**FINGERPRINT MODULE INTERFACING WITH ARDUINO UNO**

## Introducing the Fingerprint Sensor Module:

* [Fingerprint sensor modules](https://makeradvisor.com/tools/fingerprint-sensor-module/) ,like the one in the following figure, made fingerprint recognition more accessible and easier to add to your projects.
* This means that is super easy to make fingerprint collection, registration, comparison and search.
* These modules come with FLASH memory to store the fingerprints and work with any microcontroller or system with TTL serial.
* These modules can be added to security systems, door locks, time attendance systems, and much more.



**The working principle of this sensor:**

This fingerprint includes two processes. That is fingerprint registration and fingerprint matching. Also, this fingerprint matching consists of two more parts. That is fingerprint comparison and fingerprint search. When checking fingerprints, the fingerprint image is captured by the sensor and compared with the fingerprint stored in the module. If it matches, it is called a fingerprint comparison. Otherwise, if multiple fingerprints match, it is called a fingerprint search .

* The working principle of the fingerprint sensor mainly depends on the processing.
* The fingerprint processing mainly includes two elements namely enrollment and matching.
* In fingerprint enrolling, every user requires to place the finger twice.
* So that the system will check the finger images to process as well as to generate a pattern of the finger and it will be stored.

When matching, a user places the finger using an optical sensor then the system will produce a pattern of the finger & compares it with the finger library templates

**Specifications of this sensor:**

* Operation Voltage: 3.3v
* Interface: TTL Serial.
* Baud rate: (9600~57600) (default 57600).
* Rated Current: ~120mA.
* Fingerprint imaging time: <1.0 seconds
* Storage Capacity:255 templates

**The wire structure of this sensor:**

* Blue wire – No connection
* White wire – No connection
* Green wire – GND
* Yellow wire – RXD (Serial Data Input With value -2)
* Black wire – TXD (Serial Data Output with value -3)
* Red wire – VCC (+3.3~5 V)

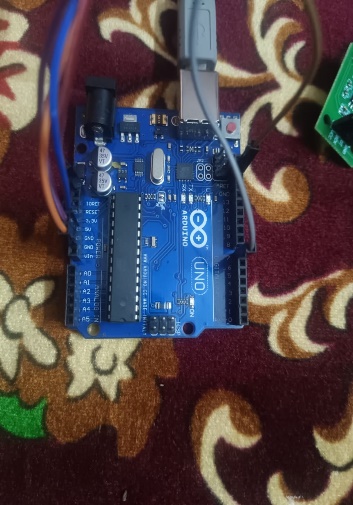
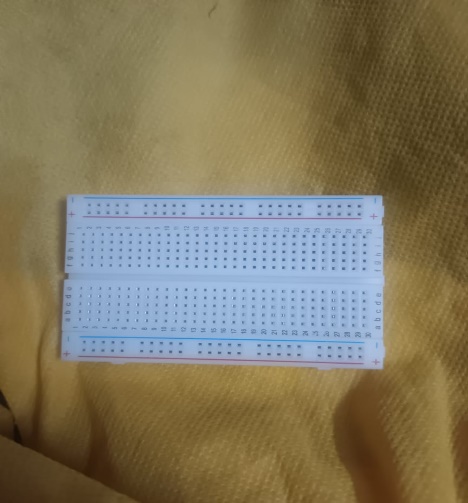
**The required components are given below:**

* Arduino UNO board
* LED bulbs (red, yellow)
* Fingerprint sensor R307S
* Jumper wires
* Mini breadboard

**Step 1:**

Firstly, identify these components.

