily activities are getting automated by using IoT with less human intervention, which in turn causes fewer requirements of human resources. This causes unemployment issue in the society.

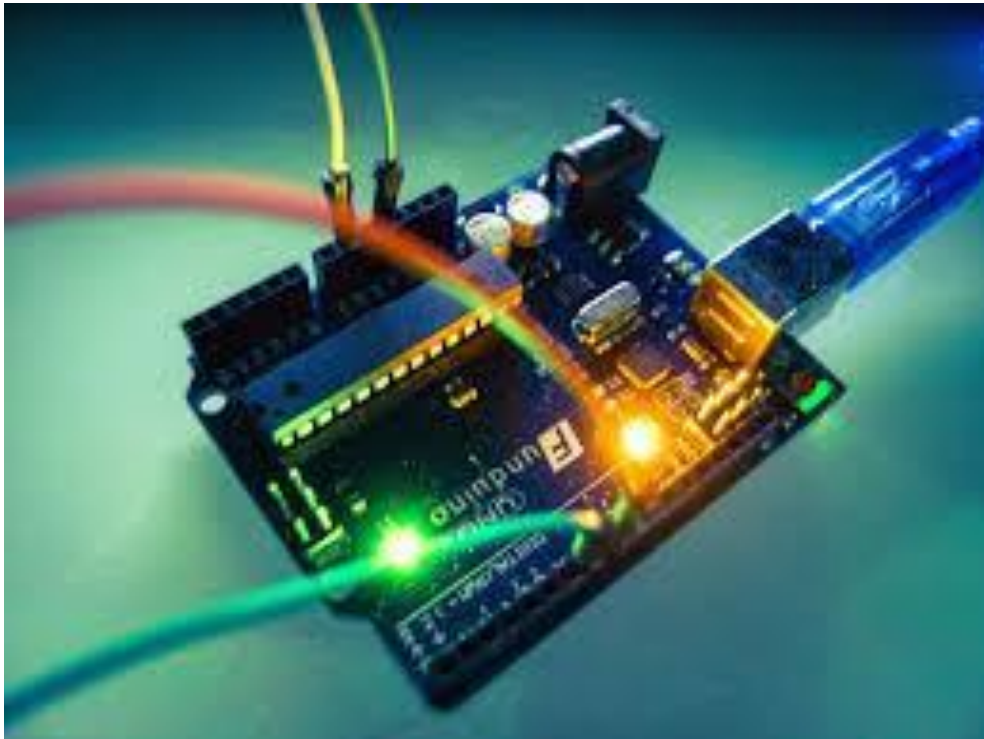
5. Technology Takes Control of Life : Our lives will be increasingly controlled by technology, and will be dependent on it. The younger generation is already addicted to technology for every little thing. With IoT, this dependency will spread amongst generations and in daily routines of users. We have to decide how much of our daily lives are we willing to mechanize and be controlled by technology.

MATERIALS REQUIRED AND MATERIAL ANALYSIS:

- ✓ Arduino chip.
- ✓ Biometric sensor.
- ✓ Solenoid lock.
- ✓ 12v 2amp Dc adapter.
- ✓ Tip transistor.
- ✓ Jumper wires.

ARDUINO CHIP:

Introduction



Arduino is an open-source prototyping platform in electronics based on easy-to-use hardware and software. Subtly speaking, Arduino is a microcontroller based prototyping board which can be used in developing digital devices that can read inputs like finger on a button, touch on a screen, light on a sensor etc. and turning it in to output like switching on an LED, rotating a motor, playing songs through a speaker etc.

The Arduino board can be programmed to do anything by simply programming the microcontroller on board using a set of instructions for which, the Arduino board consists of a USB plug to communicate with your computer and a bunch of connection sockets that can be wired to external devices like motors, LEDs etc.

The aim of Arduino is to introduce the world of electronics to people who have small to no experience in electronics like hobbyists, designers, artists etc.

Arduino is based on open source electronics project i.e. all the design specifications, schematics, software are available openly to all the users. Hence, Arduino boards can be bought from vendors as they are commercially available or else you can make your

own board by if you wish i.e. you can download the schematic from Arduino's official website, buy all the components as per the design specification, assemble all the components, and make your own board.

Hardware and Software

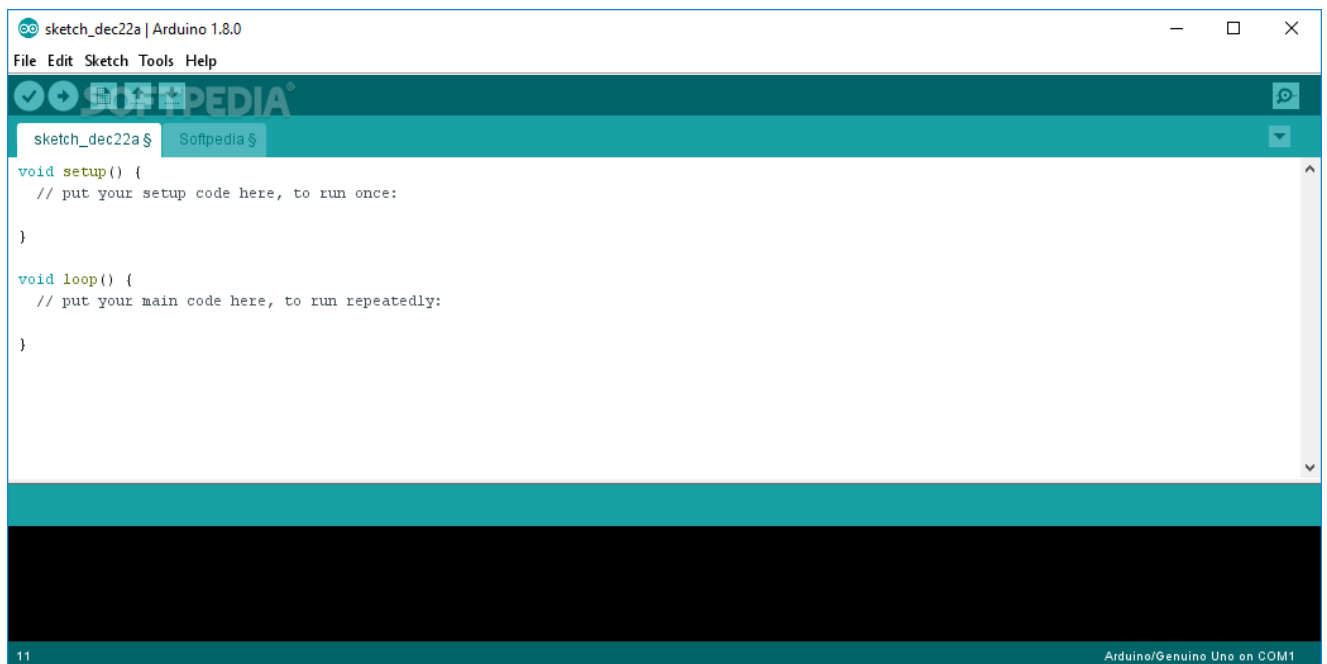


Arduino boards are generally based on microcontrollers from Atmel Corporation like 8, 16 or 32 bit AVR architecture based microcontrollers.

The important feature of the Arduino boards is the standard connectors. Using these connectors, we can connect the Arduino board to other devices like LEDs or add-on modules called Shields.

The Arduino boards also consists of on board voltage regulator and crystal oscillator. They also consist of USB to serial adapter using which the Arduino board can be programmed using USB connection.

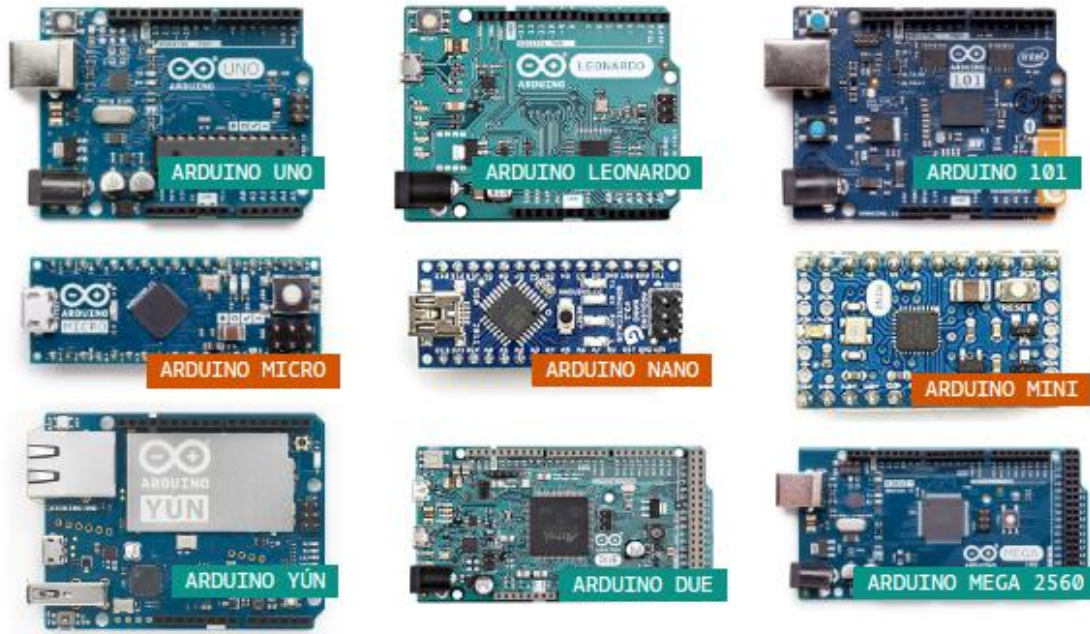
In order to program the Arduino board, we need to use IDE provided by Arduino. The Arduino IDE is based on Processing programming language and supports C and C++.



