DEPARTMENTAL VISION & MISSION

ELECTRONICS & TELECOMMUNICATION ENGINEERING DEPARTMENT

VISION

"Electronics & Telecommunication Department will strive to empower students with excellent technical knowledge and practical skills to meet the challenges of the industry and the society."

MISSION

- **To provide quality technical education through continuous faculty training and industry institute interaction.**
- **To develop and create technical ambience in the department with latest infrastructure facilities.**

Project report
On

“SMART LAB SECURITY SYSTEM”

Submitted by

AADITYA AUTI
DEVEN SATAM
MANISH KORI
DURGESH YADAV

Project Guide
MR. SHARAD YENKAR

DEPARTMENT OF
ELECTRONICS & TELECOMMUNICATION
ENGINEERING

ZAGDU SINGH CHARITABLE TRUST (REGD.)

THAKUR POLYTECHNIC

(An ISO 9001:2008 Certified Institute)

Thakur Complex, Next to W.E. Highway, Kandivli (E), Mumbai – 400 101.

2018-19

Zagdu Singh Charitable Trust's (Regd.)
THAKUR POLYTECHNIC
(Approved by AICTE, Recognised by Government of Maharashtra & Affiliated to MSBTE)
(Accredited by : National Board of Accreditation, New Delhi*, ISO 9001:2008 Certified Institute)

PROJECT APPROVAL SHEET

This Project work entitled

SMART LAB SECURITY SYSTEM

By

AADITYA AUTI
DEVEN SATAM
MANISH KORI
DURGESH YADAV

Is approved for the award of the

DIPLOMA
IN

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

MR. SHARAD YENKAR

Project Guide

EXTERNAL EXAMINER

INTERNAL EXAMINER

Zagdu Singh Charitable Trust's (Regd.)
THAKUR POLYTECHNIC
(Approved by AICTE, Recognised by Government of Maharashtra & Affiliated to MSBTE)
(Accredited by : National Board of Accreditation, New Delhi*, ISO 9001:2008 Certified Institute)

Affiliated to Maharashtra State Board of Technical Education (MSBTE) Mumbai

C E R T I F I C A T E

This is to certify that

AADITYA AUTI
DEVEN SATAM
MANISH KORI
DURGESH YADAV

Have submitted this project work entitled **"SMART LAB SECURITY SYSTEM"** is a bonafied record of project work carried out in partial fulfillment for the award of the

DIPLOMA
IN

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

This is a record of their own work carried out by them under our supervision and guidance

---
PROJECT GUIDE
(MR. SHARAD YENKAR)

H.O.D. (DEPT)
(MR. GANESH AKOLIYA)

PRINCIPAL
(Dr. S.M. GANECHARI)

2018-2019

ACKNOWLEDGEMENT

We are thankful to several individuals who have contributed towards our final year project and without whose help it would not have been possible. We would like to express our deep sense of gratitude to our **Principal**, **Dr. S.M. Ganechari** for his valuable guidance in this endeavor. We are also highly grateful to our **Head of Department**, **Mr. Ganesh Akoliya** for his thought-provoking comments, valuable suggestions, constant motivation, encouragement and support.

We offer our sincere thanks to our **Project guide**, **Mr. Sharad Yenkar** for his constant support and timely help and guidance throughout our preparation. His involvement right from conceptualization of the study through its execution and meticulous perusal of the manuscript ensured its timely completion.

We wish to take this opportunity to render thanks towards all the **Electronics and Telecommunication Engineering faculty** at **Thakur Polytechnic** for their help and encouragement towards making this project successful. Heartfelt thanks to our family for their unconditional support and last but not the least our friends and colleagues for their help and co-operation.

Mumbai, February 2017.

Thanking you,

Aaditya Auti
Deven Satam
Manish Kori
Durgesh Yadav

ABSTRACT

Networking technology plays a vital role in human life by virtually penetrating through different technologies. One of the popular domains of networking technology is Wireless Sensor Networks (WSN). Huge numbers of applications are supported by WSN. Applications of WSN include habitat monitoring, wildfire monitoring, navigation and many more. In this paper we concentrate on home automation based on WSN. The home technology is moving rapidly from the programmable thermostat to an era where all home systems are integrated into a centralized control one which can be accessible from multiple entry points like touch pads, telephones, smart phones and tablets. The result is highly personalized home environments which interpret and react to human needs. We extend the technology of home automation to college campus laboratories leading to smart labs.

INDEX

1. INTRODUCTION

Smart lab thus creates wire free systems which cause minimal cost and disruption. The routing mechanism which is implemented in routing the data collected by sensor nodes to the sink is through Labeled Based Multipath Routing (LMR) protocol. In today's world, people rapidly use internet for most of the activities in daily life. The control of smart labs can also be extended to PDA's equipped with internet. This leads to Internet of Things (IOT) where it is all about physical items talking to each other, machine-to-machine communications and person-to-computer communications will be extended to things. This paper illustrates a cost effective and flexible solution for energy management and security related aspects using sensor networks.

Smart labs are implemented with two important aspects:

1) Smart power management for energy cost reduction.

2) Providing security for intrusion from unauthorized people

2. BASIC PRINCIPLE

In many research centre or offices, the laboratory plays very important role. So, the security of the lab is plays important role. Those labs may content the very confidential and important information, so the smart lab security system is used at such a place. It consists of many sensors and module to fulfil the requirement of security. The all modules are embedded in one system.

The very important component of this system is ARDIUNO UNO. It is used as the controller in the security system. The all modules and components relate to the ARDIUNO UNO development board. The PIR sensors used to sense the movement of any moving object to controlling the light in lab or certain room. If the PIR sensor sense any moving object it will give signal to relay through the ARDIUNO UNO. Here the use of relay to isolate the 5v signal to 110v to 230v supply for controlling the light. The IR counter is used to measure the persons entered in the lab. The count is displayed on the 16x2 liquid crystal display. The display is also connected to the ARDIUNO UNO board.

The fingerprint sensor named as R-305 fingerprint sensor is used to take fingers impressions to verifying the finger prints which previously stored in the database of ARDIUNO UNO. If the finger impression is matched with the database impression the signal will be generated and go to the ARDIUNO UNO board. The controller will give the command to the solenoid lock. If the command positive means the impression is matched the lock will open. And, if the command is negative in nature then the lock didn't change its status. Here the solenoid lock is used to lock the doors. This sensor is also used for securing the documents cupboards.

The SIM-800 GSM module is used to transfer the data on your personal cell for 1 or more than 1 users. The requirement of the SIM-800 GSM is only one activated sim card which can communicate with the GSM module. It contents efficient balance to send SMS on phone. The very dedicated supply is used to power up the GSM module because it very sensitive in nature. To send the SMS to the user it drives more than equal to 3.4V & 2A current. If the supply is greater than the 3.4V & 2A the GSM module will get burned. It has on board antenna to easily communicate with outer world. We can also connect the external antenna to extend the range of the GSM module.

The emergency power backup is also designed in this system. It helps to run the system in absence of the electricity. Whenever the supply gets interrupted the backup supply is automatically get triggered.