**ARDUINO UNO BREADBOARD**

# FINGER PRINT MODULE LED

# 

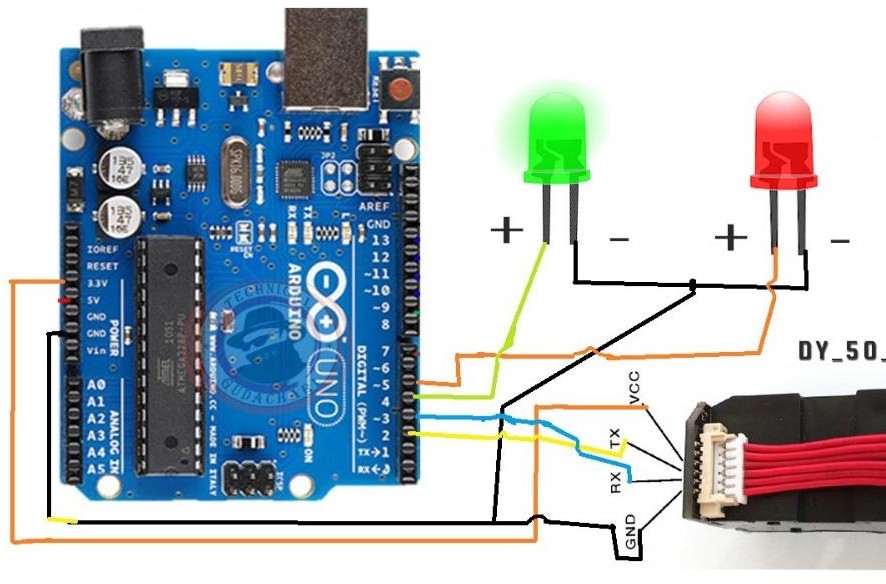
# Step 2:

Secondly, let’s set up the sensor cable. It includes eight wires. But we need four wires for serial communication (Red, Green, Black, Yellow) for that, I have soldered four Male-to-male jumper wires to it. Please use color jumper wires that match the sensor wire colors . Because you can easily identify.