Types of Arduino Boards

There are many types of Arduino boards available in the market but all the boards have one thing in common: they can be programmed using the Arduino IDE. The reasons for different types of boards are different power supply requirements, connectivity options, their applications etc.

Arduino boards are available in different sizes, form factors, different no. of I/O pins etc. Some of the commonly known and frequently used Arduino boards are Arduino UNO, Arduino Mega, Arduino Nano, Arduino Micro and Arduino Lilypad.



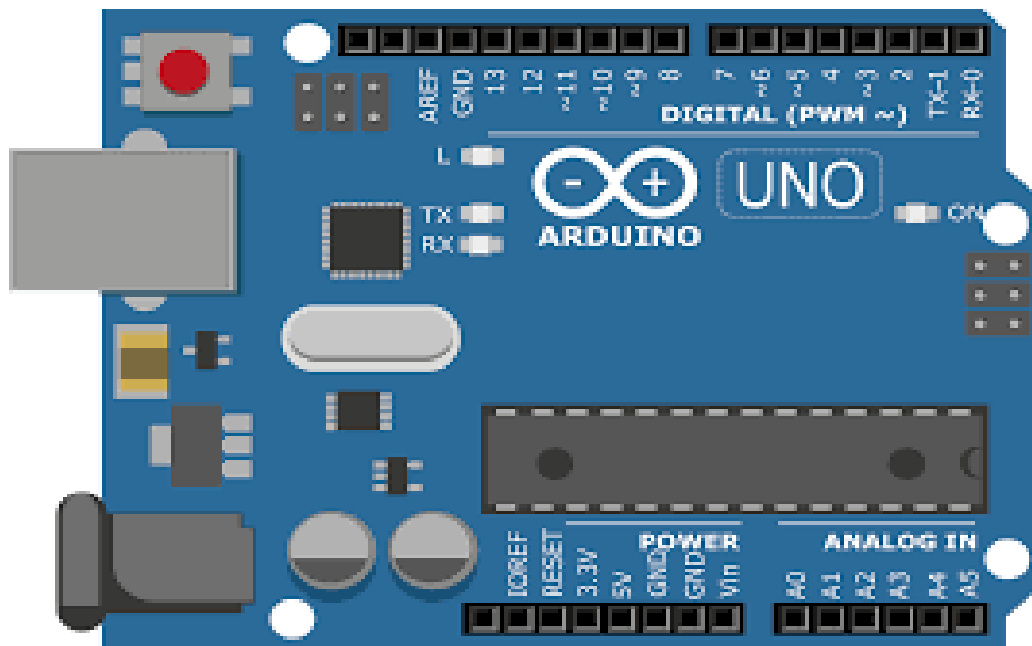
There are add-on modules called Arduino Shields which can be used to extend the functionalities of the Arduino boards. Some of the

commonly used shields are Arduino Proto shield, Arduino WiFi Shield and Arduino Yun Shield.

Arduino UNO

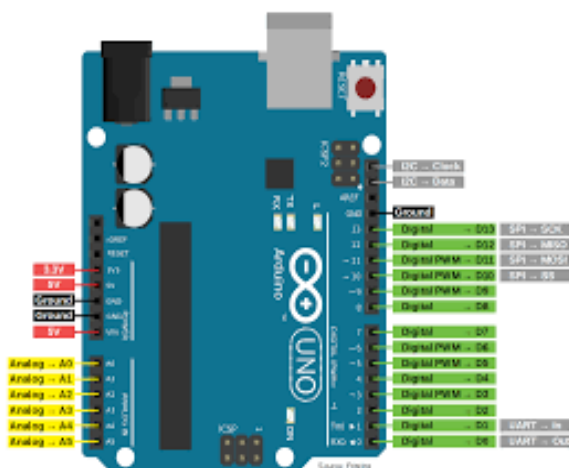
Arduino UNO is a basic and inexpensive Arduino board and is the most popular of all the Arduino boards with a market share of over 50%. Arduino UNO is considered to be the best prototyping board for beginners in electronics and coding.

UNO is based on ATmega328P microcontroller. There are two variants of the Arduino UNO: one which consists of through – hole microcontroller connection and other with surface mount type. Through-hole model will be beneficial as we can take the chip out in case of any problem and swap in with a new one.



Arduino UNO comes with different features and capabilities. As mentioned earlier, the microcontroller used in UNO is ATmega328P, which is an 8-bit microcontroller based on the AVR architecture.

UNO has 14 digital input – output (I/O) pins which can be used as either input or output by connecting them with different external devices and components. Out of these 14 pins, 6 pins are capable of producing PWM signal. All the digital pins operate at 5V and can output a current of 20mA.



Some of the digital I/O pins have special functions which are describe below.

Arduino Uno has 6 analog input pins which can provide 10 bits of resolution i.e. 1024 different values. The analog pins on the Arduino UNO are labelled A0 to A5.

- ✓ Pins 0 and 1 are used for serial communication. They are used to receive and transmit serial data which can be used in several ways like programming the Arduino board and communicating with the user through serial monitor

- ✓ Pins 2 and 3 are used for external interrupts. An external event can be triggered using these pins by detecting low value, change in value or falling or rising edge on a signal.
- ✓ As mentioned earlier, 6 of the 14 digital I/O Pins i.e. 3, 5, 6, 9, 10, and 11 can provide 8-bit PWM output.
- ✓ Pins 10, 11, 12 and 13 (SS, MOSI, MISO AND SCK respectively) are used for SPI communication.
- ✓ Pin 13 has a built-in LED connected to it. When the pin is HIGH, the LED is turned on and when the pin is LOW, it is turned off.

Arduino Uno has 6 analog input pins which can provide 10 bits of resolution i.e. 1024 different values. The analog pins on the Arduino UNO are labelled A0 to A5. By default, all the analog pins can measure from ground to 5V. Arduino UNO has a feature, where it is possible to change the upper end of the range by using the AREF pin but the value should be less than 5V.

Additionally, some analog pins have specialized functionality. Pins A4 and A5 are used for I2C communication. There are different ways in which we can power the Arduino UNO board. The USB cable, which is used to program the microcontroller, can be used as a source of power. There is a power jack, using which an external regulated power supply in the range of 7V – 12V can be supplied. Additionally, the power can also be supplied from a battery through the VIN pin. The UNO board has on-board voltage regulators for 5V and 3.3V, which can be used as power supply for small external devices like LEDs.

This is a brief introduction to Arduino and Arduino UNO board.

R305 Fingerprint Scanner Sensor Module:

Introduction:

R307 is one of the optical fingerprint readers from Hangzhou Grow Technology Co., Ltd. The scanner operates at a voltage of 4.2V~6V and 50 mA with a storage capacity of 1000 impressions. R307 has both UART and USB 2.0 interfaces to communicate with a computer system at a baud rate in multiples of 9600 bps. It is capable of both 1:1 and 1:N matching with FAR (False Acceptance Rate) less than 0.001 percent. The module can scan a live finger in less than 0.5 seconds and supports five security levels (1~5; 5 is highest). The operating temperature range of this sensor is -10°C to 40°C, making it deployable in most of the locations.



The Fingerprint module can be directly interfaced with any microcontroller as well as Arduino Board. This optical biometric fingerprint reader with great features and can be embedded into a variety of end products like access control system, attendance system, safety deposit box, car door locking system.

Features:

1. Integrated image collecting and algorithm chip together, ALL-in-One
2. Fingerprint can conduct secondary development & embedded into a variety of end products
3. Low power consumption, low cost, small size, excellent performance
4. Professional optical technology, precise module manufacturing techniques
5. Good image processing capabilities can successfully capture image up to resolution 500 dpi

Specifications

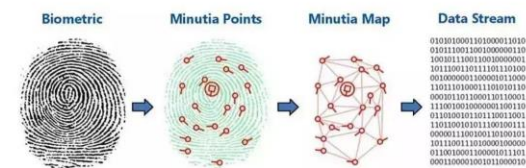
- Fingerprint sensor type: Optical
- Sensor Life: 100 million times
- Static indicators: 15KV Backlight: bright green
- Interface: USB1.1/UART(TTL logical level)
- RS232 communication baud rate: 4800BPS~115200BPS changeable
- Dimension: 55x32x11.5mm
- Image Capture Surface 15—18(mm)

- Verification Speed: 0.3 sec
- Scanning Speed: 0.5 sec
- Character file size: 256 bytes
- Template size: 512 bytes
- Storage capacity: 250
- Security level: 5 (1,2,3,4,5(highest))
- False Acceptance Rate (FAR) :0.0001%
- False Rejection Rate (FRR): 0.1%
- Resolution 500 DPI
- Voltage :3.6-6.0 VDC
- Working current: Typical 90 mA, Peak 150mA
- Matching Method: 1: N
- Operating Environment Temperature: -20 to 45° centigrades

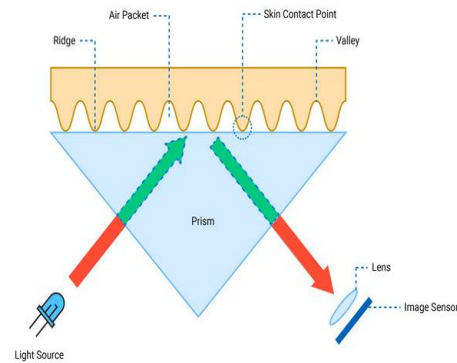
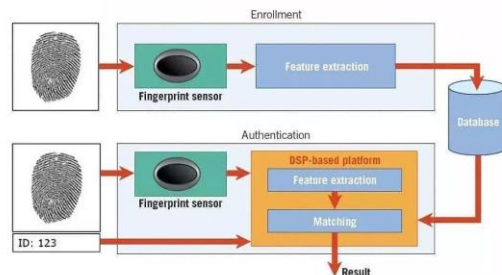
How optical fingerprint scanners work:

The skin of our palm has friction ridges to grab and hold things, and the pattern of these ridges and valleys is also present on the fingertips. A miracle of nature is that this pattern of ridges and valleys is unique for every individual. An impression of our fingerprints is left whenever we grab or hold something due to oil, moisture, dust, and dead cells over the skin. These fingerprints objects are called latent fingerprints

FINGERPRINT SENSING PROCESS



Block diagram of fingerprint process system.