3. BLOCK DIAGRAM

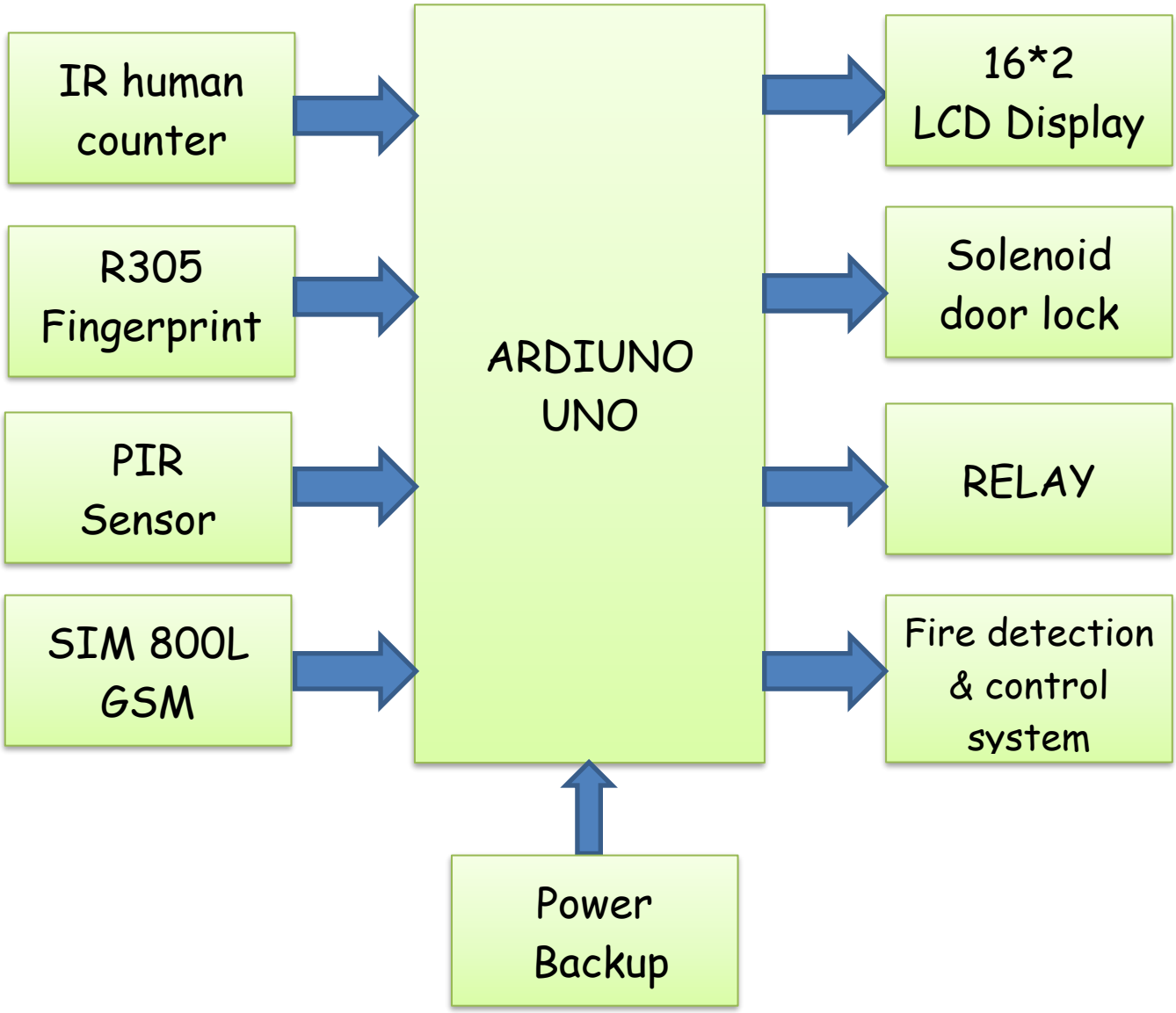


Fig3.1: block diagram of SMART LAB SECURITY SYSTEM

4. CIRCUIT DIAGRAM

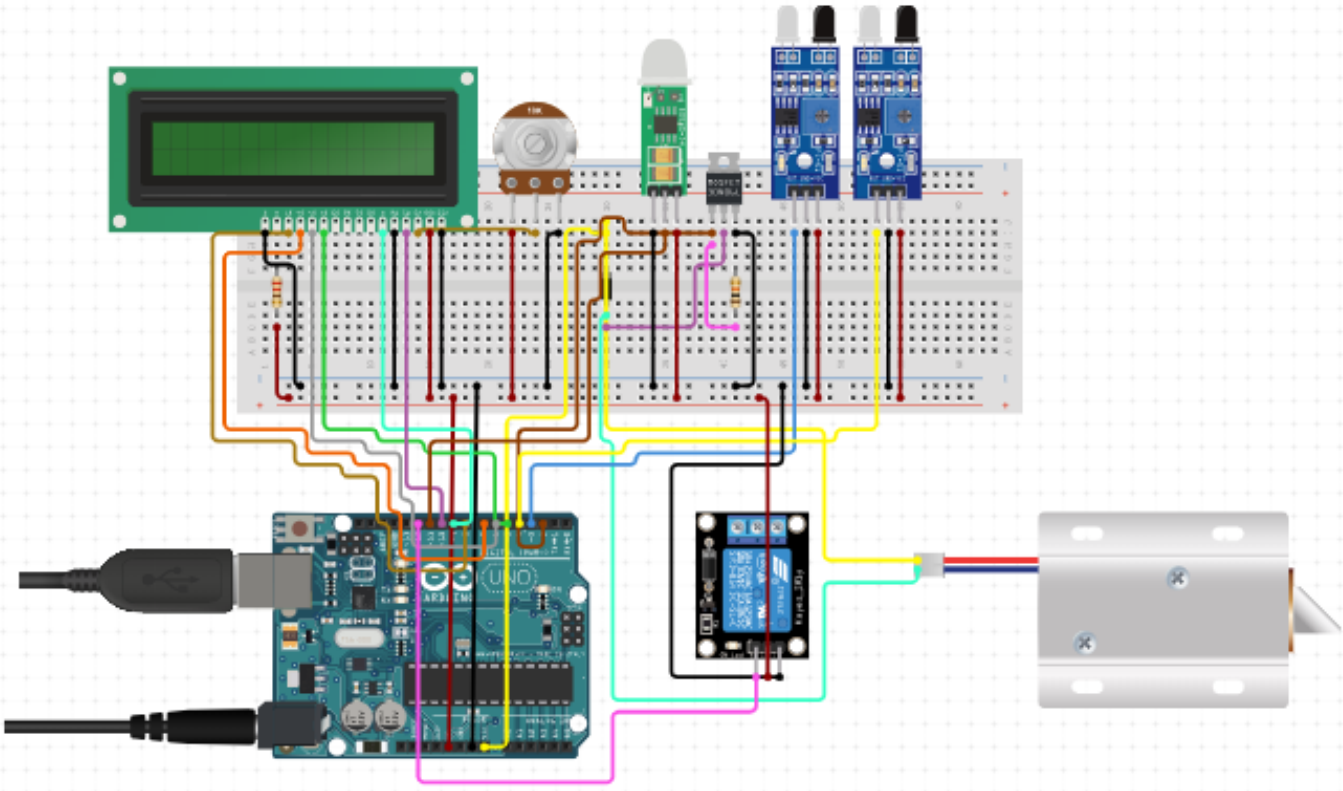


Fig4.1: circuit diagram of smart lab security system

5. EXPLAINATION OF COMPONENTS

Table 5.1: LIST OF COMPONENTS

COMPONENT	VALUE	QUANTITY
Resistors	10kΩ	1
	220Ω	1
	1kΩ	1
Diode	IN4007	1
N-channel MOSFET	30N06L	1
Voltage Regulator	L7805	1
Rotary potentiometer	10kΩ	1
ARDIUNO UNO	UNO R3	1
USB cable	A to B	1
Solenoid lock	12V	1
Display	16x2 LCD	1
Relay module	2 channels	1
Mini PIR	HC-SR505	1
IR sensor	5V	2
Jumper wires	M-M	20
Jumper wires	M-F	20
Wall adaptor	12V/2A	1

6. ARDIUNO UNO

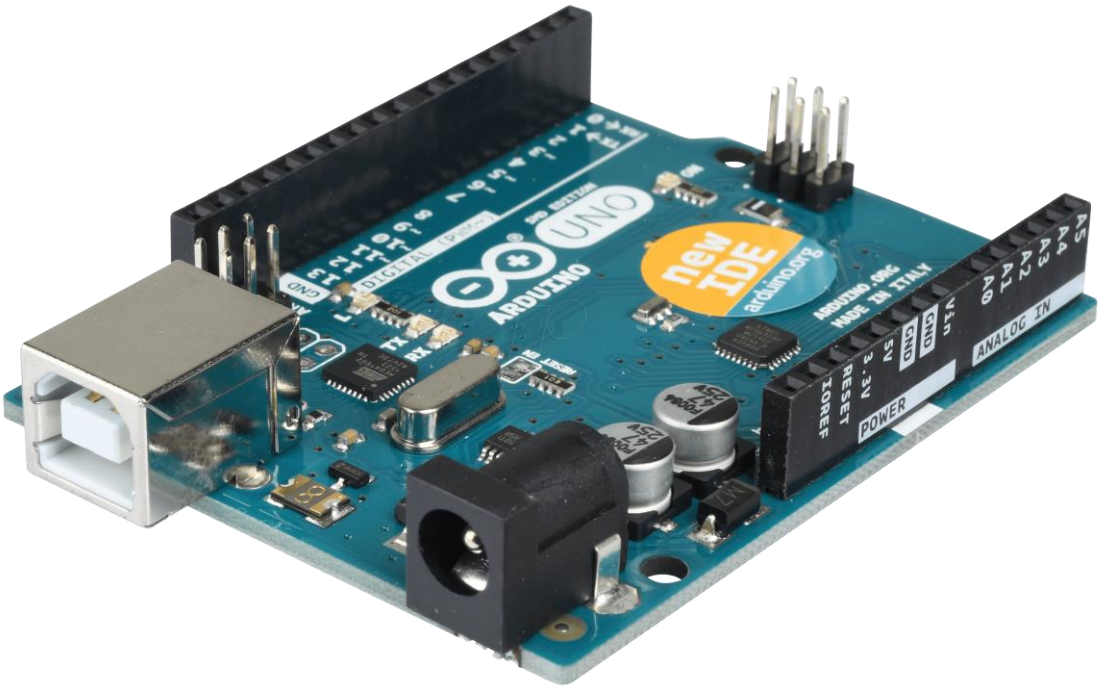


Fig 6.1 : Arduino UNO R3

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and Analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. It is also like the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogramed with a bootloader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

There is alternate function of each pin of ARDIUNO UNO board:

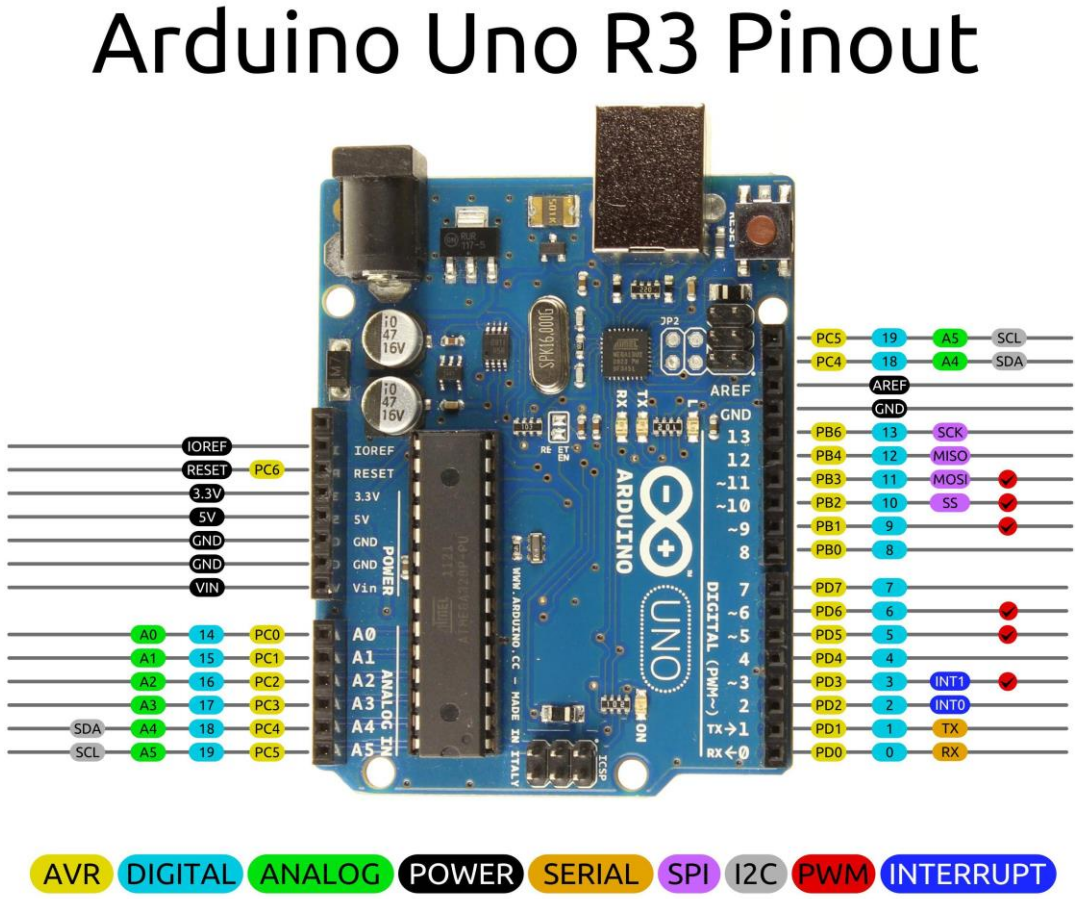


Fig 6.2 : Alternate Functions of ARDIUNO UNO

FEATURES OF ARDIUNO UNO: -

- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader

- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g

GENERAL PIN FUNCTIONS: -

- **LED**: There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **VIN**: The input voltage to the Arduino/ Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator and can damage the board.
- **3V3**: A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**: Ground pins.
- **IOREF**: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- **Reset**: Typically used to add a reset button to shields which block the one on the board.
- **Serial** / UART: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM** (**P**ulse **W**idth **M**odulation): 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite() function.

- **SPI** (**S**erial **P**eripheral **I**nterface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **TWI** (**T**wo **W**ire **I**nterface) / I²C: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- **AREF** (**A**nalog **REF**erence): Reference voltage for the analog inputs.

- **LCD section: -**

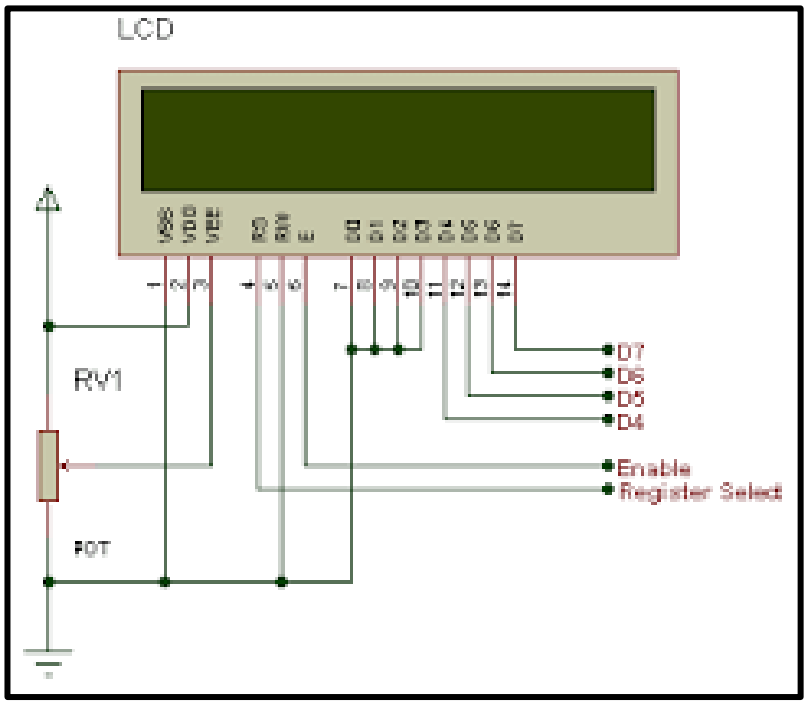


Fig 6.3: pinout of LCD display

The LCD screen is interfaced using 7 lines, out of which there are three control lines, four bidirectional data lines. Two backlight power lines are also there. The three control lines are: -

- RS (Register select): It is used to select either command register or data register.

- RW (Read/Write): It is used to select Read or Write mode of operation. Although LCD is an output device, data or status (busy flag) can be read from the LCD. Busy flag indicates if LCD is ready to accept data or is busy in performing internal operations. The busy flag must be checked through program before sending data or command to LCD, otherwise there is a possibility that LCD may not accept the data or command as it is busy performing previous operations.

- E (Enable): This pin is used to enable transfer of data or command from the output port to the LCD latch.

Table 6.1 : pin functions of LCD display

Interface Pin Function

Pin No.	Symbol	Level	Description
1	V_{SS}	0V	Ground
2	V_{DD}	5.0V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H:DATA, L:Instruction code
5	R/W	H/L	H:Read(MPU→Module)L:Write(MPU→Module)
6	E	H,H→L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	A	—	Power supply for LED backlight (+)
16	K	—	Power supply for LED backlight (-)

NOTE- PIN7 TO PIN10 ARE NOT CONNECTED AS LCD IS USED IN 4 BIT MODE.