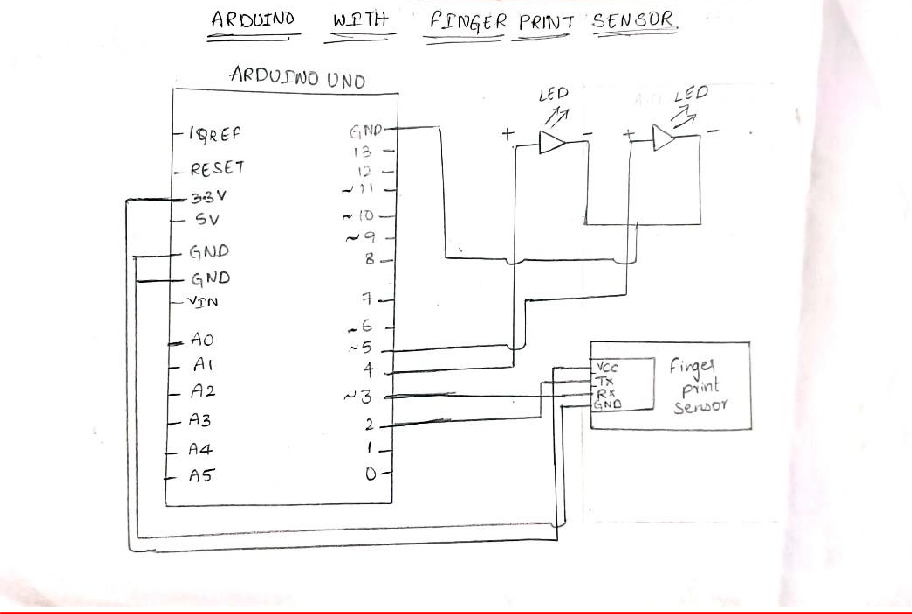
**Step 3:**

Thirdly, connect the fingerprint sensor

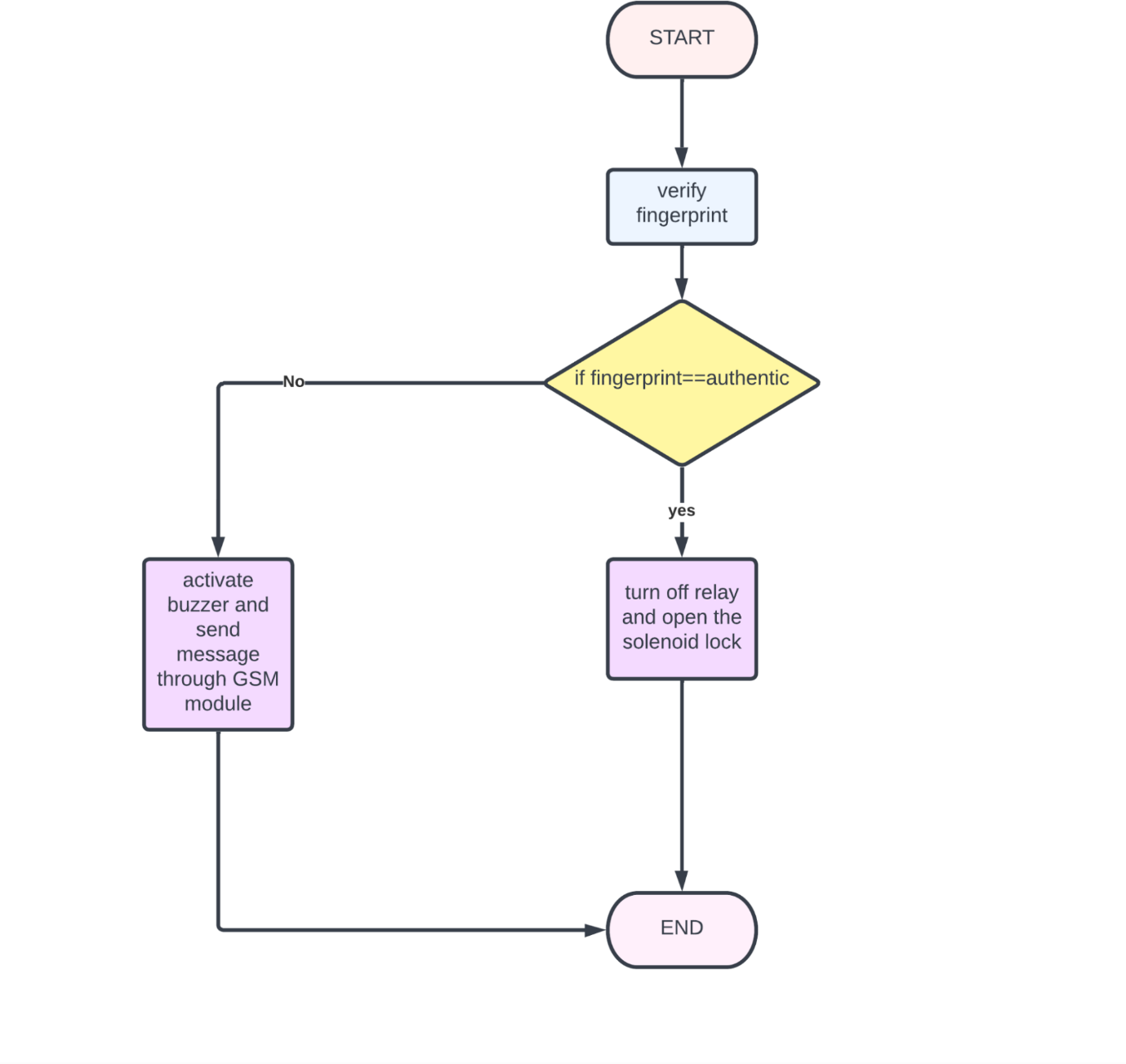
**SKETCH DIAGRAM OF FINGERPRINT MODULE:**



**BLOCK DIAGRAM OF FINGERPRINT MODULE:**

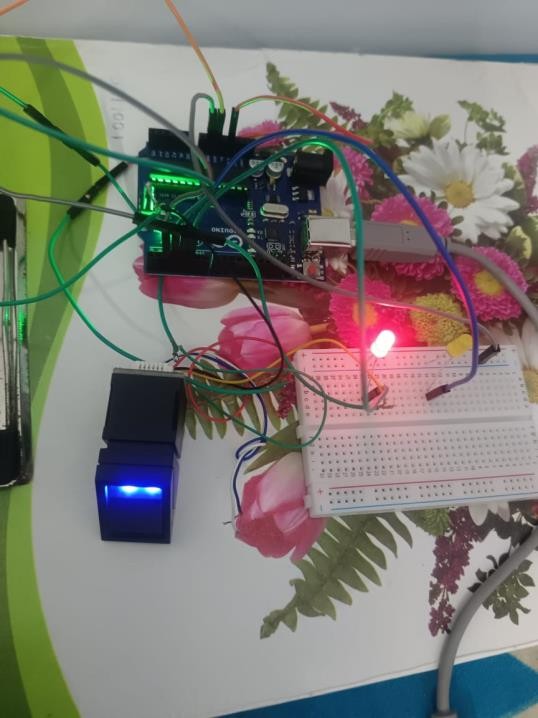


**FLOW CHART :**



**STEP 4:**

Next, place the LED bulb on the breadboard. Then connect it to the Arduino board.