The optical fingerprint readers use the principle of Total Internal Reflection (TRI). An optical fingerprint scanner consists of a prism. On one face of the prism, there is a LED light source. The light enters the prism at a certain angle such that it is reflected from the adjacent face and exits from the third face, where a lens and an image capturing sensor are placed.

When no finger or impression is placed over the sensor, the light transmitted by the LED source is completely reflected off, and the image sensor captures a plain image. However, when there is a fingertip placed over the scanner, some of the light is reflected while some of the light is leaked along the surface of the face of the prism. These are called Evanescent Waves

When no finger or impression is placed over the sensor, the light transmitted by the LED source is completely reflected off, and the image sensor captures a plain image. However, when there is a fingertip placed over the scanner, some of the light is reflected while some of the light is leaked along the surface of the face of the prism. These are called Evanescent Waves

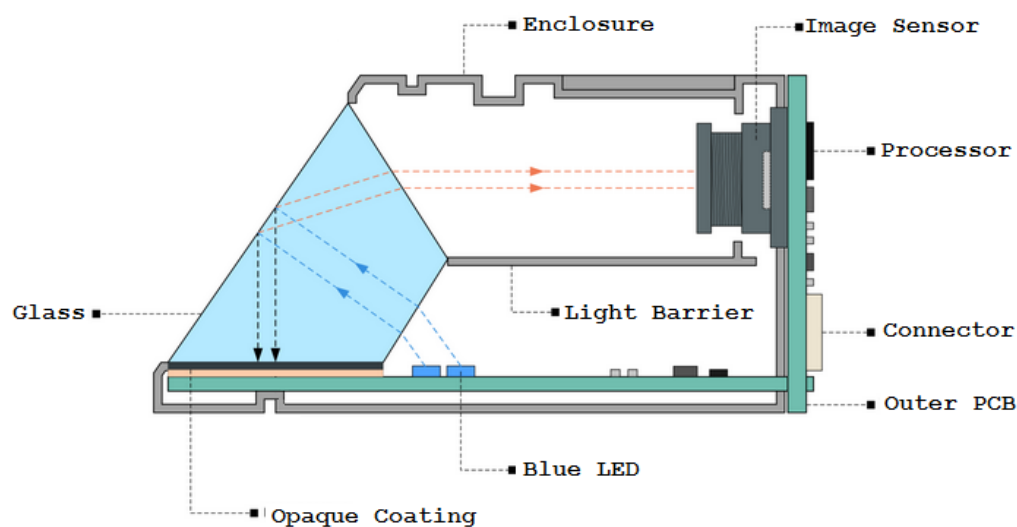
Different materials have different reflective indices and interact differently with the evanescent waves. When the fingertip is placed over the scanner, the ridges are in tight contact with the scanner's surface while the valleys are filled with air pockets. The skin and air have different reflective indices causing different evanescent waves, called Frustrated Total Internal Reflection (FTIR). As a result of different evanescent waves from ridges and valleys, the intensity of the total internally reflected light changes according to the pattern of the ridges and valleys. The image sensor captures a high contrast image recording the changed light intensity pattern, capturing the pattern of ridges and valleys as a high-contrast digital image.

The high-contrast digital image is stored in Flash memory as fingerprint ID according to a predefined template. The template indicates the presence of ridges or valleys at predefined positions in a captured or scanned image. Any fingerprint sensor is designed to perform two processes – enrollment and matching essentially. The process of reading fingerprint impression and storing it according to a predefined template is called enrollment. A

fingerprint reader can enroll several fingerprint IDs depending upon its flash memory and the built-in controller. The enrollment process usually involves confirmation of the fingerprint impression, so it requires scanning the fingerprint twice. Fingerprint IDs store the images in the module.

In fingerprint matching, a new scan is compared with stored fingerprint templates, and if it has the same template as any of the stored impressions, it is confirmed matched. Otherwise, the scan is rejected as unmatched. If the live finger is compared against a specific fingerprint ID, it is called 1:1 matching. If the live finger is compared to match against all fingerprint templates stored in the module, it is called 1:N matching.

R307 sensor assembly:



Physical assembly of R30X fingerprint sensors

R307 has a glass top where a fingertip can be placed for scanning. Below the glass top is placed a prism. The inside of the sensor is divided into two parts using a light barrier. On one side of the light barrier is a PCB consisting of four blue LED lights. At the other side of the light barrier is an image sensor connected with a processor. The outer PCB has the processor, connector, and other circuit elements. The prism, along with blue LEDs and an image sensor, is arranged such that light transmitted by blue LEDs is internally reflected through the prism to the image sensor.

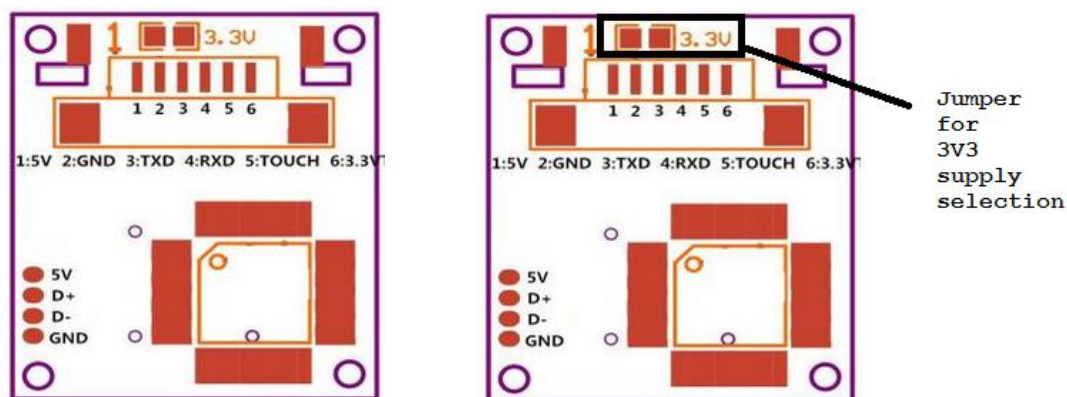
R307 sensor pinout:

The connector of the R307 fingerprint sensor has six terminals. The pin configuration of this connector is as follows.

Pin Number	Pin Name	Type	Description
1	5V	INPUT	Positive Supply (4.2~6V)
2	GND	GROUND	Ground
3	TX	OUT	Data Output
4	RX	INPUT	Data Input
5	Touch	OUT	Finger Detection Signal
6	3.3V	INPUT	Finger Detection Power (3.3~5V)

The pins are arranged in the connector, as shown in the image below.

Pin configuration of R307 fingerprint sensor



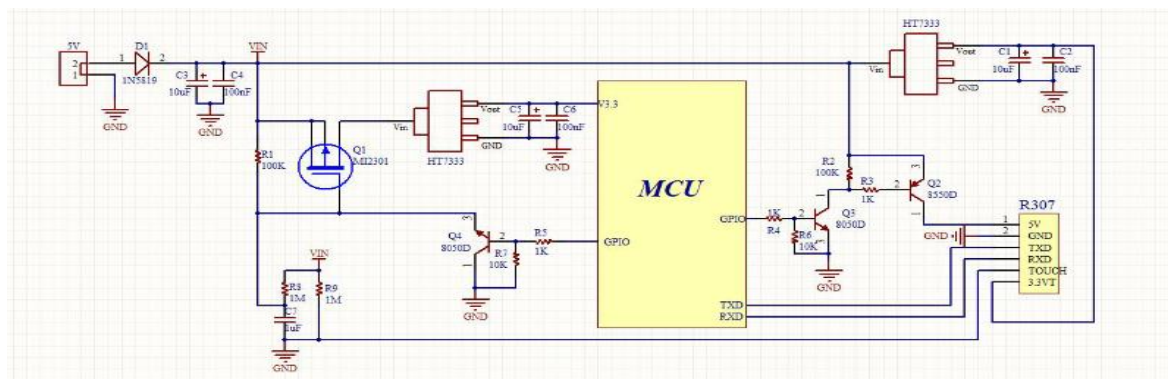
The sensor can be operated at both 5V as well as 3.3V DC. If the sensor is to be interfaced with a 3V3 controller, the 3.3V jumper must be short. If it is to be interfaced with a 5V controller, the jumper must be left open.

The scanner can communicate with a computer/controller using both TTL UART and a USB interface. When connected via a USB port, a virtual COM port is created. It should be noted that pin 6 is the supply voltage for finger detection. If pin 6 is connected to a 3.3V supply, the output of pin 5 goes HIGH when a live finger is placed over the sensor. It is helpful for manual scanning of the finger; otherwise, the sensor starts looking for a scan after few

seconds from power-up. It is important to select proper supply voltage on the fingerprint scanner. A higher voltage can damage a 3V3 controller or its GPIO pins.