There are also eight bidirectional data lines which carry data or command during write mode and data or status of LCD during read mode. The D7 line is connected to the Busy flag which indicates status of LCD. LCD screen can operate in 4-bit mode or 8-bit mode. In 4-bit mode, only four data lines are needed, but the program becomes more complex. In 8-bit mode, all eight data lines are used, but the program becomes simple. We have used 4-bit mode for interfacing LCD. The L+ and L- lines are the power lines to the backlight of LCD. Vic is connected to the +5V supply and Vss is connected to ground. Vee is used for brightness control depending upon value of potentiometer. This LCD is a 20x4 dot matrix LCD and has its own controller which performs refreshing of each pixel. Hence it is called intelligent LCD.

- **PIR sensor**

An individual PIR sensor detects changes in the amount of infrared radiation impinging upon it, which varies depending on the temperature and surface characteristics of the objects in front of the sensor. When an object, such as a human, passes in front of the background, such as a wall, the temperature at that point in the sensor's field of view will rise from room temperature to body temperature, and then back again. The sensor converts the resulting change in the incoming infrared radiation into a change in the output voltage, and this triggers the detection. Objects of similar temperature but different surface characteristics may also have a different infrared emission pattern, and thus moving them with respect to the background may trigger the detector as well.

Fig 6.4 : PIR Sensor(HC-SR 505)

PIRs come in many configurations for a wide variety of applications. The most common models have numerous Fresnel lenses or mirror segments, an effective range of about ten meters (thirty feet), and a field of view less than 180 degrees. Models with wider fields of view, including 360 degrees, are available—typically designed to mount on a ceiling. Some larger PIRs are made with single segment mirrors and can sense changes in infrared energy over thirty meters (one hundred feet) away from the PIR. There are also PIRs designed with reversible orientation mirrors which allow either broad coverage (110° wide) or very narrow "curtain" coverage or with individually selectable segments to "shape" the coverage.

- Features
    - Complete with PIR, Motion Detection.
    - Dual Element Sensor with Low Noise and High Sensitivity.
    - Supply Voltage – 5V.
    - Delay Time Adjustable.

7. R305 Fingerprint Sensor

This is an optical biometric fingerprint reader/sensor (R305) module with TTL UART interface for direct connections to a microcontroller UART. The user can store the finger print data in the module and can configure it in 1:1 or 1: N mode for identifying the person. This module can directly interface with any 3.3V or 5V microcontrollers, but a suitable level converter/serial adapter is required for interfacing with the serial port of a PC.

Working principle: -

The R305 is an optical finger print sensor which consist of a powerful AS601 DSP (digital signal processor) chip. The DSP chip does all the image rendering, calculation, feature finding and searching for the sensor. The DSP chip also works as the form of storage device which can store up 120 fingerprints on its flash memory. The chip also has a TTL serial out to connect with a microcontroller or any other system. The sensor registers a finger print by collecting the image copy of the fingerprint and store it for future use, and when repeated the sensor once again collect a new picture of a fingerprint and compare it with the store fingerprint in order the identify the individual.

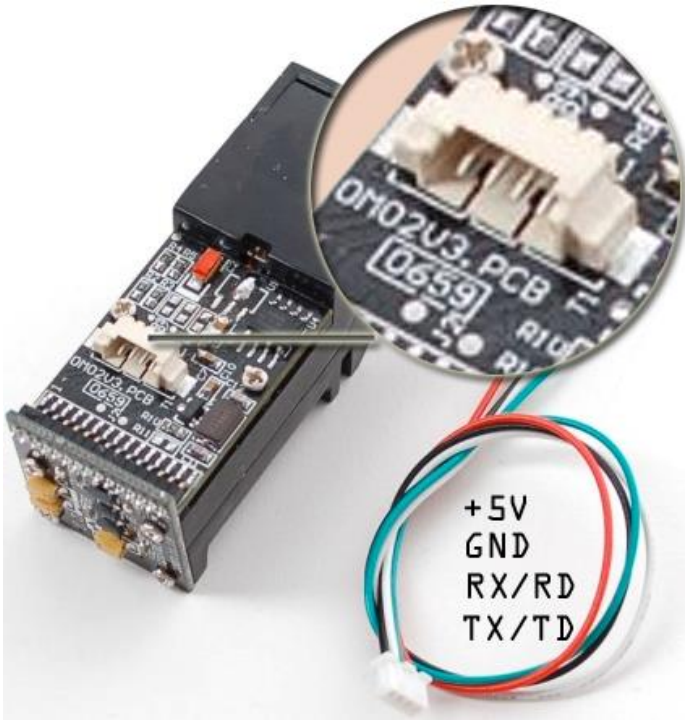


Fig 7.1 : pinout of R-305 sensor

Fingerprint processing includes two parts, fingerprint enrolment and fingerprint matching (the matching can be 1:1 or 1: N). When enrolling, user needs to enter the finger two times. The system will process the two-time finger images, generate a template of the finger based on processing results and store the template. When matching, user enters the finger through optical sensor and system will generate a template of the finger and compare it with templates of the finger library.

For 1:1 matching, system will compare the live finger with specific template designated in the Module; for 1: N matching, or searching, system will search the whole finger library for the matching finger. In both circumstances, system will return the matching result, success or failure.

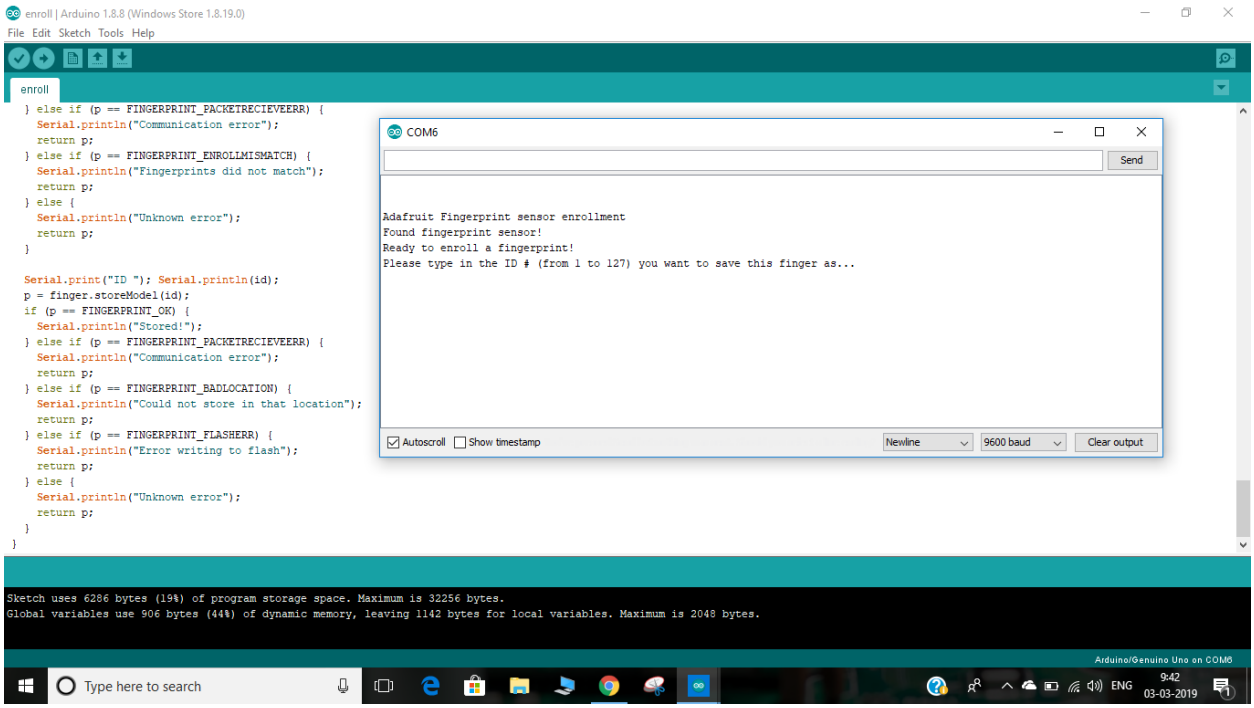
Fig 7.2 : R305 Fingerprint Sensor

Features:

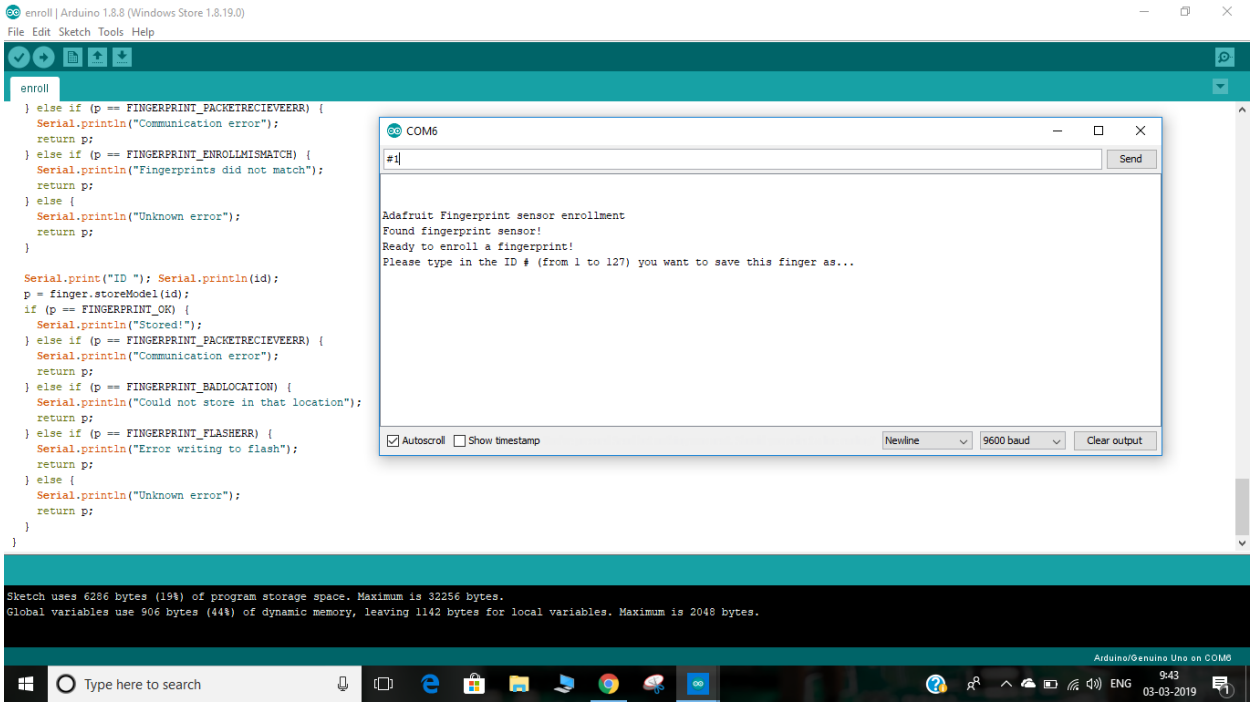
- Power DC: 3.6V-6.0V.
- Interface: UART (TTL logical level)/ USB 1.1.
- Working current: 100mA.
- Peak Current: 150mA.
- Matching Mode: 1:1 and 1: N.
- Baud rate (9600*N) bps, N=1-12 (default N=6 57600bps)
- Character file size: 256 bytes.

