Project Report 3 **Citizen Safety Device**

****

**Group 37**

Yashvi Pipaliya (AU1841092)

Kesha Bagadia (AU1841011)

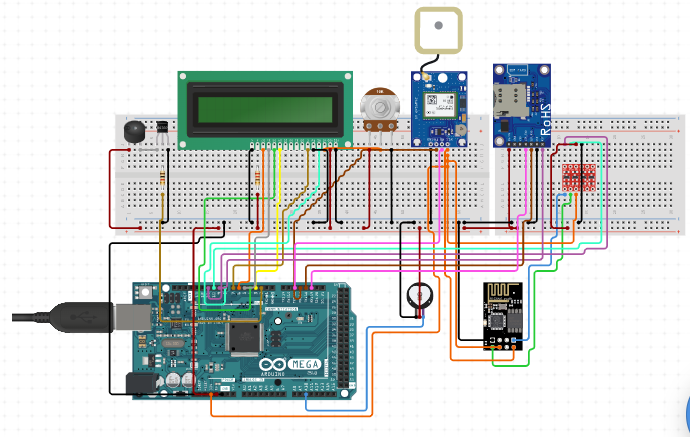
Yashvi Gandhi (AU1841033)

Manal Shah (AU1841026)



**CHAPTER : 4**

Circuit Diagram



● **Fingerprint Scanner** - TTL (GT-511C3) will read and identify fingerprints using an on-board optical sensor and CPU. When the system is activated, it will take constant input from the user’s finger pressing on it.

● **Heart Rate Pulse Sensor** will measure heart bpm. This again will be a constant input taken from the user when the system is active.

● **LCD 16x2** will be used to display intermediate messages and help with set up and status. ● **Buzzer** will ring and raise alarm as an output when danger is detected (heart bpm exceeds threshold or fingerprint input is not detected.)

● **Ublox NEO-6M GPS Module** retrieves the user' location and provides the current time and date. Output of this module will be extracted when danger is detected and alert is to be sent to the user’s contacts.

● **AE GSM MODEM SIM900A**, on insertion of a valid SIM card, can be used for voice calls, sending text messages and accessing the internet. When the system is active and there is any discrepancy in the inputs, like fingerprint is not detected or the heart bpm exceeds

1



the threshold, it will be used to communicate alert messages and location coordinates to the user’s trusted contacts. This will be the output of the system.

**CHAPTER : 5**

Comparison

We have selected to work on the Arduino microcontroller for our project. Here are some of the features of Arduino UNO and its comparison with other microcontrollers like Raspberry Pi, BeagleBone, Intel's Galileo, Intel's Edison.

|  | **Arduino** | **Raspberry-pi** | **BeagleBone Intel’s Galileo** | **Intel’s Edison** |
| --- | --- | --- | --- | --- |
| **Strengths** | Easy to connect  with some LED’s, sensors, motors into the board directly | All the Storage is provided  from a SD card. You can  connect this to your network with an  Ethernet Cable. | Similar to a  Same size and  Raspberry Pi but  shape as an SD  more powerful,  card and  Based on the TI  containing a  Sitara AM335x,  dual-core Intel  an application  Quark x86 CPU at  processor SoC  400 MHz  containing an  communicating via  ARM Cortex-A8  Bluetooth and  core  Wi-Fi | The Intel Edison module is a SoC that includes an Intel Atom  500MHz  dual-core,  dual-threaded CPU and an Intel Quark 100MHz microcontroller. |
|  | The Arduino is a  microcontroller. The arduino can be  programmed, but can’t run an  operating system | Raspberry Pi  are computers. Those devices can run an  operating  system alone | BeagleBone are  Intel's Galileo is a  computers.  microcontroller. It  Those devices  can be  can run an  programmed i, but  operating system  can’t run an  alone  operating system | Intel's Editson is a  microcontroller. It can be  programmed i, but can’t run an operating  system |

Good combination of

Digital

and Analog pins.

Perfect

availability of general-purpos e I/O pins

Abundant Digital and Analog pins are available .

Good combination of Digital and

Analog pins.

Supports

external storage Via. MicroSD card.