R307 sensor circuit

The controller chip on the R307 fingerprint scanner is AS606 from Synochip. AS606 is a microcontroller capable of digital signal processing. For touch detection, the sensor has TTP233D IC from Tontek. The outer PCB has the following circuit diagram



R307 registers

The R307 scanner has built-in flash memory, and it has many registers and buffer memory to store configuration and fingerprint data. Some of the important R307 registers are explained below.

Notepad: This is a 512 bytes non-volatile flash memory arranged in 16 pages of 32 bytes each. The entire memory is written or updated at once.

Image Buffer: It is a RAM used to store a digital image of the fingerprint impression temporarily. It stores a BMP image of dimensions 256 X 288, where each pixel is stored as a byte.

Character File Buffer: It is used to store a processed high-contrast image of the fingerprint. There are two character file buffers of 512 bytes each, and these store two-character files from two consecutive scans. The two scans are combined to form a template file representing the final version of a fingerprint impression. The template files are stored in the fingerprint library.

Fingerprint Library: It is built-in flash memory where 1000 fingerprint templates can be stored. The template files are stored sequentially in the library.

System Configuration Registers: It is a 16-byte register bank that stores configuration data and status flags. The register bank starts a 2-byte status register, followed by a 2-byte system identifier code, 2-byte library size, 2-byte security level, 4-byte device address, 2-byte data packet size, and 2-byte baud multiplier. The status register is defined as follows.

Bit Num	15	4	3	2	1	0
Description	Reserved		ImgBufStat	PWD	Pass	Busy

Where, Busy = 1 if system is executing command else Busy = 0 if system is free. Pass = 1 if a matching fingerprint is found else Pass = 0 if fingerprint is not found. PWD = 1 if handshaking password is verified else PWD = 0 if password is not matched. ImgBufStat = 1 if image buffer contains a valid image else ImgBufStat = 0 if image is not processed.

System Identifier Code has a fixed value that identifies the module in the R30X series. R307 has a code of 0x0009. Library size indicates the number of fingerprint templates the module can store. For R307, it is 1000. Security value determines the threshold of fingerprint matching. It can be 1 to 5, where 5 is the highest security level providing minimum FAR and maximum FRR. FAR is the likelihood of identifying a weakly matched fingerprint as positive. FRR (False Recognition Rate) is the likelihood of identifying a wrong fingerprint as negative. At level 5, FAR is highest, and FRR is lowest. This is the strictest level of fingerprint matching. The device address is by default 0xFFFFFFFF. It can be modified with the SetAddr command. Data Packet Size determines the maximum size of data sent in a single packet. Its value can be 0~3, where 0 = 32 bytes, 1 = 64 bytes, 2 = 128 bytes and 3 = 256 bytes. Baud Multiplier sets the data communication speed with a computer system. It can be 1~12 in multiples of 9600 bps with a minimum baud rate of 9600 bps and a maximum of 115200 bps.

R307 communication protocol

The scanner can communicate data with a computer system using a UART or USB interface. Both interfaces use a common communication protocol. The data is communicated in the form of packets. Each packet is broken into 10-bit frames. A frame starts with a start bit 0 followed by a byte and ending with an end bit 1. A packet is divided into the following frames.

The Header is 2-byte long, having a fixed value of 0xEF01. The high byte is always sent first. The Address is the 32-bit device address of the scanner. The module accepts a command or data only if the address is correct. The default device address is 0xFFFFFFFF. The Packet Identifier determines the type of packet. It is 0x01 for command, 0x02 for data, 0x07 for acknowledgement packet, 0x08 for indicating end of data transfer packet. A command

packet must follow a data packet. The acknowledgment packet is sent from the module to the computer system. Packet Length indicates the size of the packet content, including a checksum byte. Packet Content can be a command, data, or parameter of varying length as indicated by Packet Length. Checksum is the arithmetic sum of all bytes in Packet Identifier, Packet Length, and Packet Content.

The R307 supports the following instruction set.

type	num	code	description	Type	num	Code	description
System-related	1	13H	To verify password	Fingerprint processing	13	08H	to upload template
	2	12H	To set password		14	09H	To download template
	3	15H	To set device address		15	06H	To store template;
	4	0EH	To set system Parameter		16	07H	to read/load template
	5	17H	Port control		17	0CH	to delete tempates
	6	0FH	To read system Parameter		18	0DH	to empty the library
	7	1DH	To read finger template numbers		19	03H	Carry out precise matching of two templates;
Fingerprint processing	8	01H	Collect finger image	others	20	04H	Search the finger library
	9	0AH	To upload image				
	10	0BH	To download image		21	14H	to get random code
	11	02H	To generate character file from image		22	18H	to write note pad
	12	05H	To combine character files and generate template		23	19H	To read note pad

Character File Buffer: It is used to store a processed high-contrast image of the fingerprint. There are two character file buffers of 512 bytes each, and these store two-character files from two consecutive scans. The two scans are combined to form a template file representing the final version of a fingerprint impression. The template files are stored in the fingerprint library. dges to grab and hold things, and the pattern of these ridges and valleys is also present on the fingertips. A miracle of nature is that this pattern of ridges and valleys is unique for every individual. An impression of our fingerprints is left whenever we grab or hold something due to oil, moisture, dust, and dead cells over the skin. These fingerprints objects are called latent fingerprints.

Fingerprint Library: It is built-in flash memory where 1000 fingerprint templates can be stored. The template files are stored sequentially in the library.

System Configuration Registers: It is a 16-byte register bank that stores configuration data and status flags. The register bank starts a 2-byte status register, followed by a 2-byte system identifier code, 2-byte library size, 2-byte security level, 4-byte device address, 2-byte data packet size, and 2-byte baud multiplier. The status register is defined as follows.

Bit Num	15	4	3	2	1	0
Description	Reserved		ImgBufStat	PWD	Pass	Busy

Where, Busy = 1 if system is executing command else Busy = 0 if system is free. Pass = 1 if a matching fingerprint is found else Pass = 0 if fingerprint is not found. PWD = 1 if handshaking password is verified else PWD = 0 if password is not matched. ImgBufStat = 1 if image buffer contains a valid image else ImgBufStat = 0 if image is not processed.

System Identifier Code has a fixed value that identifies the module in the R30X series. R307 has a code of 0x0009. Library size indicates the number of fingerprint templates the module can store. For R307, it is 1000. Security value determines the threshold of fingerprint matching. It can be 1 to 5, where 5 is the highest security level providing minimum FAR and maximum FRR. FAR is the likelihood of identifying a weakly matched fingerprint as positive. FRR (False Recognition Rate) is the likelihood of identifying a wrong fingerprint as negative. At level 5, FAR is highest, and FRR is lowest. This is the strictest level of fingerprint matching. The device address is by default 0xFFFFFFFF. It can be modified with the SetAddr command. Data Packet Size determines the maximum size of data sent in a single packet. Its value can be 0~3, where 0 = 32 bytes, 1 = 64 bytes, 2 = 128 bytes and 3 = 256 bytes. Baud Multiplier sets the data communication speed with a computer system. It can be 1~12 in multiples of 9600 bps with a minimum baud rate of 9600 bps and a maximum of 115200 bps.

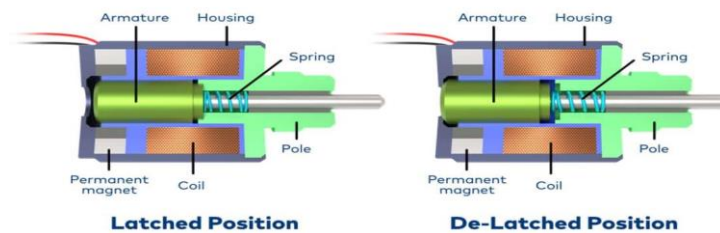
Solenoid Door lock



It is a type of door lock that is based on electromagnetism. It has two components, one that produces a magnetic field(a coil) and another is the moveable armature. When you apply a voltage to the solenoid coil, the armature moves outside the coil and when you do not apply any voltage to the solenoid coil, the armature comes back inside. The solenoid

door lock that we gonna use here, works on 12 volts, and Arduino UNO is not capable of providing 12 volts. So, in order to give power to the door lock, we will use a 5-volt relay module. We will connect the inputs of the relay module to the Arduino UNO and OUTPUT of the relay module to the solenoid door lock. Also, we have to connect a 12-volt power supply to the relay module in order to give power to the lock.

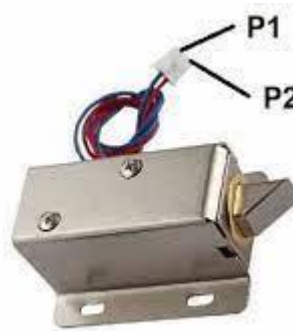
Working of the Solenoid Door Lock



Pin configuration of the Solenoid Door Lock

Features of the door lock

- ☐ The operating voltage is between 9-12 volts DC.
- ☐ The operating current is 900mA at 12v and 750mA at 9v.
- ☐ It has a long activation time of 1-12s.
- ☐ It is rustproof, durable, safe, convenient to use.