Procedure to stored and detection of fingerprint using R-305 sensor :-

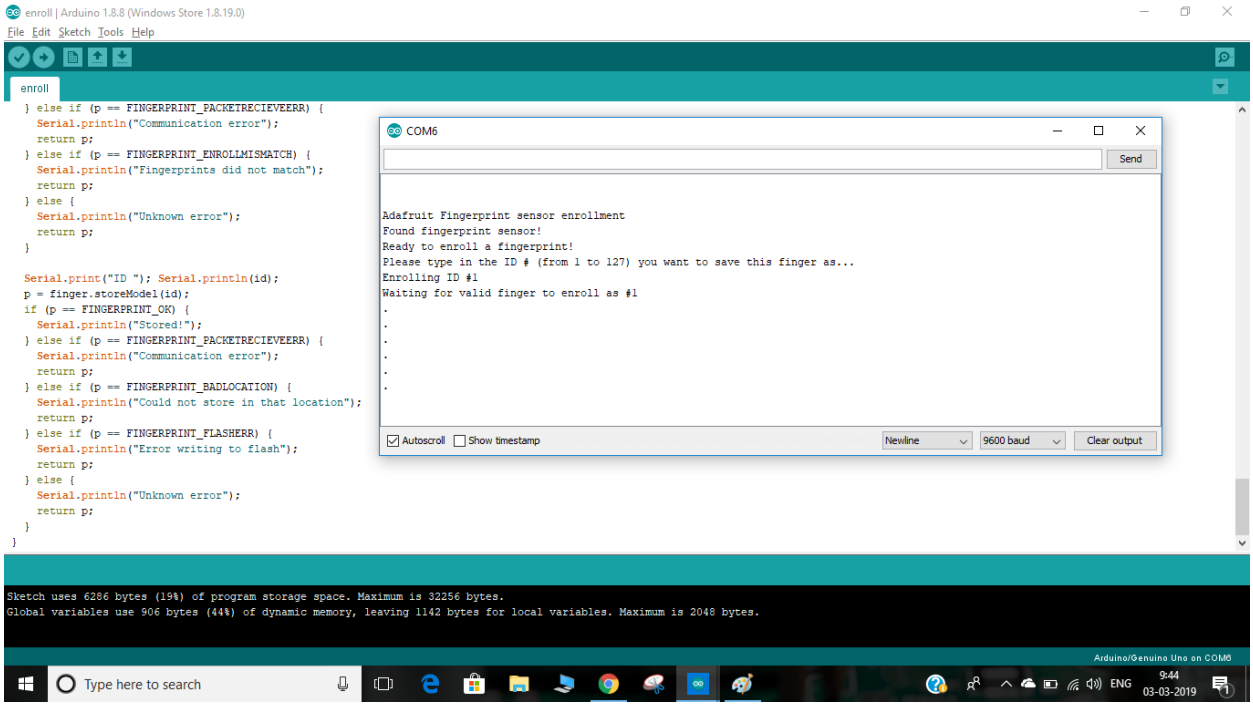
1) Connect the sensor to Arduino with jumper wire as per the circuit diagram and connect the Arduino to system with A to B USB cable. Go to the "Examples→ Adafruit fingerprint sensor→ Enrol" & open the program of enrolling the fingerprint into the flash memory of sensor.

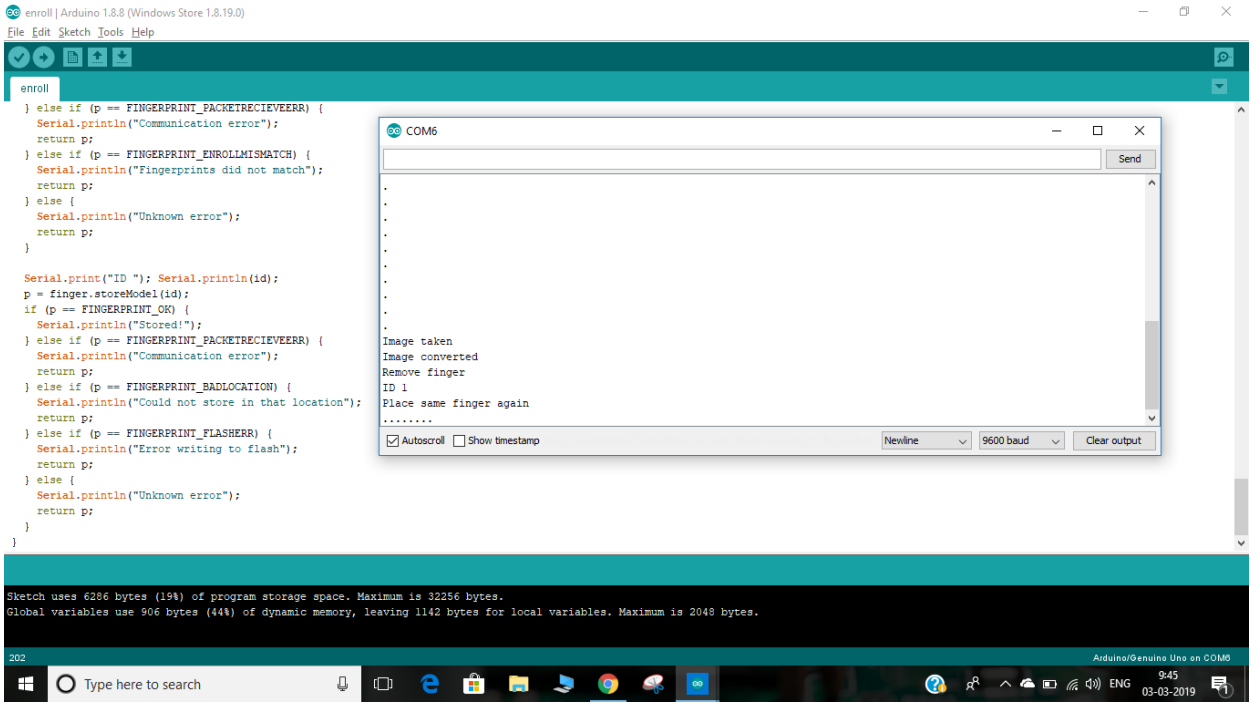


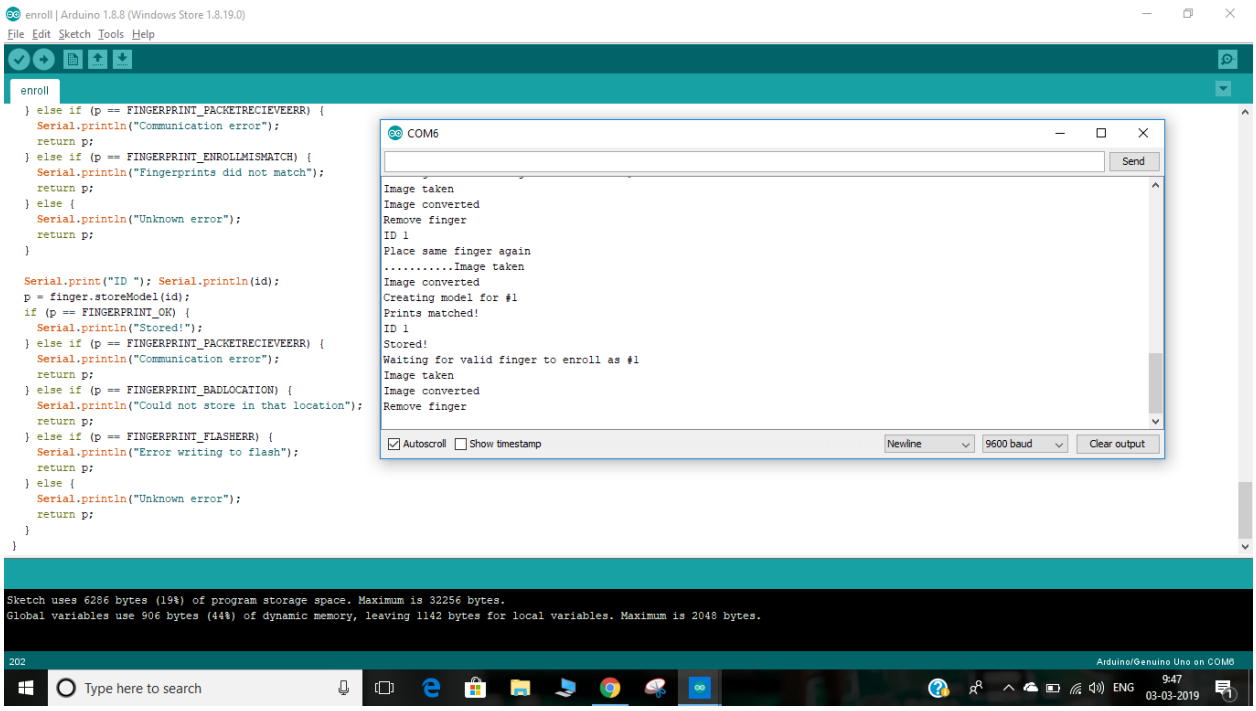
Give the allocation ID to each detected fingerprint. Before to allocate the id just write the "#" this symbol. It takes it as the ID of the fingerprint. After detection of the sensor it will take some time to calibrate and get ready for scanning the prints.



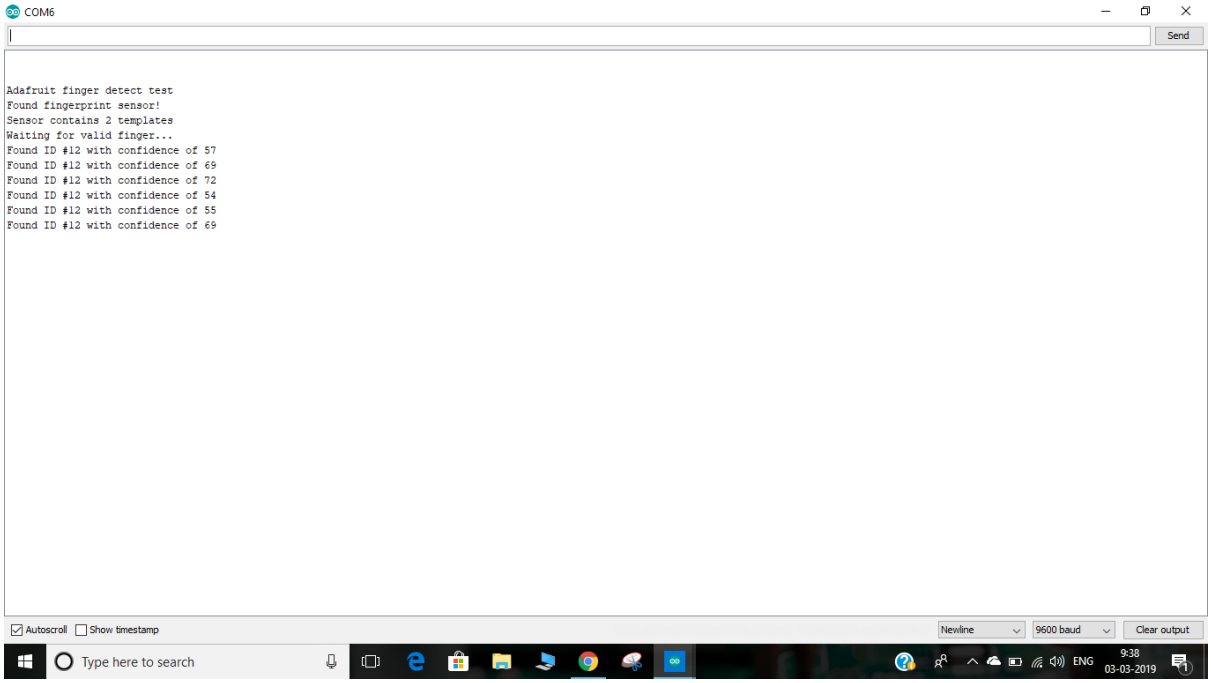
2) Put the finger on fingerprint sensor and it will take the first impression of your finger and ask same finger for confirmation of same persons or accuracy of the print. When the second print is taken by the sensor its stores the print in digital form and ask for the second finger for scanning.







3) After storing all the fingerprint, we upload the fingerprint detection program. . Go to the "Examples→ Adafruit fingerprint sensor→ Fingerprint Detection". Open the program and upload the program in the ARDUINO UNO board. Now open the serial monitor put the finger on sensor and wait for the detection, if the finger is correct, it will show the message.


COM6
Send
Adafruit finger detect test
Found fingerprint sensor!
Sensor contains 2 templates
Waiting for valid finger...
Found ID #12 with confidence of 57
Found ID #12 with confidence of 69
Found ID #12 with confidence of 72
Found ID #12 with confidence of 54
Found ID #12 with confidence of 55
Found ID #12 with confidence of 69
Autoscroll
Show timestamp
Newline
9600 baud
Clear output
Type here to search
ENG
9:38
03-03-2019

8. SIM800L GSM Module

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

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

SIM800L is a miniature cellular module which allows for GPRS transmission, sending and receiving SMS and making and receiving voice calls. Low cost and small footprint and quad band frequency support make this module perfect solution for any project that require long range connectivity. After connecting power module boots up, searches for cellular network and login automatically. On board LED displays connection state (no network coverage - fast blinking, logged in - slow blinking).

