

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

*DEPARTMENT OF INFORMATION TECHNOLOGY*  April 2018

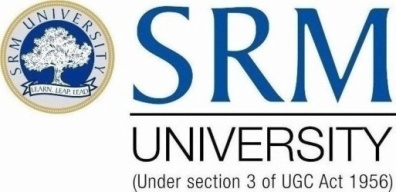
##### SECURE THE DEVICE - IOT PROJECT REPORT

###### ***Report Submitted by***

##### M ARUNA SUKRUTHA – RA1511008010160

##### N AVINASH – RA1511008010086

##### PRIYANSHI GARG – RA1511008010224



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

*DEPARTMENT OF INFORMATION TECHNOLOGY*  April 2018

**BONAFIDE CERTIFICATE**

Certified to be the bonafide record of work done by Aruna Sukrutha ,Avinash , Priyanshi Garg of Information Technology, Btech Degree course in the IOT project

In SRM Institute of science and technology, Kattankulathur during the academic year 2018-2019.

Date :12-04-2018. Lab Incharge

Head of the Department

Submitted for project held in April,2018 in SRM Institute of Science and technology, Kattankulathur.

# INDEX

# 

|  |  |
| --- | --- |
| Content | Page number |
| Abstract | 4 |
| Introduction | 5 |
| Components and description | 6 |
| Sample code | 9 |
| Result | 24 |
| Conclusion | 25 |

**ABSTRACT**

The entire thought of the project "SECURE THE THINGS" is to give security to the things. Now and again we may confront trouble in keeping things securely. It may be either stolen or overlooked. It is very well may be in any way similar to imperative archives, mobiles and so on. Imagine a scenario where we lose a portable and it should be followed subsequent to turning off. We can have sensors to follow it. So in this task the concept of IoT is utilized to verify the item. Here utilizing the IoT idea, there will be two sensors associated with the item which is to be verified. The unique finger impression sensor will know your touch id. In the event that any other individual endeavors to access or take your article, it will advise to you through SMS on telephone.At that point we can come to realize that our thing has been stolen.

**INTRODUCTION**

In most developing countries, provision of basic preventive, promotive and curative services is a major concern of the Government. With developing populace and headway in the restorative innovation and expanding desire for the general population particularly for quality remedial consideration, it has now turned out to be basic to give quality human services administrations through the established institutions. In open Sector 15,393 allopathic medical clinics (Health Information of India 2003) are working.. In the rural areas, the secondary level care is being provided through 3222 CHCs (Bulletin on Rural Health Statistics in India 2005) with 30 beds each with specialist services of physicians, pediatricians, O & G specialists, and surgeons being made available.

## Be that as it may, these administrations have not been effective in picking up the confidence and certainty of the general population in view of absence of authorities, offices and responsibility, along side the scarcity of assets and non-involvement of the community. Hospital Management System is a simple yet effective management structure. This framework represents the medical clinics to deal with the issues of the emergency clinic.HMS is free to prescribe, generate and use the functioning and maintaining the quality of services.

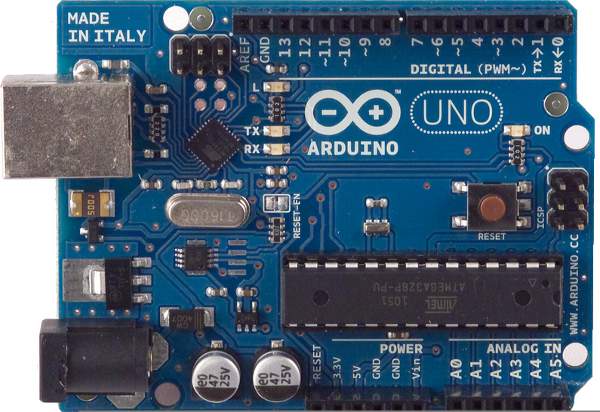
**RELATED WORK**

## 

## **COMPONENTS USED FOR PROJECT**

Arduino Uno board is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.

**Arduino Uno** is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worring too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.



R305 is a fingerprint sensor module with TTL UART interface for direct connections to microcontroller UART or to PC through MAX232 / USB-Serial adapter. 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. The FP module can directly interface with 3v3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with PC serial port.