P1 - Negative
P2 - Positive solenoid

voltage is

current is

Components needed:

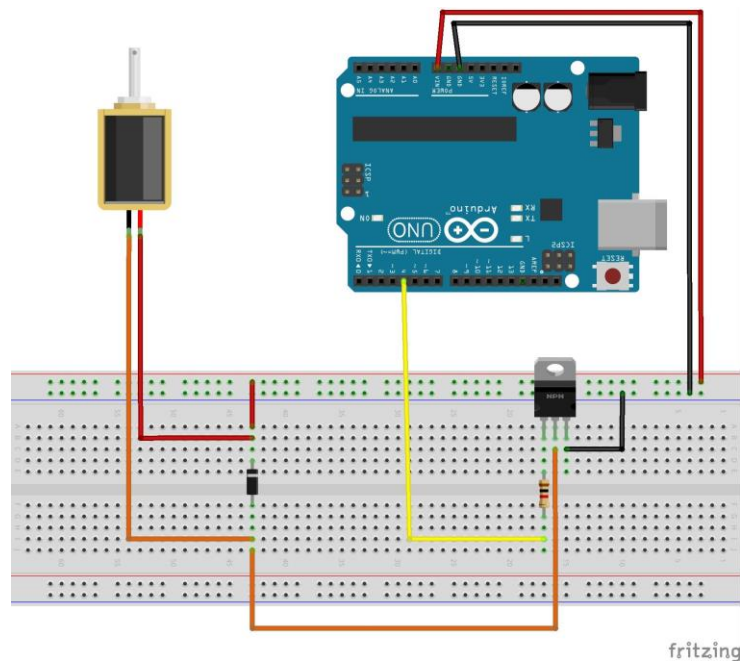
- Arduino UNO
- Tip Transistor
- 12-volt adapter or battery
- Two pushbuttons
- Wires

Pin connections of the solenoid door lock with Arduino UNO

- Connect the VCC pin of the relay module to the 5v pin of the Arduino UNO.
- Connect the GND pin of the relay module to the GND pin of the Arduino UNO.
- Connect the positive wire of the lock to the positive wire of the battery.
- Connect the negative wire of the lock to the NO terminal of the tip transistor.

- Connect the negative wire of the battery to the COM pin of the tip transistor.

Circuit diagram of the Solenoid Door Lock



12V 2AMP DC ADAPTER:



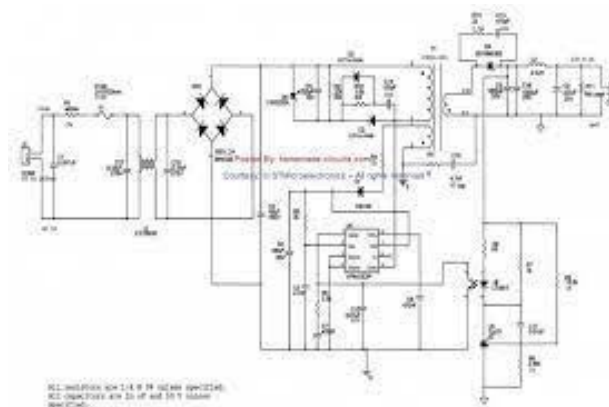
Unlike AC, DC, or "direct current," only flows in one direction. A DC power supply has two wires--one with a negative charge and the other with a positive charge. A device called a rectifier is used to turn AC into DC. The central component of a rectifier is the diode. Diodes are one-way electric valves. When the electricity in the circuit turns negative, a diode lets it flow down the negative wire. When the electricity cycles back to positive, that diode closes

automatically, and another diode lets the positive current flow down the positive wire. There are several different types of rectifiers, but they all use diodes in essentially the same way to separate the negative current from the positive.

SPECIFICATIONS OF ADAPTER:

- AC input voltage: 100V - 240V, 50Hz / 60Hz
- Input current (mA) :100
- Output Voltage (V) :12
- Output current (A) :2
- Load regulation (%) :±5
- Input Plug :2-Pin
- Output Plug :Dual Pin [5.5mm x 2.1mm DC plug] & [4mm x 1.7mm]
- Output DC Voltage: 12V
- Output Current: 2A
- Standby power: <0.3W

CIRCUIT OF THE ADAPTER:



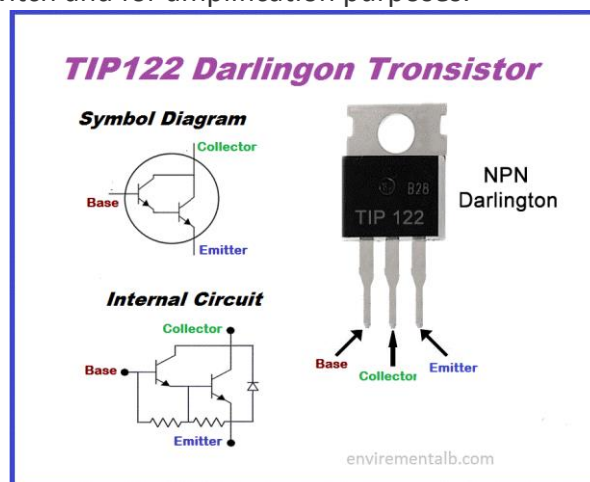
Features:

- ☐ This is a durable product with a 6-month warranty.
- ☐ The adapter is easy to use.
- ☐ It comes with a long wire.
- ☐ Wide input voltage range (100VAC-280VAC).
- ☐ Very low no-load power consumption.
- ☐ Very low ripple & noise output for device safety.

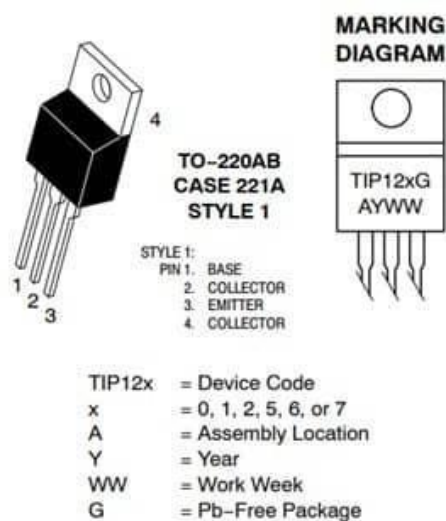
- ☐ Protections against- under-over voltage/ overload/ short circuit with auto-restart on fault removal.
- ☐ Thermal shut down (140°C) with auto-restart on cold condition.
- ☐ Soft start & low inrush current.
- ☐ Isolation up to 3kv for 5sec.time period.
- ☐ High operating ambient temperature up to 60°C.
- ☐ Highly efficient, compact, durable and long life.

TIP 122 TRANSISTOR:

TIP122 is an NPN Darlington transistor. Darlington transistor means there are two transistor in one package connected to increase gain at output. TIP122 transistor has a lot of good features like 5A collector current, max emitter-base voltage is 5V, max collector dissipation is 65 watt, minimum & maximum current gain is equal to 1000. This transistor is designed to use as a switch and for amplification purposes.



TIP 122 PINOUT:



Pin Number	Pin Name	Description
1	Base	It governs the biasing of the transistor and works to turn ON or OFF the transistor.
2	Collector	Current flows in through collector, usually connected to load
3	Emitter	Current comes out by the emitter, it is usually linked to ground.

TIP TRANSISTOR APPLICATIONS:

- ◆ Audio Amplifier
- ◆ Audio Amplifier Stages
- ◆ Audio Preamplifiers
- ◆ Switching Loads Under 5A

WORKING OF TIP TRANSISTOR:

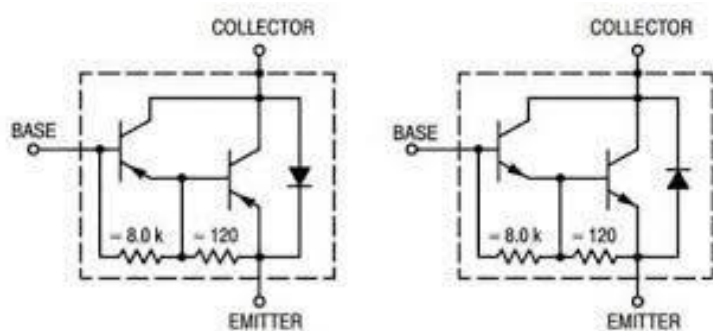
This transistor is recognized for its higher current gain which is 1000 and higher collector current 5 amperes, therefore, it is usually used to switch.

This transistor has less base and emitter Voltage of the merely 5V henceforth can be effortlessly organized by a Logic instrument such as a microcontroller.

Though precaution has to be engaged to check, if the logic instruments can font up to 120mA.

Though TIP122 has extraordinary current at collector and current gain, it is impartially modest to switch the expedient meanwhile it has an Emitter-Base voltage (VBE) of the only 5V and Ib of merely 120mA.

DARLINGTON CIRCUIT SCHEME:



TIP TRANSISTOR IS USED IN:

This transistor is known for its high current gain ($h_{fe} = 1000$) and high collector current ($I_C = 5A$) hence it is normally used to control loads with high current or in applications where high amplification is required. This transistor has a low Base-Emitter Voltage of the only 5V hence can be easily controlled by a Logic device like microcontrollers. Although care has to be taken to check if the logic device can source up to 120mA.

So, if you looking for a transistor that could be easily controlled by a Logic device to switch high power loads or to amplify high current then this Transistor might be an ideal choice for your application.