# Step 5:

# After that, connect the Arduino board to the computer and then install the fingerprint library. For that, follow the instructions below

# First open the Arduino IDE and go to the Manage libraries.

# Then type fingerprint in the search bar and install the Adafruit fingerprint sensor library file.

# Step 6:

# Ok, lets enroll the fingerprints. For that follow the instructions below

* First, copy and paste the following program into the Arduino IDE.

1. #include <Adafruit\_Fingerprint.h>
2. // On Leonardo/Micro or others with hardware serial, use those! #0 is green wire, #1 is white
3. // uncomment this line:
4. // #define mySerial Serial1
5. // For UNO and others without hardware serial, we must use software serial...
6. // pin #2 is IN from sensor (GREEN wire)
7. // pin #3 is OUT from arduino (WHITE wire)
8. // comment these two lines if using hardware serial
9. #include <SoftwareSerial.h>
10. SoftwareSerial mySerial(2, 3);
11. Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial);
12. void setup()

13) {

1. pinMode(4,OUTPUT);
2. pinMode(5,OUTPUT);
3. Serial.begin(9600);
4. while (!Serial); // For Yun/Leo/Micro/Zero/...

18) delay(100);

1. Serial.println("\n\nAdafruit finger detect test");
2. // set the data rate for the sensor serial port
3. finger.begin(57600);
4. if (finger.verifyPassword()) {
5. Serial.println("Found fingerprint sensor!");
6. } else {
7. Serial.println("Did not find fingerprint sensor :(");
8. while (1) { delay(1); }

28) }

1. finger.getTemplateCount();
2. Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");
3. Serial.println("Waiting for valid finger...");

32) }

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

34) {

1. getFingerprintIDez();
2. delay(50); //don't ned to run this at full speed.
3. digitalWrite(5,HIGH);

38) }

1. uint8\_t getFingerprintID() {
2. uint8\_t p = finger.getImage();
3. switch (p) {
4. case FINGERPRINT\_OK:
5. Serial.println("Image taken");
6. break;

45) }

1. // OK success!
2. p = finger.image2Tz();
3. switch (p) {
4. case FINGERPRINT\_OK:
5. Serial.println("Image converted");
6. break;
7. // found a match!
8. digitalWrite(5,LOW);
9. digitalWrite(4,HIGH);

55) delay(1000);

56) digitalWrite(4,LOW);

57)

1. Serial.print("Found ID #"); Serial.print(finger.fingerID);
2. Serial.print(" with confidence of "); Serial.println(finger.confidence);
3. return finger.fingerID;

61) }

* Now, select the board and port. After that, click the upload button.
* Next, place any finger on the sensor window.
* Then, follow the instructions on the serial monitor. Finally, you can store your fingerprint ID on the module.
* Next, place any finger on the sensor window. Then, follow the instructions on the serial monitor. Finally, you can store your fingerprint ID on the module.

# Step 7:

OK, now set up the main program.

* First, copy and paste the following program into the Arduino IDE.

1) #include <Adafruit\_Fingerprint.h>

2) #include <SoftwareSerial.h>

3) SoftwareSerial mySerial(2, 3);

4) Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial);

5) Voidsetup()

6) {

7) pinMode(4,OUTPUT);

8) pinMode(5,OUTPUT);

9) Serial.begin(9600);

10) while (!Serial); // For Yun/Leo/Micro/Zero/...

11) delay(100);

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

13) finger.begin(57600);

14) if (finger.verifyPassword()) {

15) Serial.println("Found fingerprint sensor!");

16) } else {

17) Serial.println("Did not find fingerprint sensor :(");

18) while (1) { delay(1); }

19) }

20) finger.getTemplateCount();

21) Serial.print("Sensor contains ");

22) Serial.print(finger.templateCount); Serial.println(" templates");

23) Serial.println("Waiting for valid finger...");

24) }

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

26) {

27) getFingerprintIDez();

28) delay(50);

29) digitalWrite(5,HIGH);

30) }

31) uint8\_t getFingerprintID() {

32) uint8\_t p = finger.getImage();

33) switch (p) {

34) case FINGERPRINT\_OK:

35) Serial.println("Image taken");

36) break;

37) p = finger.fingerFastSearch();

38) if (p == FINGERPRINT\_OK) {

39) Serial.println("Found a print match!");

40) } else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

41) Serial.println("Communication error");

42) return p;

43) } else if (p == FINGERPRINT\_NOTFOUND) {

44) Serial.println("Did not find a match");

45) return p;

46) } else {

47) Serial.println("Unknown error");

48) return p;

49) }

50) // found a match!