Specifications:

- Recommended supply voltage: 4V
- Supply voltage: 3.8V - 4.2V
- Power consumption:
- sleep mode < 2.0mA
- idle mode < 7.0mA
- GSM transmission (avg): 350 mA
- GSM transmission (peek): 2000mA
- Interface: UART (max. 2.8V) and AT commands
- SIM card socket: microSIM (bottom side)
- Supported frequencies: Quad Band (850 / 950 / 1800 /1900 MHz)
- Antenna connector: IPX
- Working temperature range: -40 do + 85 ° C

PINOUT OF SIM 800L: -

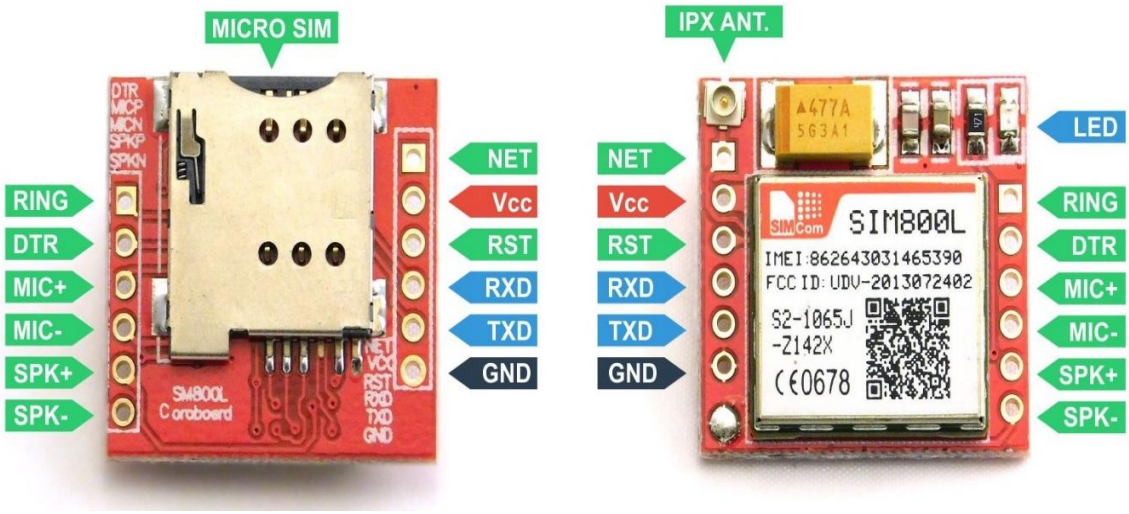


Fig 8.1 : pinout of SIM800L GSM module

9. IR sensor

An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measures only infrared radiation, rather than emitting it that is called as a passive IR sensor. Usually in the infrared spectrum, all the objects radiate some form of thermal radiations. These types of radiations are invisible to our eyes, that can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and these output voltages, change in proportion to the magnitude of the IR light received.

An infrared sensor circuit is one of the basic and popular sensor module in an electronic device. This sensor is analogous to human's visionary senses, which can be used to detect obstacles and it is one of the common applications in real time. This circuit comprises of the following components

- LM358 IC 2 IR transmitter and receiver pair
- Resistors of the range of kilo ohms.
- Variable resistors.
- LED (Light Emitting Diode).

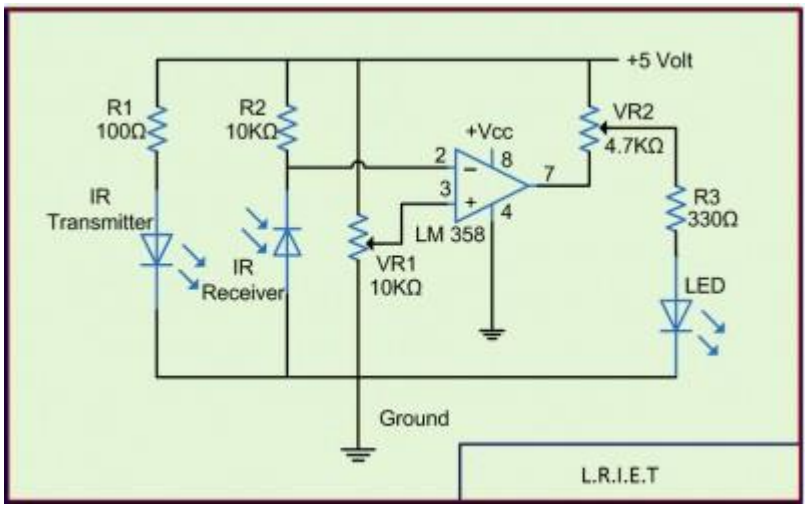


Fig 9.1 : working of IR sensor

In this project, the transmitter section includes an IR sensor, which transmits continuous IR rays to be received by an IR receiver module. An IR output terminal of the receiver varies depending upon its receiving of IR rays. Since this variation cannot be analysed as such, therefore this output

can be fed to a comparator circuit. Here an operational amplifier (op-amp) of LM 339 is used as comparator circuit.

When the IR receiver does not receive a signal, the potential at the inverting input goes higher than that non-inverting input of the comparator IC (LM339). Thus, the output of the comparator goes low, but the LED does not glow. When the IR receiver module receives signal to the potential at the inverting input goes low. Thus, the output of the comparator (LM 339) goes high and the LED starts glowing. Resistor R1 (100), R2 (10k) and R3 (330) are used to ensure that minimum 10 mA current passes through the IR LED Devices like Photodiode and normal LEDs respectively.

Resistor VR2 (preset=5k) is used to adjust the output terminals. Resistor VR1 (preset=10k) is used to set the sensitivity of the circuit Diagram. Read more about IR sensors.

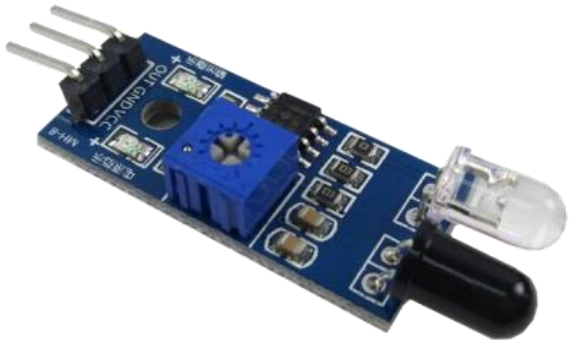


Fig 9.2 : IR module

Specifications:

- Operating Voltage: 3.0V – 5.0V
- Detection range: 2cm – 30cm (Adjustable using potentiometer)
- Current Consumption: at 3.3V : ~23 mA, at 5.0V: ~43 mA
- Active output level: Outputs Low logic level when obstacle is detected
- On board Obstacle Detection LED indicator

- **Solenoid lock**

A solenoid is a coil of insulated or enamelled wire wound on a rod-shaped form made of solid iron, solid steel, or powdered iron. Devices of this kind can be used as electromagnets, as inductors in electronic circuits, and as miniature wireless receiving antennas.

In a solenoid, the core material is ferromagnetic, meaning that it concentrates magnetic lines of flux. This increases the inductance of the coil far beyond the inductance obtainable with an air-core coil of the same dimensions and the same number of turns. When current flows in the coil, most of the resulting magnetic flux exists within the core material. Some flux appears outside the coil near the ends of the core; a small amount of flux also appears outside the coil and off to the side.

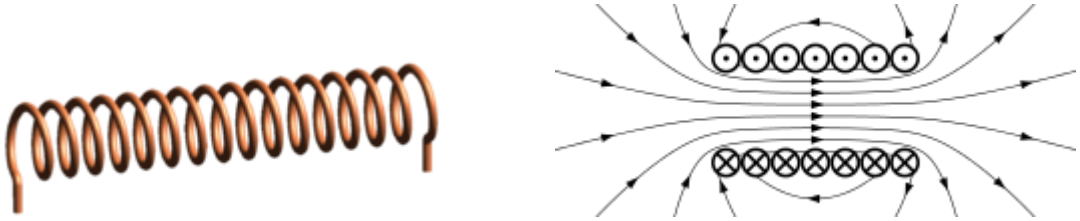


Fig 9.3 : working of solenoid lock

A solenoid chime is wound on a cylindrical, hollow, plastic or phenolic form with a movable, solid iron or steel core. The core can travel in and out of the coil along its axis. The coil is oriented vertically; the core normally rests somewhat below the coil centre. When a current pulse is applied to the coil, the magnetic field pulls the core forcefully upward. Inertia carries the core above the centre of the coil, where the core strikes a piece of metal like a xylophone bell, causing a loud “ding”.

In this project we are using the solenoid lock for lock and unlock the door of laboratory. Whenever the electric current passes through out the coil of copper wire, the magnetic field is generated, and the linear motion is getting placed.

Fig 9.4 : solenoid lock

ADVANTAGES

- Remote operation: Magnetic locks can be turned on and off remotely by adjusting the power source.
- Easy to install: Magnetic locks are generally easier to install than other locks since there are no interconnecting parts.
- Quick to operate: Magnetic locks unlock instantly when the power is cut, allowing for quick release in comparison to other locks.
- Sturdy: Magnetic locks may also suffer less damage from multiple blows than do conventional locks. If a magnetic lock is forced open with a crowbar, it will often do little or no damage to the door or lock. There are no moving parts in an electromagnetic lock to break.

DISADVANTAGES

- Requires a constant power source to be secure.
- Can de-energize in the event of a power outage, disabling security.
- Expensive in comparison to mechanical locks.
- Requires additional hardware for safe operation.

- **IN4007 Diode**

Fig 9.4 **:** P-N junction diode

1N4007 is a PN junction rectifier diode. These types of diodes allow only the flow of electrical current in one direction only. So, it can be used for the conversion of AC power to DC. 1N 4007 is electrically compatible with other rectifier diodes and can be used instead of any of the diode belonging to 1N400X series. 1N-4007 has different real-life applications e.g. freewheeling diodes applications, general purpose rectification of power supplies, inverters, converters etc.

FEATURES

- Glass passivated chip junction
- High current capability, Low VF
- High reliability
- High surge current capability
- Low power loss, high efficiency
- Compliant to RoHS Directive 2011/65/EU and in accordance to WEEE 2002/96/EC

APPLICATIONS

- Rectifying a voltage, such as turning the AC into DC voltages
- Isolating signals from a supply
- Voltage Reference
- Controlling the size of a signal
- Mixing signals
- Detection signals
- Lighting systems

- **30N06L (N-channel MOSFET)**

The UTC is a low voltage MOSFET and is designed to have better characteristics, such as fast switching time, low gate charge, low on-state resistance and excellent avalanche characteristics. This power MOSFET is usually used at automotive applications in power supplies, highly efficient to DC converters and battery-operated products.

The MOSFET (Metal Oxide Semiconductor Field Effect Transistor) transistor is a semiconductor device which is widely used for switching and amplifying electronic signals in the electronic devices. The MOSFET is a three terminal device such as source, gate, and drain. The MOSFET is very far the most common transistor and can be used in both analogy and digital circuit.

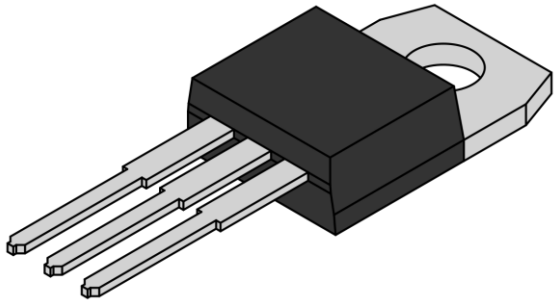


Fig 9.5 : 30N06L MOSFET

That the N-channel, Enhancement-mode MOSFET (e-MOSFET) operates using a positive input voltage and has an extremely high input resistance (almost infinite) making it possible to interface with nearly any logic gate or driver capable of producing a positive output.

APPLICATIONS

- MOSFET Used as a Switch
- Auto Intensity Control of Street Lights using MOSFET
- Marx Generator Based High Voltage Using MOSFETs
- Marx Generator Based High Voltage Using MOSFETs
- LDR Based Power Saver for Intensity Controlled Street Light
- SVPWM (Space Vector Pulse Width Modulation)

- # L7805 VOLTAGE REGULATORS

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.