2



|  | Very versatile and extendable  due to the  availability of  various  external modules for different  tasks. | Supports  gigabit  ethernet,  WiFi,  and Bluetooth | Supports gigabit  Supports external  ethernet,  storage  WiFi, and  via. MicroSD card.  Bluetooth | Supports all WiFi protocols  and Bluetooth |
| --- | --- | --- | --- | --- |
|  | Available in  portable size | Video and  Audio output  ports are  available | Enough memory  Supports Gigabit,  storage(4GB) is  Bluetooth  available  and Ethernet | Power adapter or USB port both can be used |

Light weight and comparatively cheapest

DDR3 RAM of 512 KB

Smallest in size

|  | **Arduino** | **Raspberry-pi** | **BeagleBone Intel’s Galileo** | **Intel’s Edison** |
| --- | --- | --- | --- | --- |
| **Weaknesses** | Very low clock  frequency (16  MHz) | Requires  standard 5V  power supply. Highest clock frequency (1.2 GHz) | Does not support  Comparatively  external  slower clock  memory  frequency (400  (such as MicroSD  MHz) and the  cards).  product is not  longer supported  by Intel | Comparatively slower clock  frequency (500 MHz) and the  product is not  longer  supported by  Intel  . |

Very low RAM(2KB).

Huge size Huge size Huge size and weight and not

easily portable

Not cost effective

Lowest storage (32KB)

No port for Analog pins.

Audio and video outputs

are handled by a microHDMI connection.

.

A compatible WiFi chip is must to use WIFi functionality

High Power consumption

3



|  | Very tedious and tiring connections | Supports only one WiFi  protocol.  . | Lack of reference  Does not support  materials  audio and video  available  output | Does not  provide an  Ethernet port. |
| --- | --- | --- | --- | --- |
|  | Slightly high  power  consumption | Storage is not available,  needs  a MicroSD card to function. | Quite expensive | It is  complicated  and difficult  to use. |

**General characteristics**

|  | **Arduino** | **Raspberry-pi BeagleBone Intel’s Galileo** | **Intel’s Edison** |
| --- | --- | --- | --- |
| Price | $29.95 | $35 $89 $50 | $79.90 |
| Processor | ATMega  328 | ARM11 ARM  Intel Quark  Cortex-A8 | Intel Atom  CPU |
| Size | 2.95"x2.10" | 3.37"x2.125" 3.4"x2.1" 4.8 x 2.8 | 25 x 4mm |
| Clock  Speed | 16MHz | 700MHz 700MHz 500 MHz, 100 MHz | 400 MHz |
| Flash | 32KB | SD Card 4GB(microSD) 4 GB eMMC | 8 Mb |
| RAM  Input  Voltage | 2KB  7-12v | 256MB 256MB 256MB 5v 5v 3.3 to 4.5 V | 1GB  3V3, 5V |
| Analog  Input  USB | 6 10-bit  N/A | N/A 7 12-bit 12-bit  USB 2.0 USB 2.0 USB 2.0 | 12-bit  USB 2.0 |

Min Power 42mA (.3W) 700mA 170mA 450mA 100mA 4



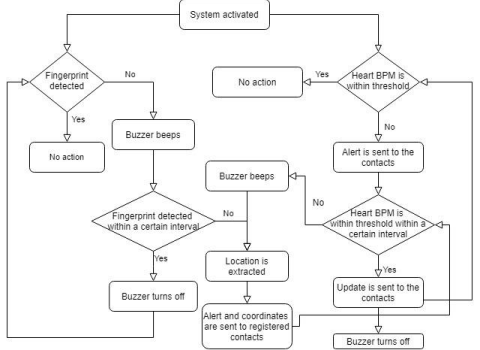
|  |  | (3.5W) (.85W) (2.24W) | (0.5W) |
| --- | --- | --- | --- |
| EEPROM | 1KB | N/A N/A 11KB | N/A |
| PWM | 6 | N/A 8 6 | 4 |
| Dev IDE | Arduino Tool | IDLE,  Python,  Arduino IDE  Scratch,  Scratch,  Squeak/Linux  Squeak,  Cloud9/Linux | Arduino,  Python,  Node.js |

5



**CHAPTER : 6**

Program Flow chart

6



**CHAPTER : 7**

Sensors

**Fingerprint Scanner - TTL (GT-511C3)**

**1. Details of operating principle with diagrams:**

This device is one chip module with:

● Fingerprint algorithm

● Optical sensor

**2. Physical dimensions:**

7

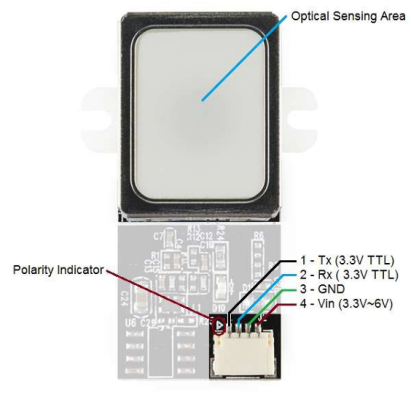


**3. Details of power ratings:**

Operating Voltage: 3.3V ~ 6Vdc

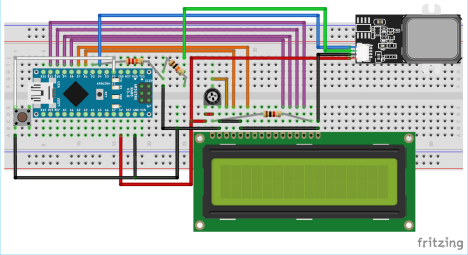
Operating Current: < 130mA

**4. Pin diagram:**

****



**5. Type of interface with Arduino/Raspberry Pi:**

****

**6. Details of interfacing diagram:**

The GT511C3 FPS has two power pins which can be powered by +5V pin of Arduino and two communication pins Rx and Tx which can be connected to any digital pin of Arduino for serial communication. Additionally, we have also added a push button and a LCD to display the sensor status.

**7. Code to communicate with the sensors**

/\* Connect Tx of FPS to Arduino Pin D4 and Rx of FPS to D5\*/ #include "FPS\_GT511C3.h"

#include "SoftwareSerial.h" //Software serial library

#include <LiquidCrystal.h> //Library for LCD

9



FPS\_GT511C3 fps(4, 5); //FPS connected to D4 and D5

const int rs = 6, en = 7, d4 = 8, d5 = 9, d6 = 10, d7 = 11; //Mention the pin number for LCD connection

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);//Initialize LCD method

void setup()

{

Serial.begin(9600);

lcd.begin(16, 2); //Initialise 16\*2 LCD

lcd.print("GT511C3 FPS"); //Intro Message line 1

lcd.setCursor(0, 1);

lcd.print("with Arduino"); //Intro Message line 2

delay(2000);

lcd.clear();

fps.Open(); //send serial command to initialize fps fps.SetLED(true); //turn on LED so fps can see fingerprint pinMode(2,INPUT\_PULLUP); //Connect to internal pull up resistor as input pin

}

void loop()

{

if (fps.IsPressFinger())

{

fps.CaptureFinger(false);

int id = fps.Identify1\_N();

lcd.clear();

lcd.print("Detected");

if (id==200) lcd.print("Unknown"); //If not recognised

lcd.print(id);

delay(1000);

}

else

{

lcd.clear(); lcd.print("Hi!....."); //Display hi when ready to scan }

}

10



**8. Photos of working hardware:**

****

**Heart Rate Pulse Sensor**

**1. Details of operating principle with diagrams**

The front of the sensor is the side with the heart logo. This is where you place your finger. On the front side you will see a small round hole, from where the Kingbright’s reverse mounted green LED shines.



Just below the LED is a small ambient light photo sensor – APDS-9008 from Avago, similar to that used in cell phones, tablets and laptops, to adjust the screen brightness in different light conditions.

11



On the back of the module you will find the rest of the components including a microchip’s MCP6001 Op-Amp and a bunch of resistors and capacitors that make up the R/C filter network. There is also a reverse protection diode to prevent damage if the power leads are accidentally reversed.



**2. Physical dimensions**

L x W (PCB) 15.8mm (0.625″)

Lead Length 20cm (7.8″)

**3. Details of power ratings**

VCC: 3.0 – 5.5V

IMax (Maximum Current Draw): < 4mA

VOut (Output Voltage Range): 0.3V to Vcc

12



**4. Pin diagram**

****

**5. Type of interface with Arduino**

****

13



**6. Details of interfacing diagram**

The module can be powered from 3.3 or 5V. The positive voltage connects to ‘+’ and ground connects to ‘-‘. The 3rd ‘S’ wire is the analog signal output from the sensor and this will connect to the A0 analog input of an Arduino.

**7. Code to communicate with the sensors**

int const PULSE\_SENSOR\_PIN = 0; // 'S' Signal pin connected to A0 int Signal; // Store incoming ADC data. Value can range from 0-1024 int Threshold = 550; // Determine which Signal to "count as a beat" and which to ignore.

void setup() {

pinMode(LED\_BUILTIN,OUTPUT); // Built-in LED will blink to your heartbeat Serial.begin(9600); // Set comm speed for serial plotter window }

void loop() {

Signal = analogRead(PULSE\_SENSOR\_PIN); // Read the sensor value

Serial.println(Signal); // Send the signal value to serial plotter

if(Signal > Threshold){ // If the signal is above threshold, turn on the LED

digitalWrite(LED\_BUILTIN,HIGH);