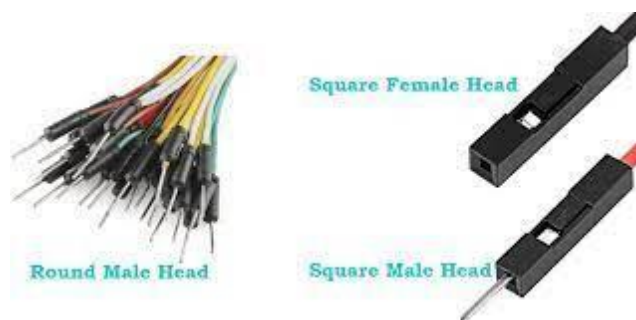
JUMPER WIRES:



Jumper wires are connector pins at to be used to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.

simply wires that have each end, allowing them connect two points to

Types of Jumper Wires



Jumper wires typically come in three versions: **male-to-male**, **male-to-female** and **female-to-female**. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into

Product Specification

Current	4-20 mA
Voltage	12 V
Rated Pressure	25 kPA
Pitch	2.54 mm
Cable Length	20 cm - 8 Inch
Weight	30 gm
Accuracy	0.25%F.S.

ARDUINO IDE:




The **Arduino Integrated Development Environment (IDE)** is a cross-

platform application (for [Windows](#), [macOS](#), [Linux](#)) that is written in functions from [C](#) and [C++](#).^[3] It is used to write and upload programs to [Arduino](#) compatible boards, but also, with the help of third-party cores, other vendor development boards.^[4] The source code for the IDE is released under the [GNU General Public License](#), version 2.^[5] The Arduino IDE supports the languages [C](#) and [C++](#) using special rules of code structuring.^[6] The Arduino IDE supplies a [software library](#) from the [Wiring](#) 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](#) program with the [GNU toolchain](#), also included with the IDE distribution.^[7] 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.[8] By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards.[9]

Arduino IDE is a derivative of the [Processing IDE](#),[10] however as of version 2.0, the Processing IDE will be replaced with the [Visual Studio Code](#)-based [Eclipse Theia](#) IDE framework.[2]

Arduino Pro IDE

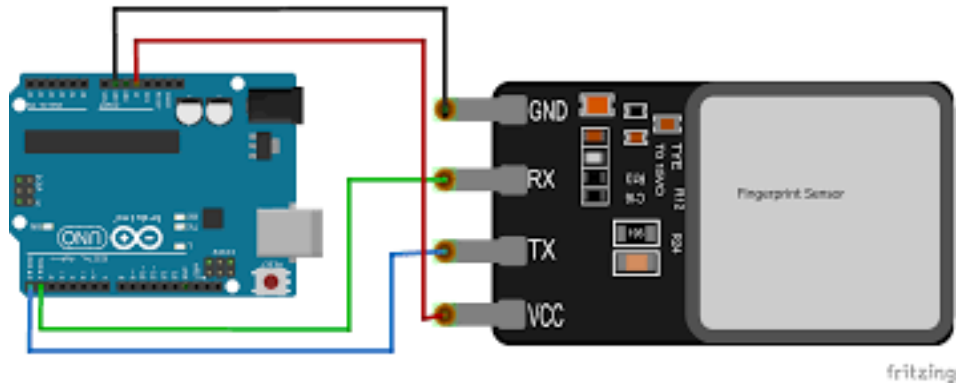
Developer(s)	Arduino Software
Preview release	v0.1.2 / 14 September 2020; 14 months ago[11]
Repository	github 
Written in	C, C++
Operating system	Windows, macOS, Linux
Platform	IA-32, x86-64, ARM
Type	Integrated development environment
License	LGPL or GPL license
Website	blog

With the rising popularity of Arduino as a software platform, other vendors started to implement custom open source compilers and tools (cores) that can build and upload sketches to other [microcontrollers](#) that are not supported by Arduino's official line of microcontrollers.

In October 2019 the [Arduino](#) organization began providing early access to a new Arduino Pro IDE with debugging[12] and other advanced features.[13]

STEPS TO EXECUTE OUR PROJECT:

step 1 :connect the fingerprint sensor with the aurdino board as given below connections



Fingerprint Sensor Connection

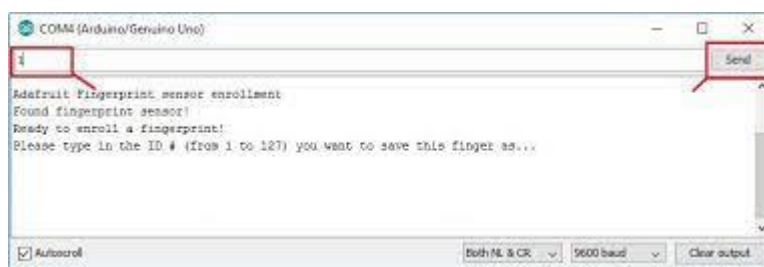
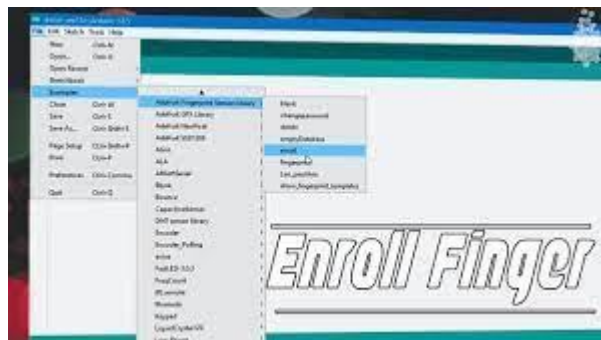
Black Wire ► Arduino GND

Red Wire ► Arduino 5V

Green Wire ► Digital Pin 2

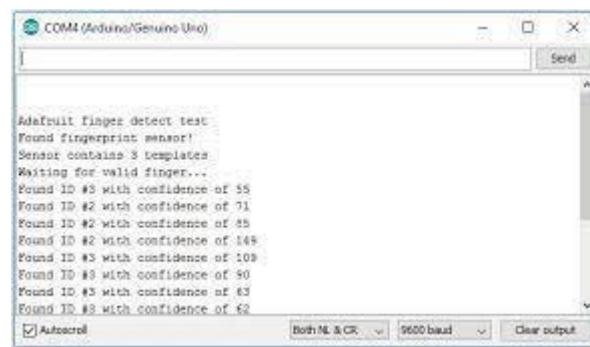
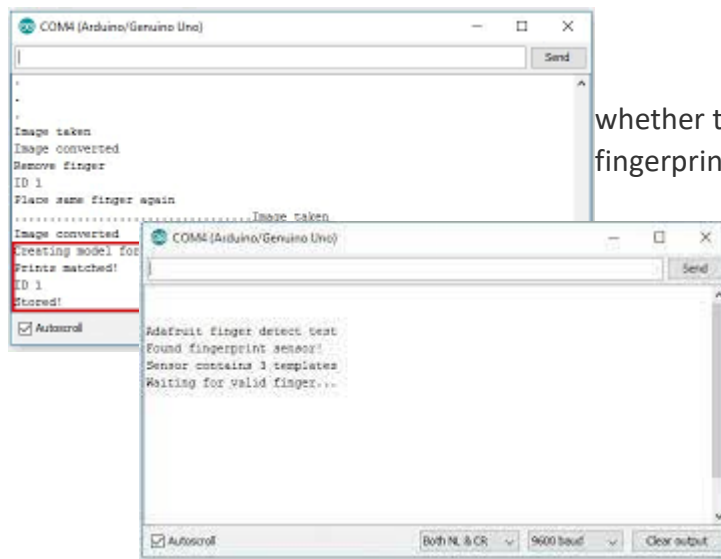
White Wire ► Digital Pin 3

step 2 open the aurdino ide include adafruit fingerprint libraries then include the enroll code



step 3 :chek
not by using

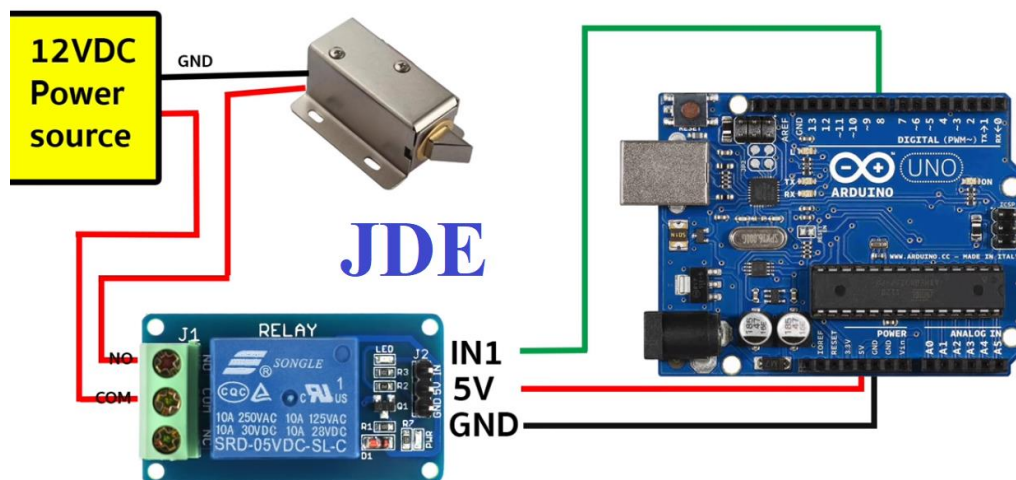
whether the fingerprint saved or
fingerprint template code



verify whether given fingerprint is saved or not if not enroll once again the enroll code and save the deired fingerprintsyou can save upto 127 fingerprints in this module

step 4:connect the solenoid lock to aurdino as given below circuit and block diagram