Pin No.	Pin	Function	Description
1	INPUT	Input voltage (7V-35V)	In this pin of the IC positive unregulated voltage is given in regulation.
2	GROUND	Ground (0V)	In this pin where the ground is given. This pin is neutral for equally the input and output.
3	OUTPUT	Regulated output; 5V (4.8V-5.2V)	The output of the regulated 5V volt is taken out at this pin of the IC regulator.

7805 IC Rating

- Input voltage range 7V- 35V
- Current rating I_c = 1A
- Output voltage range $V_{Max=5.2V}$, $V_{Min=4.8V}$

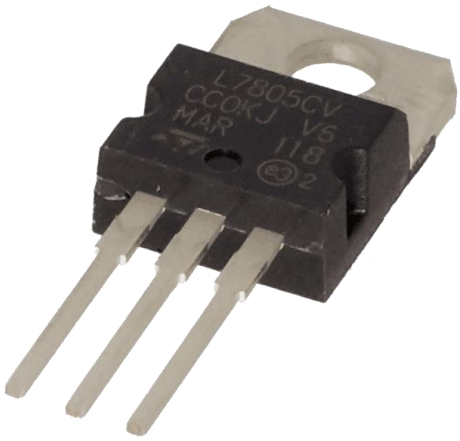


Fig 9.6 : 7805 voltage regulators

- **Application areas for 7805 IC**

7805 IC is used in a wide range of circuits. The major ones being:

- Fixed-Output Regulator
- Adjustable Output Regulator
- Current Regulator
- Adjustable DC Voltage Regulator
- Regulated Dual-Supply
- Output Polarity-Reversal-Protection Circuit
- Reverse bias projection Circuit

10. RELAY MODULE (2 CHANNELS)

A **relay** is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

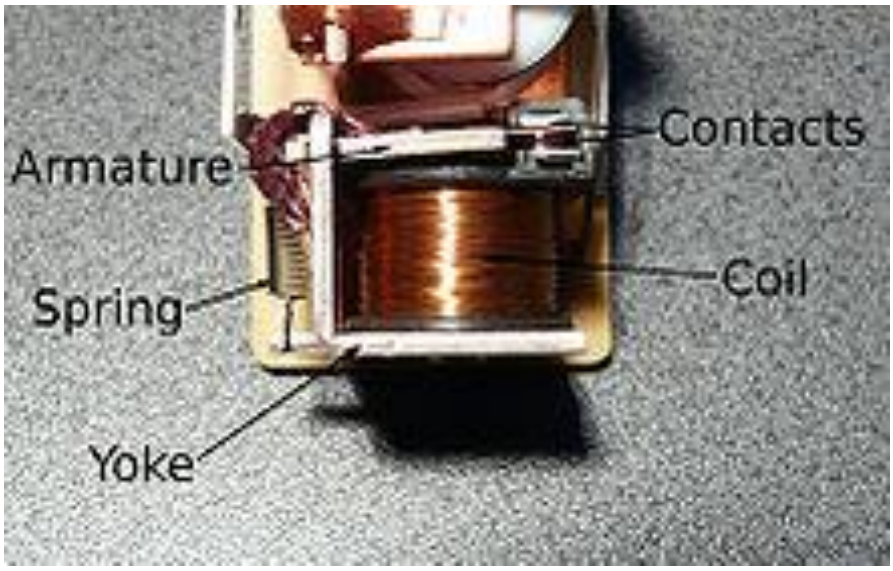


Fig 10.1 : Basic Electromagnetic Relay

Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in applications where interrupted power should not be able to transition the contacts.

Magnetic latching relays can have either single or dual coils. On a single coil device, the relay will operate in one direction when power is applied with one polarity and will reset when the polarity is reversed. On a dual coil device, when polarized voltage is applied to the reset coil the contacts will transition. AC controlled magnetic latch relays have single coils that employ steering diodes to differentiate between operate and reset commands.

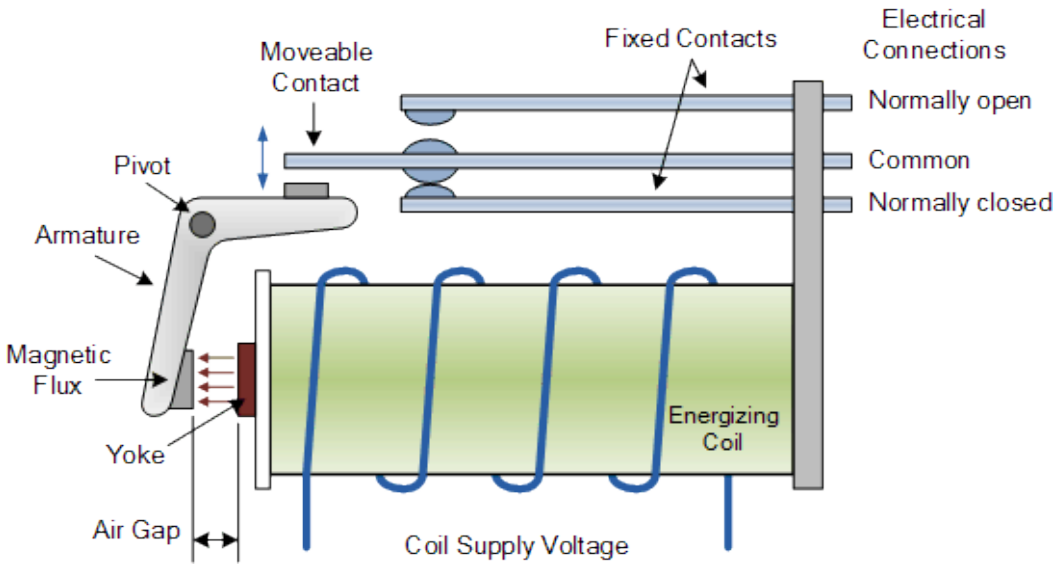


Fig 10.2 : Construction of Relay

When an electric current is passed through the coil it generates a magnetic field that activates the armature, and the consequent movement of the movable contact(s) either makes or breaks a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly.

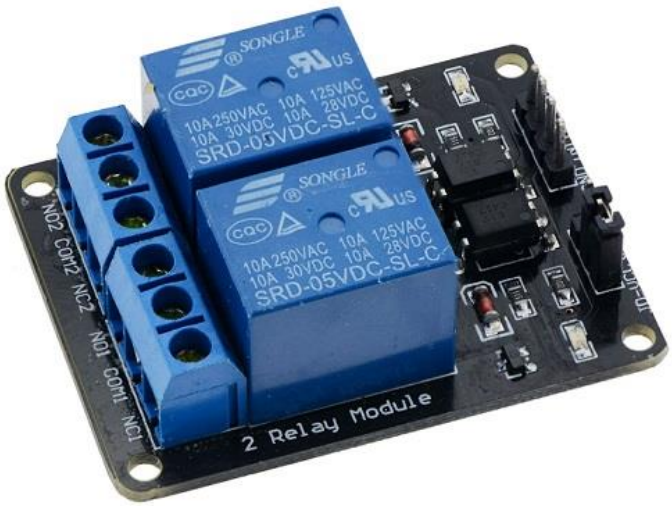


Fig 10.3 : 4-Channel Electromechanical Relay

11. FIRMWRE REQIRED

In smart la security system we are using the two basic software namely as “ARDUINO” & “CIRCUITO.IO”. in the “ARDUINO” we are write our firmware for the project. And, the “CIRCUITO.IO” is used for the designing purpose of circuit diagram. You can also write project code in the circuit.io, where the program is automatically written by software. Let’s get more information about this software: -

➢ ARDUINO

The **Arduino integrated development environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board.

Fig 11.1 : ARDUINO IDE

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. 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.

The Arduino IDE employs the program argued 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.

First download the ARDUINO IDE from www.arduino.cc. then add the libraries in to the IDE then past the code from circuito.io. then compile the code for checking the errors, if there is any error solve the error and upload to the board. Before you click on the upload button just get confirmed you selected the appropriate COM port and Board. After the burning of code is done we can reboot the controller and power it up.

- **CIRCUITO.IO**

circuito.io gives you an opportunity to focus on the essence of your idea, while we solve the nitty gritty details of the design & automagically compute a detailed Bill of materials, wiring instructions and sample code for you to start building.

Fig 11.2 : Circuito.io

No more searches for samples of code and schematic that won't do the work. It's here, and it's free, as we believe everyone should have the tools to build their next crazy invention.

In this software we can make the schematic circuit for any project. For make this schematic visit the www.circuito.io. First you must choose the controller which you have to use in your project.in

my case it is a “ARDIUNO UNO”. the total description of” ARDIUNO UNO” is given separately. So just take the first component to interface with “ARDIUNO UNO”. We are taking the PIR sensor, just search on the search column and drag the component in the working place. It will do

directly wiring of the component to ARDIUNO UNO. After that take one by one components and place according your priorities.

The circuio.io will connect all the wiring in different colours shades to identify each connection of each component. Then we must turn to firmware page. Just click on the next to the schematic. The coding window will get opened and it will automatically generate the required code for the schematic which you are designed on the designed window. Just read the code and do editing if any.

Now we will go for the next step of connect all the component on breadboard. The circuito.io give the instruction as simple as the way of doing connection. First mount the ARDIUNO UNO on the centre of the breadboard. And placed the all component as per shown in the schematic. Using the jumper wire we can connect all component to ARDIUNO UNO. Then copy the code from the circuito.io and paste it into Arduino IDE software. Burn the code into the ARDIUNO UNO.

12. PROGRAM OF THE PROJECT

- **IR DETECTION**

```
#include<LiquidCrystal.h>

LiquidCrystal lcd(13,12,11,10,9,8);

#define in 14

#define out 19

#define relay 4
```

```
int count=0;

int sensorPin = 2;

int pirState = LOW;

int val = 0;

int relayPin = 3;

void IN()

{

   count++;

   lcd.clear();

   lcd.print("Person In Room:");

   lcd.setCursor(0,1);

   lcd.print(count);

   delay(1000);

}

void OUT()

{

  count--;

   lcd.clear();

   lcd.print("Person In Room:");

   lcd.setCursor(0,1);

   lcd.print(count);

   delay(1000);
```

```
}
void setup()
{
  lcd.begin(16,2);
  lcd.print("Visitor Counter");
  delay(2000);
  pinMode(in, INPUT);
  pinMode(out, INPUT);
  pinMode(relay, OUTPUT);
  lcd.clear();
  lcd.print("Person In Room:");
  lcd.setCursor(0,1);
  lcd.print(count);
}
{
{
pinMode(sensorPin, INPUT);
pinMode(relayPin, OUTPUT);
Serial.begin(9600);
}
}
void loop()
```

```
{
  if(digitalRead(in))
 IN();
 if(digitalRead(out))
 OUT();
  if(count<=0)
 {
  lcd.clear();
  digitalWrite(relay, HIGH);
  lcd.clear();
  lcd.print("Nobody In Room");
  lcd.setCursor(0,1);
  lcd.print("Light Is Off");
  delay(200);
 }
  else
  digitalWrite(relay, LOW);
 }
{
{
val = digitalRead(sensorPin);
if (val == LOW)
```

```
{
digitalWrite(relayPin, HIGH);
delay(150);
if (pirState == LOW)
{
Serial.println("Motion detected!");
pirState = LOW;
}}
else
{
digitalWrite(relayPin, LOW);
delay(150);
if (pirState == HIGH)
{
Serial.println("Motion ended!");
pirState = LOW;
}}}}
```

- **FINGERPRINT ENROLMENT PROGRAM**

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
```

```
int8_t id;

void setup()

{

  Serial.begin(9600);

  while (!Serial);

  delay(100);

  Serial.println("\n\nAdafruit Fingerprint sensor enrollment");

  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()