}

else {

digitalWrite(LED\_BUILTIN,LOW); // Else turn off the LED

}

delay(10);

}

14



8. **Photos of working hardware**

****

**CHAPTER : 8**

Actuators

**Buzzer**

1. Details of operating principle with diagrams



15





At the heart of all piezo-type buzzers is the piezoelectric element. The piezoelectric element is composed of a piezoelectric ceramic and a metal plate held together with adhesive. Both sides of the piezoelectric ceramic plate contain an electrode for electrical conduction. Piezo materials exhibit a specific phenomenon known as the piezoelectric effect and the reverse piezoelectric effect. Exposure to mechanical strain will cause the material to develop an electric field, and vice versa.



When an alternating voltage is applied to the piezoceramic element, the element extends and shrinks diametrically. This characteristic of piezoelectric material is utilized to make the ceramic plate vibrate rapidly to generate sound waves.

16



**2. Physical dimensions**

****

**3. Details of power ratings**

Rated Voltage: 6V DC

Operating Voltage: 4 to 8V DC

Rated Current\* : ≤30mA

Sound Output at 10cm\* : ≥85dB

Resonant Frequency: 2300 ±300Hz

Tone: Continuous

**4. Pin diagram**

****17



**5. Type of interface with Arduino/Raspberry Pi**

**6. Details of interfacing diagram**

Here, the cathode is connected to the ground while the anode is connected in series with the resistor to the pin 9 of the arduino.

**7. Code to communicate with the sensors**

const int buzzer = 9; //buzzer to arduino pin 9

void setup(){

pinMode(buzzer, OUTPUT); // Set buzzer - pin 9 as an output }

void loop(){

tone(buzzer, 1000); // Send 1KHz sound signal...

delay(1000); // ...for 1 sec

noTone(buzzer); // Stop sound...

delay(1000); // ...for 1sec

}

18



**8. Include photos of working hardware.**

****

Displays

**LCD Display 16X2**

**1. Details of operating principle with diagrams**

In LCD 16×2, the term LCD stands for Liquid Crystal Display that uses a plane panel display technology, used in screens of computer monitors & TVs, smartphones, tablets, mobile devices, etc. Both the displays like LCD & CRTs look the same but their operation is different. Instead of electron diffraction at a glass display, a liquid crystal display has a backlight that provides light to each pixel that is arranged in a rectangular network.

19





Every pixel includes a blue, red, green sub-pixel that can be switched ON/OFF. Once all these pixels are deactivated, then it will appear black and when all the sub-pixels are activated then it will appear white. By changing the levels of each light, different color combinations are achievable.

**2. Physical dimensions**

Size: 85.0 x 29.5 x 13.5 mm

Viewing area: 64.5 x 16.4 mm

Dot size: 0.56 x 0.61 mm

Character size: 3.00 x 5.23 mm

Weight: 35 g

3. **Details of power ratings**

Operating Voltage: 4.7V to 5.3V

Operating Current: 1mA (without backlight)

20



4. **Pin diagram**

****

21



**5. Type of interface with Arduino**

**6. Details of interfacing diagram**

Before interfacing the LCD screen to the Arduino board, a pin header strip need to be solder to pin-14 or 16 of the LCD. We can notice this in the following circuit diagram. The following pins need to connect to wire the LCD to an Arduino board.

● RS pin of LCD to digital pin-12

● Enable pin is connected to digital pin-11

● D4 pin is connected to digital pin -5

● D5 pin is connected to digital pin- 4

22



● D6 pin is connected to digital pin-3

● D7 pin is connected to digital pin-2

● Read/Write pin is connected to GND

● VSS pin is connected to the GND terminal

● VCC pin is connected to 5V

● A 220-ohm resistor is connected from LED+ to 5V

● LED is connected to the GND terminal

**7. Code to communicate with the sensors’**

#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 6, d5 = 5, d6 = 4, d7 = 3; LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {

lcd.begin(16, 2); // set up the LCD's number of columns and rows: lcd.print("Hello World!"); // Print a text to the LCD.

}

void loop() {

// set the cursor to column 0, line 1

// (note: line 1 is the second row, since counting begins with 0): lcd.setCursor(0, 1);

// print the number of seconds since reset:

lcd.print(millis() / 1000);

}

23



**8. Photos of working hardware**

****24



**Appendix A**

Datasheets

**Fingerprint sensor** 25



26



27



28



**Pulse Sensor**

****

29



30



31



32



**Buzzer**

****33



34



35