Optical biometric fingerprint reader with great features and can be embedded into a variety of end products, such as: access control, attendance, safety deposit box, car door locks



GSM/GPRS Modem-RS232 is built with Dual Band GSM/GPRS engine- SIM900A, works on frequencies 900/ 1800 MHz. The Modem is coming with RS232 interface, which allows you connect PC as well as microcontroller with RS232 Chip(MAX232). The baud rate is configurable from 9600-115200 through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable you to connect with internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer application in M2M interface. The onboard Regulated Power supply allows you to connect wide range unregulated power supply . Using this modem,you can make audio calls, SMS, Read SMS, attend the incoming calls and internet ect through simple AT commands.

GSM modules are manufactured by different companies. They all have different input power supply specs. You need to double check your GSM modules power requirements. In this tutorial, our gsm module requires a 12 volts input. So we feed it using a 12V,1A DC power supply. I have seen gsm modules which require 15 volts and some other which needs only 5 volts. They differ with manufacturers. If you are having a 5V module, you can power it directly from Arduino’s 5V out.



5V Battery, Jumper wires, USB cable to interface between arduino board and the device.Arduino software is installed on laptopo to execute the code.



## **Sample code**

**Enroll,ino :-**

#include <Adafruit\_Fingerprint.h>

#include <SoftwareSerial.h>

uint8\_t getFingerprintEnroll(int id);

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

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

SoftwareSerial mySerial(2, 3);

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

void setup()

{

Serial.begin(9600);

Serial.println("fingertest");

// 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);

}

}

void loop() // run over and over again

{

Serial.println("Type in the ID # you want to save this finger as...");

int id = 0;

while (true) {

while (! Serial.available());

char c = Serial.read();

if (! isdigit(c)) break;

id \*= 10;

id += c - '0';

}

Serial.print("Enrolling ID #");

Serial.println(id);

while (! getFingerprintEnroll(id) );

}

uint8\_t getFingerprintEnroll(int id) {

int p = -1;

Serial.println("Waiting for valid finger to enroll");

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;

}

}

// 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\_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();

}

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;

}

}

// 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!

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;

}

}

**Check\_match.ino** :-

#include <Adafruit\_Fingerprint.h>

#include <SoftwareSerial.h>

int getFingerprintIDez();

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

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

SoftwareSerial mySerial(2, 3);

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

void setup()

{

Serial.begin(9600);

Serial.println("fingertest");

// 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);

}

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.

}

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\_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;

}

// found a match!

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

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

}

// 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!

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

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

return finger.fingerID;

}

**Message.ino:-**

void setup()

    {

    Serial.begin(9600);

    }

    void loop()

    {

    delay(1200);

    Serial.print("AT");

    delay(1200);

    bool bOK = false;

    while (Serial.available() > 0)

    {

    char inChar = (char)Serial.read();

    bOK = true;

    }

    if(bOK)

    {

    index = 0;

    Serial.println();

    Serial.println("AT+CMGF=1"); // sets the SMS mode to text

    delay(100);

    delay(1200);

    bool bOK = false;

    while (Serial.available() > 0) {

    //Serial.write(Serial.read());

    char inChar = (char)Serial.read();

    bOK = true;

    }

    if(bOK)

    {

    Serial.println();

    Serial.print("AT+CMGS=""); // send the SMS number

    Serial.print("+917358274445");

    Serial.println(""");

    delay(500);

    Serial.print("A new post is created by Zain."); // SMS body

    delay(500);

    Serial.write(0x1A);

    Serial.write(0x0D);

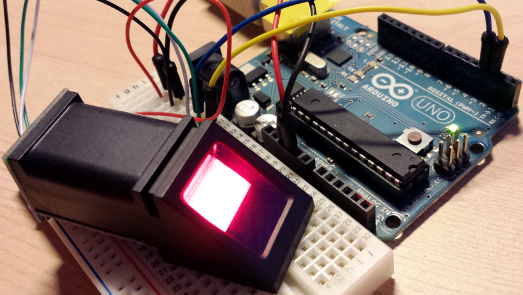
    Serial.write(0x0A);

  }

    }

    }

## **RESULT**





## **CONCLUSION**

We here by conclude that IOT project on secure the things is executed successfully and the result is shown to the respective faculty. The output is that a fingerprint can be enrolled. Checking whether fingerprint is enrolled in the database or not. If its enrolled then green led will blow and send a message to the registered mobile number that the login is successful. If its not registered then red led will blow and send the message to the given mobile number that Someone else is trying to access the device.