{

  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_PACKETRECIEVEERR:
    Serial.println("Communication error");
    break;
   case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
   default:
    Serial.println("Unknown error");
    break;
  }}
 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_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;
}
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_PACKETRECIEVEERR:
    Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
  default:
    Serial.println("Unknown error");
    break;}}
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;
  }
  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_PACKETRECIEVEERR) {
  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 DETECTION PROGRAM**

```
#include <Adafruit_Fingerprint.h>

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()

{

 Serial.begin(9600);

 while (!Serial);

 delay(100);

 Serial.println("\n\nAdafruit finger detect test");

 // 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()

{

  getFingerprintIDez();

  delay(50);

}

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;
 }
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;
 }
  p = finger.fingerFastSearch();
 if (p == FINGERPRINT_OK) {
  Serial.println("Found a print match!");
 } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
  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;
 }
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
 Serial.print(" with confidence of "); Serial.println(finger.confidence);
 return finger.fingerID;
}
```

```
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;



Serial.print("Found ID #"); Serial.print(finger.fingerID);

  Serial.print(" with confidence of "); Serial.println(finger.confidence);

  return finger.fingerID;
```

13. ADVANTAGES OF SYSTEM

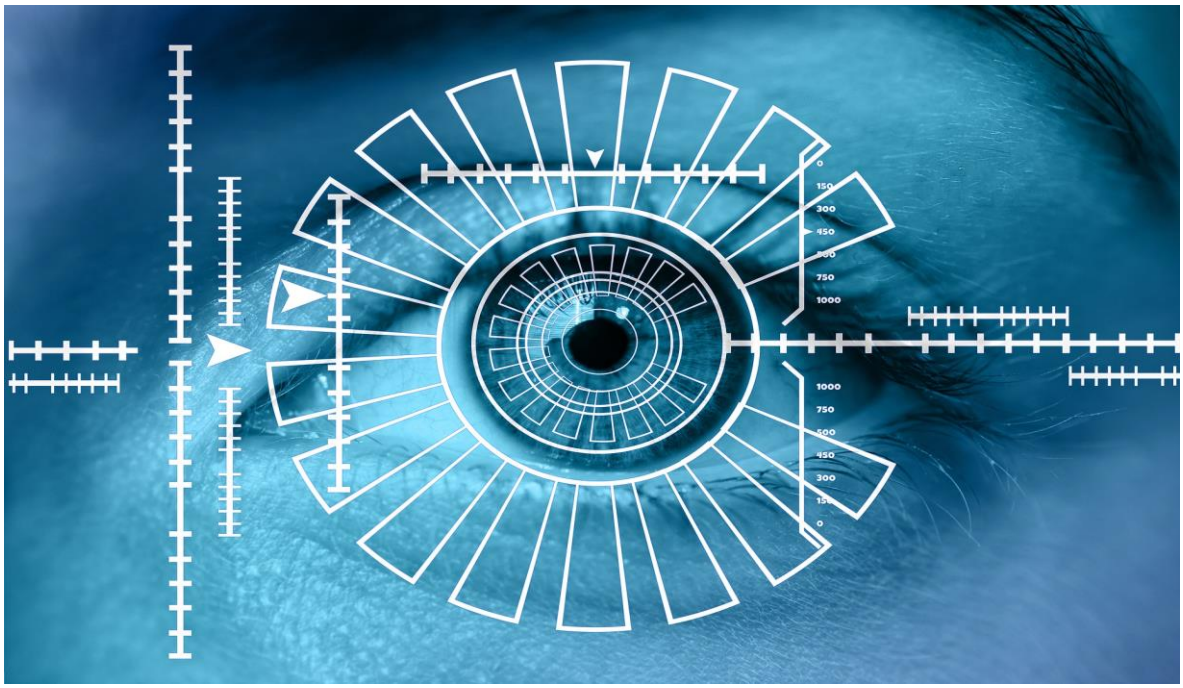
- The system required less maintenance.
- It is a user-friendly system.
- Provide invertor for power backup.
- The range of GSM module is very large, so it can send message to everywhere.
- No access for unauthorized persons.
- Automatic light controller, so it is works as electricity sever.
- Counting the number of persons enter in the lab.

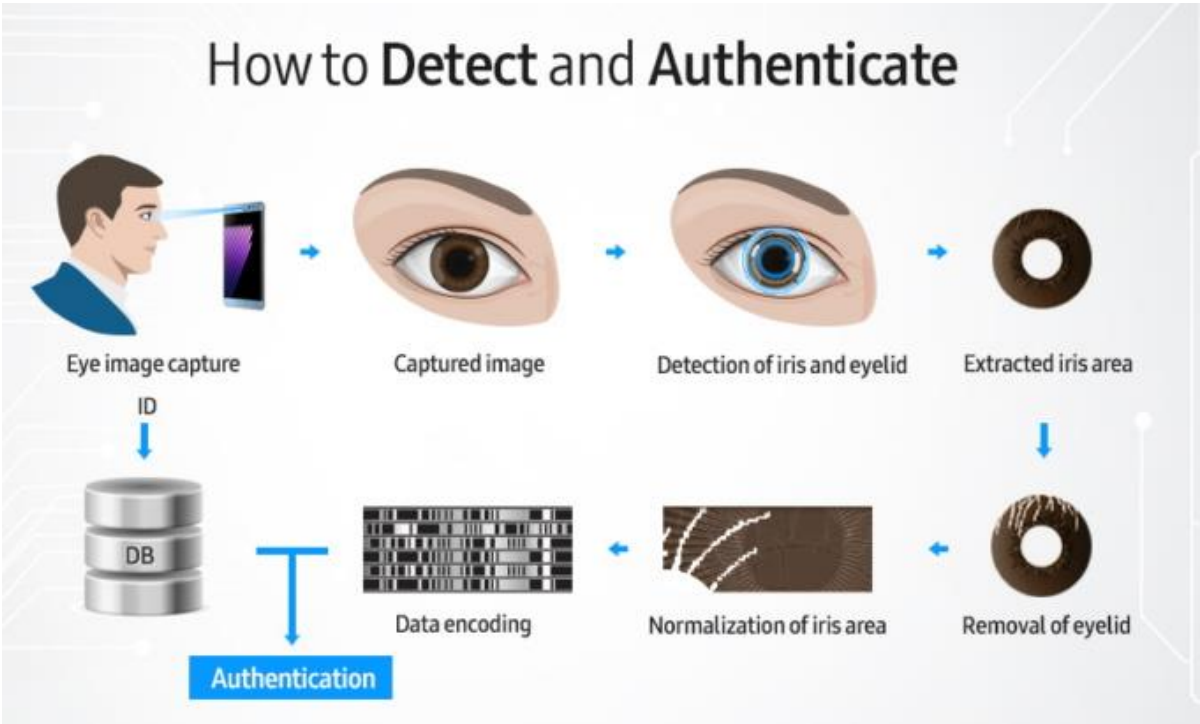
14. DISADVANTAGES OF SYSTEM

- Continues power supply is required, either system get collapse.
- The GSM module is very sensitive for the input power supply, if there is any fluctuation in power supply it may get damaged.

15. Future scope of the project:

1. We can use retina scan security system in this lab security system. for more security options we can use the retina scanner instead of R-305 fingerprint sensor. This is very secure as compare to the R-305 fingerprint sensor but at the same it is very expensive to install.





2. We can also use the DTH-11 temperature and humidity sensor. This sensor measures the temperature of the laboratory and give analogue value to the ARDIUNO UNO & it will show the values on the LCD display. We can also use the LM-35 IC as temperature sensor.

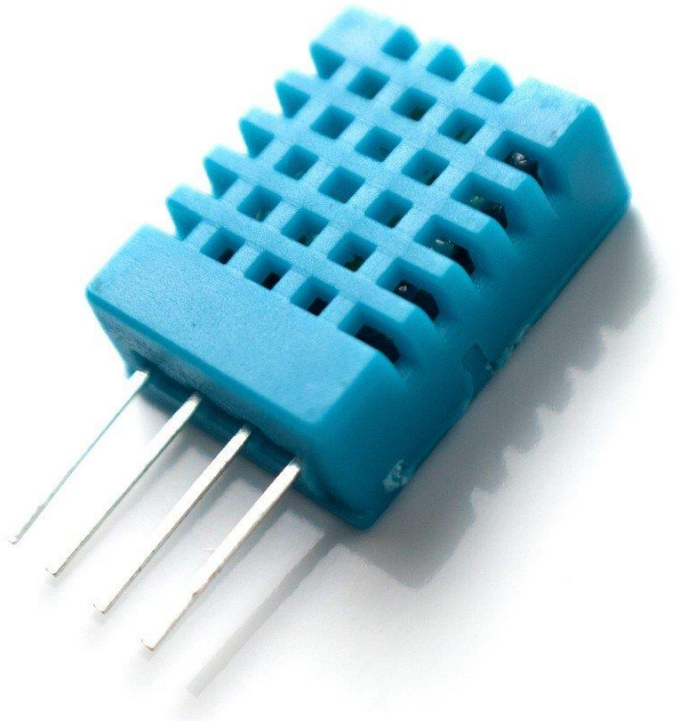


Fig 15.1 : DTH-11 temperature and humidity sensor

Pinout of DTH-11 :-

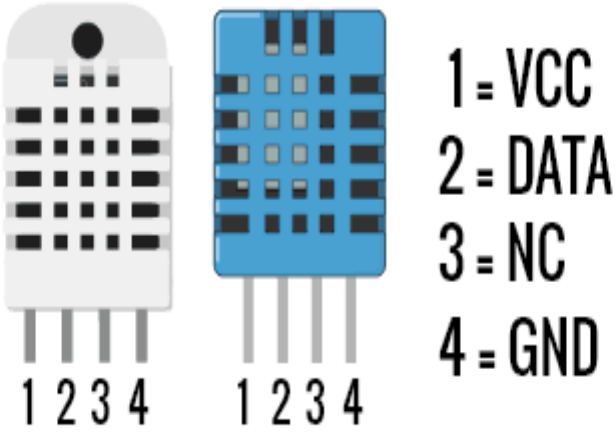


Fig 15.2 : pin diagram of DTH-11

3. We can also see the live footage of the laboratory from anywhere. For this we must use the ARDIUNO UNO as cloud server. Using this cloud server, we can actually see real time live camera footage.

4. We can also use the alphanumeric or OLED display instead of 16*2 display. It gives more flexibility for displaying the data. We can also adjust or set the font size from large to tiny.

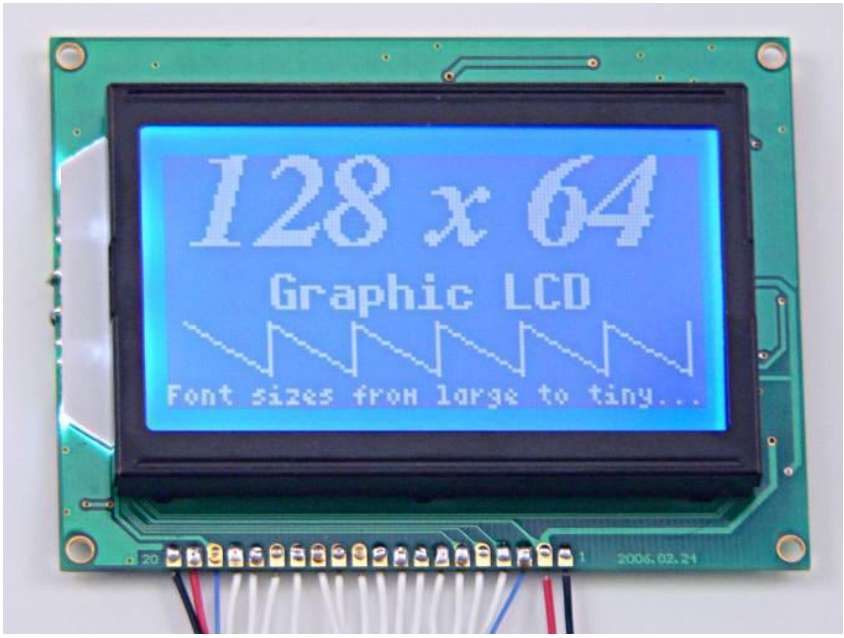


Fig 15.3 : OLED display

5. We can also find the missing equipment's using heart pulse sensor. The pulse sensor will give the continuous pulses, if the pulse is out of the range of the ARDIUNO UNO, it will give indication if form of buzzer and send the message on the registered mobile number.

16. PROBLEMS FACED DURING MAKING OF PROJECT:-

- Procurement of SIM-800L GSM module.
- Interfacing R-305 fingerprint sensor with ARDIUNO UNO.
- Interfacing the SIM-800L GSM module.
- Configuring the GSM module.
- Merging the two program files in Arduino IDE.
- Doing wiring of whole project as per circuit diagram.
- Uploading the main program in the ARDIUNO UNO.