Arduino UNO + Relay Module + Solenoid door lock



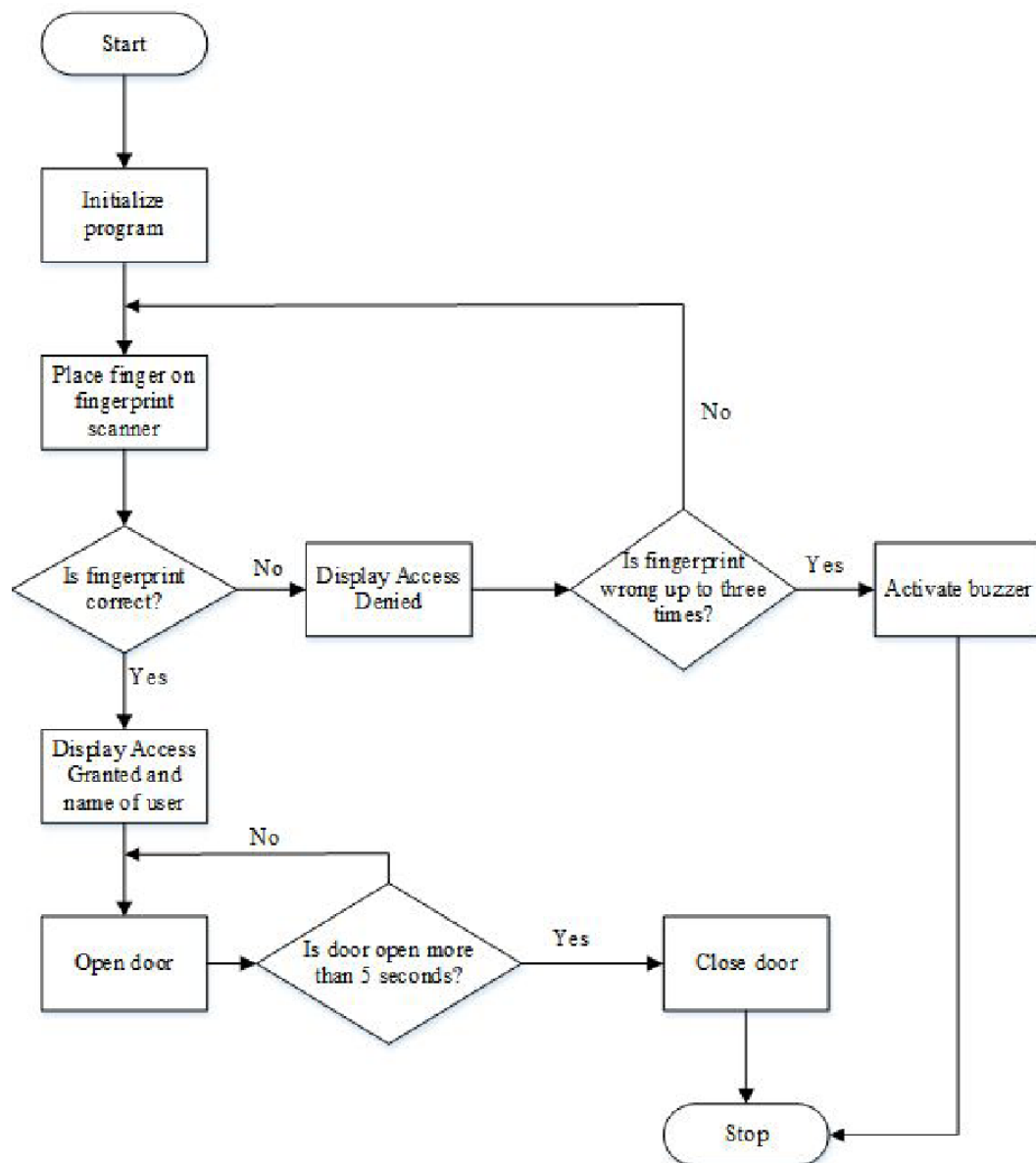
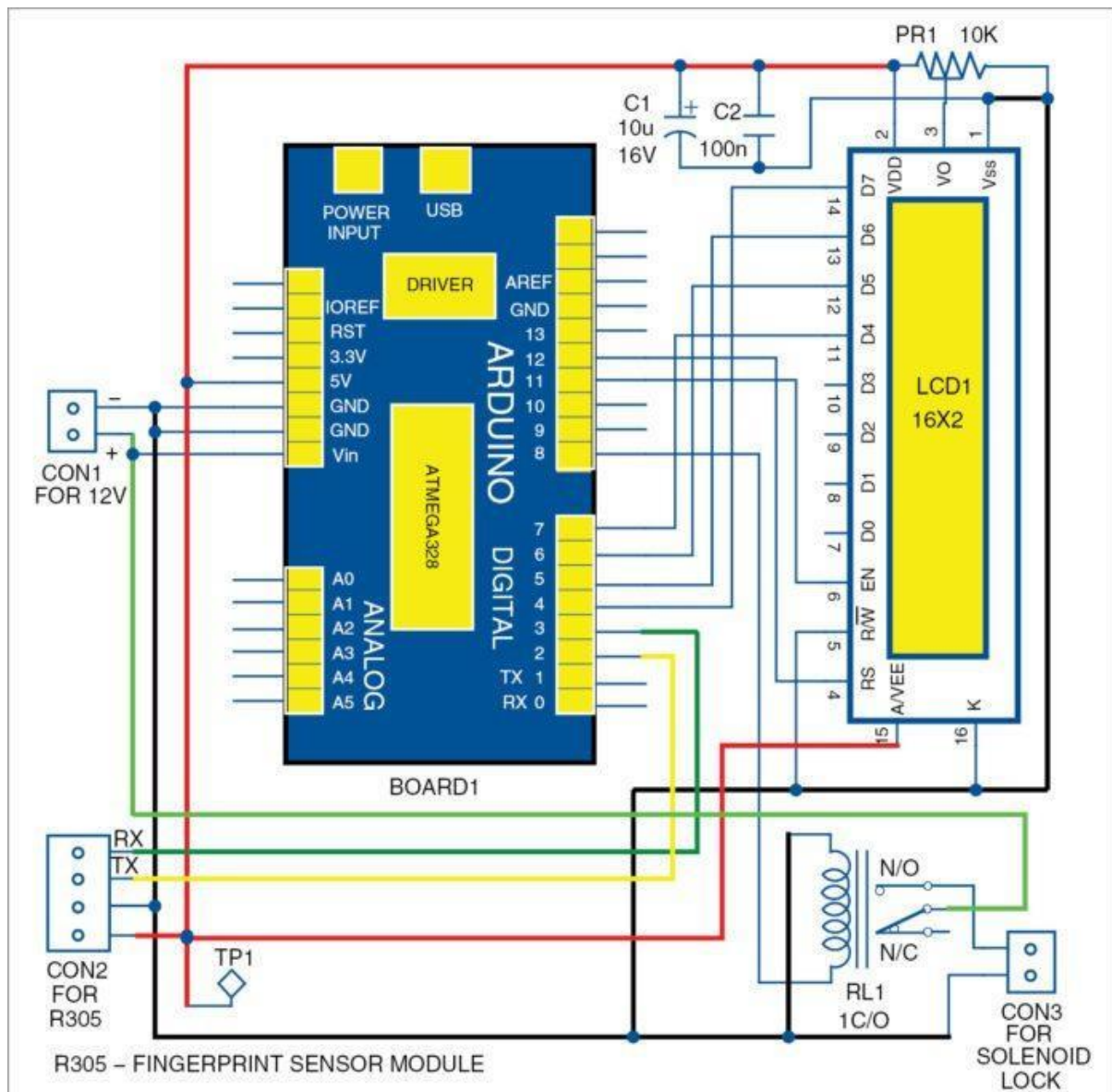


Figure 5: Flowchart of the biometric door lock authentication process

step 6: now connect the 12v dc adaptor to the aurdino board before connecting remove the cable and connection between aurdino and systemor pc or else laptop



step 7: our project is finished now check whether the project is working or not.

CODE FOR FINGERPRINT ENROLLMENT:

Enrollment code

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); //you can change them if it is not working on 2 or 3

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

uint8_t id;

void setup()
{
  Serial.begin(9600);
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nFingerprint sensor enrollment");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }
}

uint8_t readnumber(void) {
  uint8_t num = 0;

  while (num == 0) {
    while (! Serial.available());
    num = Serial.parseInt();
  }
  return num;
}

void loop()          // program wil repeat this part (loop here)
{
```

```

Serial.println("Ready to enroll a fingerprint!");
Serial.println("Please type in the ID # (from 1 to 127) you want to save this finger as...");
id = readnumber();
if (id == 0) { // ID #0 not allowed, try again!
    return;
}
Serial.print("Enrolling ID #");
Serial.println(id);

while (! getFingerprintEnroll() );
}

uint8_t getFingerprintEnroll() {

    int p = -1;
    Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println(".");
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Communication error");
                break;
            case FINGERPRINT_IMAGEFAIL:
                Serial.println("Imaging error");
                break;
            default:
                Serial.println("Unknown error");
                break;
        }
    }
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        break;

```

```

case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

```

```

Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.print(".");
            break;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Communication error");
            break;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Imaging error");
            break;
        default:
            Serial.println("Unknown error");
            break;
    }
}

```

```

    }
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

// OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

```

```

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
  Serial.println("Stored!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
  Serial.println("Communication error");
  return p;
} else if (p == FINGERPRINT_BADLOCATION) {
  Serial.println("Could not store in that location");
  return p;
} else if (p == FINGERPRINT_FLASHERR) {
  Serial.println("Error writing to flash");
  return p;
} else {
  Serial.println("Unknown error");
  return p;
}
}

```

FINGERPRINT TEMPLATE PROGRAME:

/*****

This is an example sketch for our optical Fingerprint sensor

Adafruit invests time and resources providing this open source code,
please support Adafruit and open-source hardware by purchasing
products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.

BSD license, all text above must be included in any redistribution

*****/