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

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

53) return finger.fingerID;

54) }

55) // returns -1 if failed, otherwise returns ID #

56) int getFingerprintIDez() {

57) uint8\_t p = finger.getImage();

58) if (p != FINGERPRINT\_OK) return -1;

59) p = finger.image2Tz();

60) if (p != FINGERPRINT\_OK) return -1;

61) p = finger.fingerFastSearch();

62) if (p != FINGERPRINT\_OK) return -1;

63) // found a match!

64) digitalWrite(5,LOW);

65) digitalWrite(4,HIGH);

66) delay(1000);

67) digitalWrite(4,LOW);

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

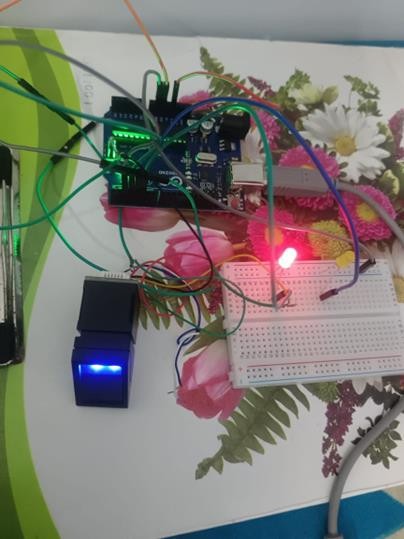
69) Serial.print(" with confidence of "); Serial.println(finger.confidence);

70) return finger.fingerID;

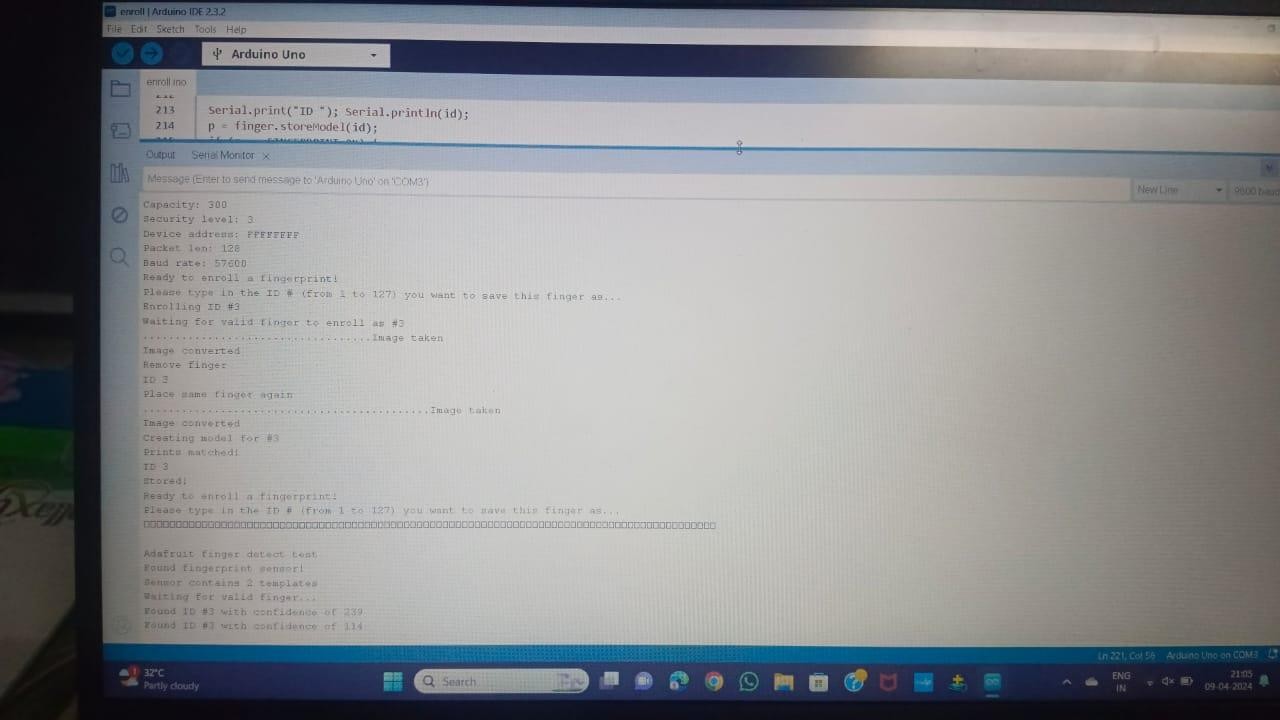
71) }

If you want to delete the Fingerprint IDs. Please visit the fingerprint library example.