17. REFERENCES :-

While making this project, we referred to several books, technical magazines, websites and visited some technical exhibitions. We have listed these references below.

Books :

- Byron Francis, THE ARDUINO, CreateSpace publication, united states, November 2016.
- Massimo Banzi, MAKE, shroff/maker media, 27 Dec 2014
- Ganesh Akoliya, Elements of Electronics, Nirali Prakashan, January 2014.

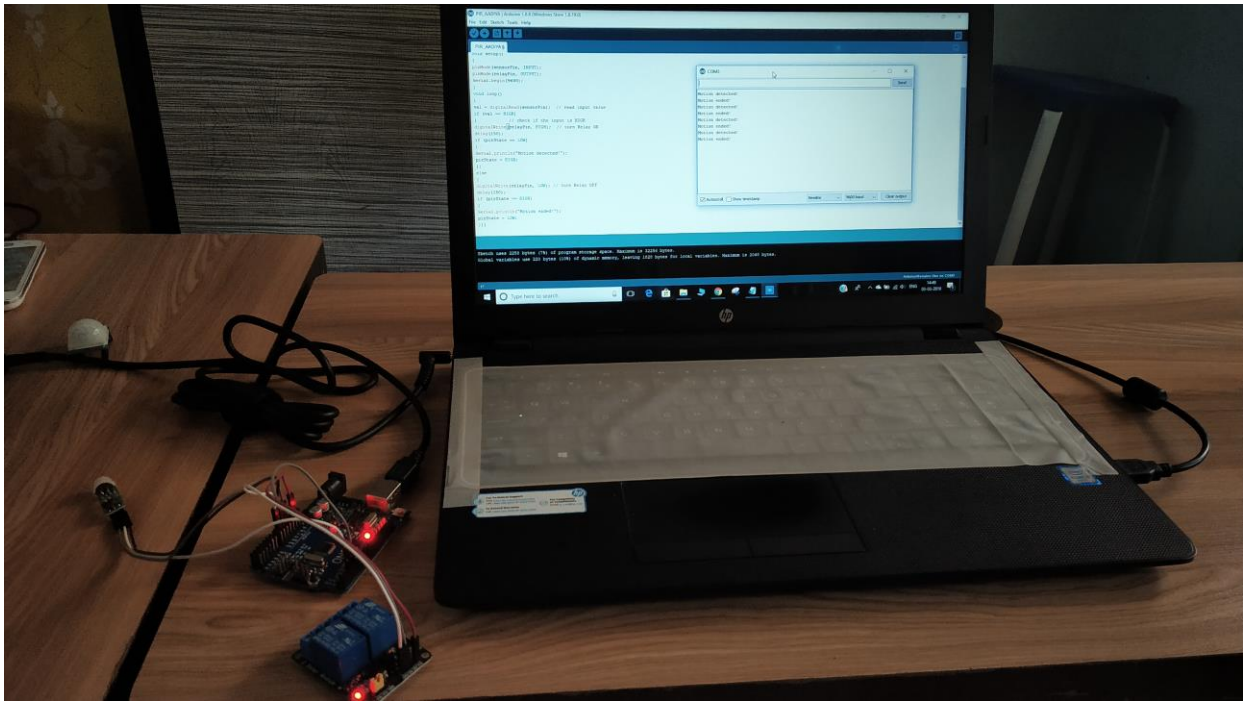
Technical magazines :

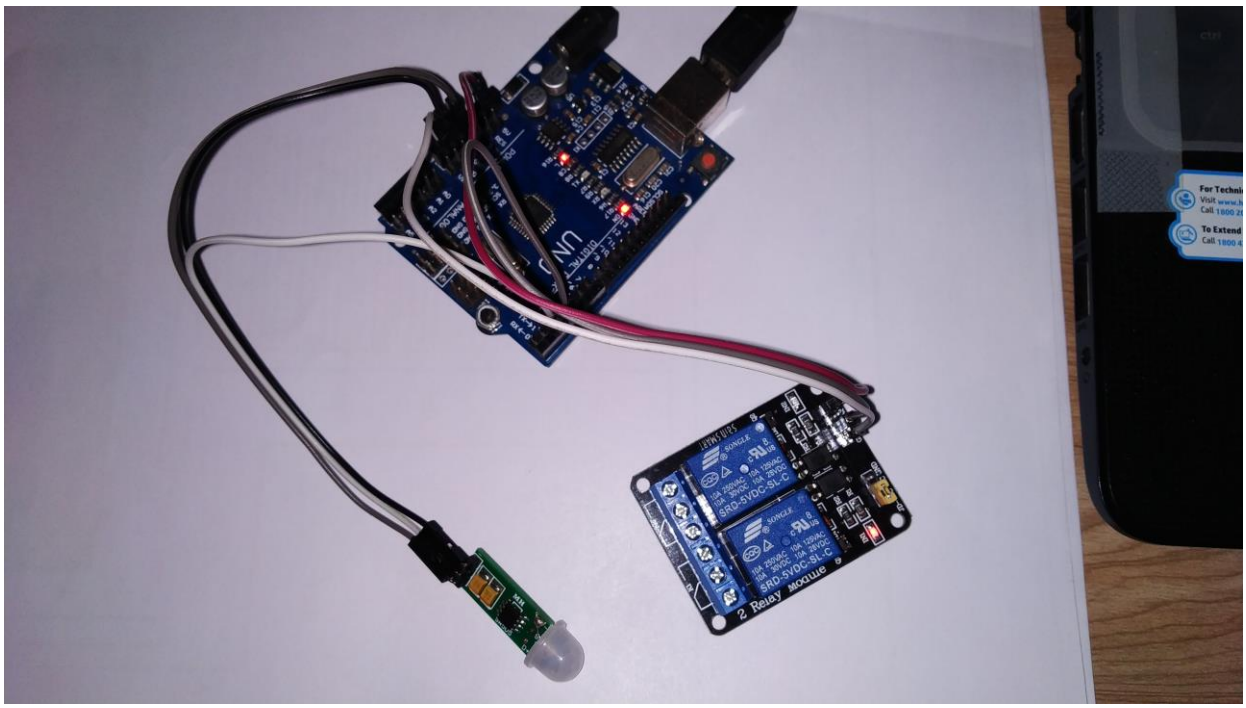
- Electronics for You

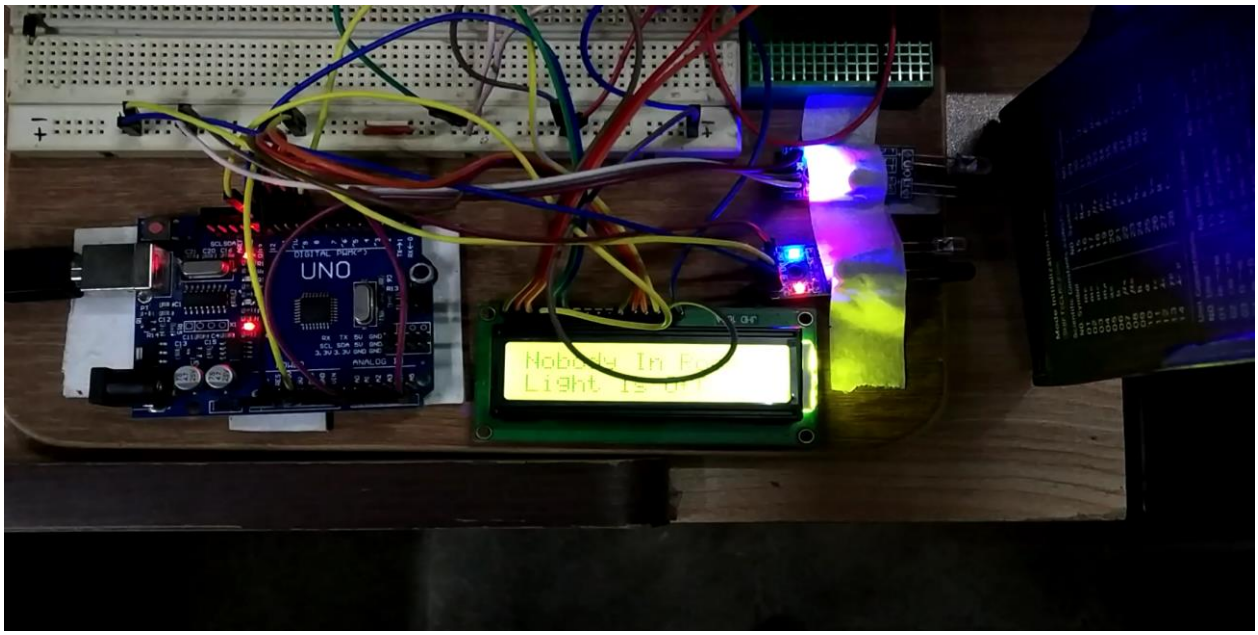
Websites:

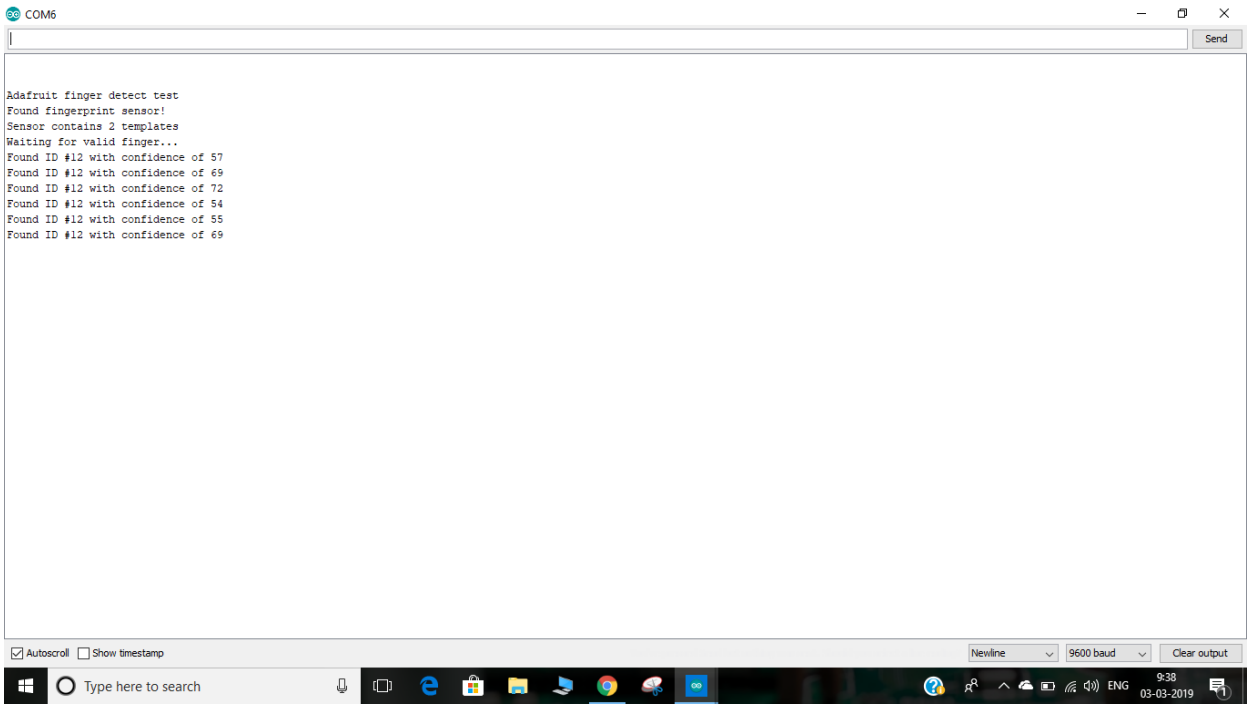
- www.circuito.io
- www.arduino.cc
- www.electroschematics.com
- www.lastminuteengineers.com

18. PHOTOS OF THE PROJECT






UNO


COM6
Send
Adafruit finger detect test
Found fingerprint sensor!
Sensor contains 2 templates
Waiting for valid finger...
Found ID #12 with confidence of 57
Found ID #12 with confidence of 69
Found ID #12 with confidence of 72
Found ID #12 with confidence of 54
Found ID #12 with confidence of 55
Found ID #12 with confidence of 69
Autoscroll
Show timestamp
Newline
9600 baud
Clear output
Type here to search
ENG
9:38
03-03-2019