```
#include <Adafruit_Fingerprint.h>
```

```
#if (defined(__AVR__) || defined(ESP8266)) && !defined(__AVR_ATmega2560__)
```

```
// For UNO and others without hardware serial, we must use software serial...
```

```
// pin #2 is IN from sensor (GREEN wire)
```

```
// pin #3 is OUT from arduino (WHITE wire)
```

```
// Set up the serial port to use software serial..
```

```
SoftwareSerial mySerial(2, 3);
```

```
#else
```



```
// On Leonardo/M0/etc, others with hardware serial, use hardware serial!  
// #0 is green wire, #1 is white  
#define mySerial Serial1  
  
#endif
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);  
int getFingerprintIDez();
```

```
void setup()  
{  
  while (!Serial);  
  Serial.begin(9600);  
  Serial.println("Fingerprint template extractor");
```

```
  // set the data rate for the sensor serial port  
  finger.begin(57600);
```

```
  if (finger.verifyPassword()) {  
    Serial.println("Found fingerprint sensor!");  
  } else {  
    Serial.println("Did not find fingerprint sensor :(");  
    while (1);  
  }
```

```
  // Try to get the templates for fingers 1 through 10  
  for (int finger = 1; finger < 10; finger++) {  
    downloadFingerprintTemplate(finger);  
  }  
}
```

```
uint8_t downloadFingerprintTemplate(uint16_t id)  
{  
  Serial.println("-----");  
  Serial.print("Attempting to load #"); Serial.println(id);  
  uint8_t p = finger.loadModel(id);  
  switch (p) {  
    case FINGERPRINT_OK:  
      Serial.print("Template "); Serial.print(id); Serial.println(" loaded");  
      break;  
    case FINGERPRINT_PACKETRECEIVEERR:  
      Serial.println("Communication error");  
      return p;
```

```

default:
    Serial.print("Unknown error "); Serial.println(p);
    return p;
}

// OK success!

Serial.print("Attempting to get #"); Serial.println(id);
p = finger.getModel();
switch (p) {
    case FINGERPRINT_OK:
        Serial.print("Template "); Serial.print(id); Serial.println(" transferring:");
        break;
    default:
        Serial.print("Unknown error "); Serial.println(p);
        return p;
}

// one data packet is 267 bytes. in one data packet, 11 bytes are 'usesless' :D
uint8_t bytesReceived[534]; // 2 data packets
memset(bytesReceived, 0xff, 534);

uint32_t starttime = millis();
int i = 0;
while (i < 534 && (millis() - starttime) < 20000) {
    if (mySerial.available()) {
        bytesReceived[i++] = mySerial.read();
    }
}
Serial.print(i); Serial.println(" bytes read.");
Serial.println("Decoding packet...");

uint8_t fingerTemplate[512]; // the real template
memset(fingerTemplate, 0xff, 512);

// filtering only the data packets
int uidx = 9, index = 0;
memcpy(fingerTemplate + index, bytesReceived + uidx, 256); // first 256 bytes
uidx += 256; // skip data
uidx += 2; // skip checksum
uidx += 9; // skip next header
index += 256; // advance pointer
memcpy(fingerTemplate + index, bytesReceived + uidx, 256); // second 256 bytes

```

```

for (int i = 0; i < 512; ++i) {
    //Serial.print("0x");
    printHex(fingerTemplate[i], 2);
    //Serial.print(", ");
}
Serial.println("\ndone.");

return p;

/*
uint8_t templateBuffer[256];
memset(templateBuffer, 0xff, 256); //zero out template buffer
int index=0;
uint32_t starttime = millis();
while ((index < 256) && (millis() - starttime) < 1000))
{
    if (mySerial.available())
    {
        templateBuffer[index] = mySerial.read();
        index++;
    }
}

Serial.print(index); Serial.println(" bytes read");

//dump entire templateBuffer. This prints out 16 lines of 16 bytes
for (int count= 0; count < 16; count++)
{
    for (int i = 0; i < 16; i++)
    {
        Serial.print("0x");
        Serial.print(templateBuffer[count*16+i], HEX);
        Serial.print(", ");
    }
    Serial.println();
}*/
}

void printHex(int num, int precision) {
    char tmp[16];
    char format[128];

    sprintf(format, "%%.%dX", precision);

    sprintf(tmp, format, num);

```

```
Serial.print(tmp);  
}
```

```
void loop()  
{}
```

finger print sensor programme:

```
#include <Adafruit_Fingerprint.h>  
#include <SoftwareSerial.h>
```

```
SoftwareSerial mySerial(2, 3);
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
```

```
void setup()  
{  
  Serial.begin(9600);  
  while (!Serial); // For Yun/Leo/Micro/Zero/...  
  delay(100);  
  Serial.println("fingertest");  
  pinMode(12, OUTPUT);  
  pinMode(11, OUTPUT);
```

```
  // set the data rate for the sensor serial port  
  finger.begin(57600);
```

```
  if (finger.verifyPassword()) {  
    Serial.println("Found fingerprint sensor!");  
  } else {  
    Serial.println("Did not find fingerprint sensor :(");  
    while (1) {  
      delay(1);  
    }  
  }  
}
```

```
  finger.getTemplateCount();  
  Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println("  
templates");  
  Serial.println("Waiting for valid finger...");  
}
```

```
void loop()          // run over and over again
```

```

{
  getFingerprintIDez();
  delay(50);      //don't ned to run this at full speed.
  digitalWrite(12, LOW);
  digitalWrite(11, LOW);
}

```

```

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }
}

```

// OK success!

```

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:

```

```

    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}
{digitalWrite(11, HIGH);
delay(3000);
digitalWrite(11, LOW);
Serial.print("Not Found");
Serial.print("Error");
return finger.fingerID;
}

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);

return finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;

```

```
p = finger.fingerFastSearch();  
if (p != FINGERPRINT_OK) return -1;
```

```
// found a match!
```

```
{  
  digitalWrite(12, HIGH);  
  delay(3000);  
  digitalWrite(12, LOW);  
  Serial.print("Found ID #"); Serial.print(finger.fingerID);  
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
```

```
} }
```

Advantages and Disadvantages of our project:

Advantages:

1).Security

The traditional lock and key based system have plenty of vulnerabilities. The major one among them is the fact that the key can be easily duplicated. This would result in unpleasant surprises at any point in time. The consequences are even more severe when an establishment is being used as a place for business.

The legal ramifications of losing your client's sensitive data and the consequent breach of privacy is huge. With fingerprints being unique, it is next to impossible for someone to break into your premises with use of a biometric lock.

2).No problem of lost keys

If you have got the habit of losing keys, then it can be annoying to wait. So the only way to keep your home and business life easy is to use the fingerprint locks..It will require a total change of the locks for installing the best fingerprint locks.

3).Hard to override

Unlike what happens with door locks operated with keys, a finger print lock cannot be opened by any other person. This is unless the person can use your fingers to access it. Such a situation is almost impossible because it might be hard for anyone to fake your fingerprints. Conventional locks are very easy to open especially for people who are experts in lock picking. This means that when you have the finger print lock, your business will not be prone to robberies as it offers high level defense against any kind of intrusion.

4).User-friendly

It is very simple to use a finger print lock compared to the conventional locks. They are not complicated even the way they are designed, so it does not require complex ways to operate it too. All what you need to do is to place your finger on the lock and it will open and close automatically.

5).You do not need a key

One of the major advantages of this lock is that you do not require a key to operate your lock system. This is more advantageous if you are the kind of a person who keeps on losing or misplacing your business keys. With finger print lock, a huge burden will be eliminated from your shoulder because keeping and maintaining your key can be a major thing for many people. The other advantage is that it is impossible to misplace your fingerprints, so no inconvenience is caused at any given time.

6).Cost effectiveness

Though you would have to pay more upfront for the fingerprint-based door lock, it is bound to last longer than your regular lock and key, which is susceptible to rust, wear, and tear. Fingerprint door lock systems are also perfect for business establishments where they are

used frequently. Hence the recurrent cost of replacement can be done away with by installing the former. This would ensure you save your hard-earned money in the long run.

7).No codes to remember

Another great advantage is that there are no codes involved in operating this type of lock. This means that you do not have to keep on remembering codes as you operate the lock. It is not possible to forget your fingerprints as it happens with codes, making this type of lock more efficient and reliable. This saves you valuable time, which translates to more money in your business. In case one user is no longer permitted to use the lock, their fingerprint is just erased from the registry and they cannot operate the lock again. This also boosts the security of your business because no one can fake the fingerprints as it happens with lock codes.

8).Give peace of mind

There is nothing good as having peace of mind knowing that no one can access your business even when you are far away. With finger print lock, you have this peace of mind because it is only authorized users who can operate the lock. It is hard to break the lock as it happens with conventional locks during robbery.

9).NO RISK OF MISPLACING

The thing with keys is that they tend to fall out of hands/bags/pockets and disappear for eternity. Once that happens, the entire hassle of changing locks and making a fresh set of keys ensues. It can be reasonably argued that you cannot lose your fingers (thus your fingerprint). Hence once again, such a system would come in handy.

10).AESTHETICS

The presence of a fingerprint-based door lock adds to the look and charm of the premise. It immediately commands awe and sends out a warning signal to those who might think about breaking in.