Now, you can turn on or off the LED bulb using your fingerprint.



**OUTPUT:**



**REFERENCES:**

[**https://youtu.be/OX-\_D3s04yc?si=vprWZfnobhnKUs5Y**](https://youtu.be/OX-_D3s04yc?si=vprWZfnobhnKUs5Y)**.** [**https://youtu.be/jBFTLJQd8kY?si=gCyIwmH3o9vgqyVU**](https://youtu.be/jBFTLJQd8kY?si=gCyIwmH3o9vgqyVU)**.**

1. **Y. Leng Ng, C. Siong Lim, K. A. Danapalasingam, M. Loong Peng Tan, and C. Wei Tan,(2015) “Automatic Human Guided Shopping cart with Smart Shopping System'', Jurnal Teknologi, vol. 73, no. 3.**
2. **N. Anju Latha, B. Rama Murthy, K. Bharat Kumar,(2016) Distance Sensing with Ultrasonic Sensor and Arduino, International Journal of Advance Research, Ideas and Innovations in Technology.**
3. **J. Borenstein and Y. Koren, (1988)"Obstacle avoidance with ultrasonic sensors," in IEEE Journal on Robotics and Automation, vol. 4, no. 2, pp. 213-218, doi: 10.1109/56.2085.**
4. **G. Roussos,(2006) “Enabling RFID in Retail" IEEE, vol. 39, no. 3.**
5. **.Mr.P. Chandrasekar and Ms.T. Sangeetha,(2014) “Smart Shopping Cart with Automatic Billing System through RFID and ZigBee”, IEEE.**
6. **Stewart Watkiss. “Learn electronics with Raspberry Pi : physical computing with circuits, sensors, outputs, and projects”,Apress, New York.**
7. **S. Son and Y. Shin,(2015) "Design of Smart Shopping Application Using Barcode Scanning and Location Based Coupon Service," 8th International Conference on Grid and Distributed Computing (GDC), 2015, pp. 5-8, doi: 10.1109/GDC.2015.18.**
8. **Russ, J.C. (2006). The Image Processing Handbook (5th ed.). CRC Press.**
9. **I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek,(2012) "A brief introduction to OpenCV," 2012 Proceedings of the 35th International Convention MIPRO,, pp. 1725-1730**
10. **M. E. Joorabchi, A. Mesbah and P. Kruchten,(2013) "Real Challenges in Mobile App Development," ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, 2013, pp. 15-24, doi: 10.1109/ESEM.2013.9.**
11. **E. D. Dickmanns,(2002) "The development of machine vision for road vehicles in the last decade," Intelligent Vehicle Symposium,. IEEE, 2002, pp. 268-281 vol.1, doi: 10.1109/IVS.2002.1187962.**
12. **Setting up your Raspberry Pi - Introduction | Raspberry Pi Projects [Online]. Available:(https://projects.raspberrypi.org/en/projects/raspberry pi-setting-up/0) .**
13. **Build a digital Raspberry Pi Scale (with Weight Sensor HX711)[Online].Available:(https://tutorials raspberrypi.com/digital-raspberry-pi-scale- weight-sensor-hx711/) .**
14. **How to Build a Beginner's Robot Using Raspberry Pi | Raspberry Pi | Maker Pro[Online].Available:(https://maker.pro/raspberry pi/projects/raspberry-pi-robot).**
15. **How to setup a Raspberry Pi RFID RC522 Chip - Pi My Life Up[Online].Available:(https://pimylifeup.com/raspberry-pi-rfid rc522/) .**
16. **Stewart Watkiss. “Learn electronics with Raspberry Pi : physical computing with circuits, sensors, outputs, and projects”,Apress, New York.**
17. **Swati Zope, Maruti Limkar, “RFID based Bill Generation and Payment through Mobile”,**

**Volume 1, Issue 3, June 2012. IJERTV11IS110042 www.ije.**