**CHAPTER1**

**INTRODUCTION**

* 1. **INTRODUCTION**

India is the biggest democratic country in the world which has taken giant strides in the last few decades and we are now the world’s sixth-largest economy. But in between this, we cannot forget everyday issues that still continue to pose hurdles in the progress of the nation. Though our country holds the title as democratic, there’s still a **communication gap** between the people and the government. This is the gap that is under covering and ignoring the sufferings of the people living in harsh conditions in the slums. Technologies are meant not only to learn but to implement and utilize it at the most. With the same aim, we have built a smart helping booth named as S2D (Slums to Democratic) that could be implemented in the slums and serve its need in this era of nation’s development. Now let us consider a major problem – food scarcity. For this we have built the booth which consists of fingerprint sensor. When comes to the real time implementation, it would work with big data analysis of the aadhaar card basis. For demonstration, we have enrolled few finger ID’s through the arduino software. Whenever there is a need of food by the slum resident, he/she need to approach the booth and scan his/her fingerprint. Once the fingerprint is detected, the ID would traverse through the thingspeak channel which would be present at the nearby NGO. The alarm would ring so as caution the NGO person. Thus, once the NGO person identifies the fingerprint ID in the channel, the immediate action would take place.

**Advantages:**

Interpret technology as an implementation tool not only as a learning means! Thus, we being engineering peers believe in serving the nation by coming up with the real time applications by updating ourselves with the technology support. As now we brought this S2D booth in light so as to remove the malaise of sufferings. In this regard our smart booth would generate the following means:

1. Through this booth we can generate great support from the government and NGO’s to the people living in the slums and by this poverty levels would be decreased.
2. Food that is being wasted adequately in the cities at different circumstances like functions, restaurants, can be communicated to the NGO’s so as to supply to the need with the help of S2D.
3. We can create employment opportunities to the youth present in the slums who can be able to transport the food from the NGO’s to the slums and can organize the marketing things.
4. As the S2D booth requires the Wi-Fi router, we can help in digitalizing slums at an extent.
5. The bonds with the IT sector must be maintained so that they could contribute in donating the electronic goods which in turn help in inculcating the skills in the people living in slums. Thus, literacy levels would be increased.
6. The equipments which are used in the smart booth are affordable and can be flexible in implementing in slums.

Thus, one smart innovation can solve all the major problems that are becoming barrier for the country’s growth.

**Disadvantages:**

1. If fingerprint sensor used is not of good quality.
2. Sometimes there would arise the improper signal problems due to different reasons like bad weather.

**1.2 OVERVIEW OF PROJECT:**

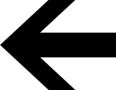
* In this era of the growth of technology, there must even lead to the growth of the nation. Thus, we need to utilize the technology at large extent.
* The basic idea of this project is generated from the telephone booths that were used in the olden days where there was no proper networking means.
* The NGO’s nearby can easily get to know about the people in need through the channel and take immediate action.
* Those who don’t have proper identity in slums will be recognized if their fingerprint is not detected and this would be passed to the govt. estimations.
* Easy to create awareness about different schemes and programs implemented by the govt. through this booth.

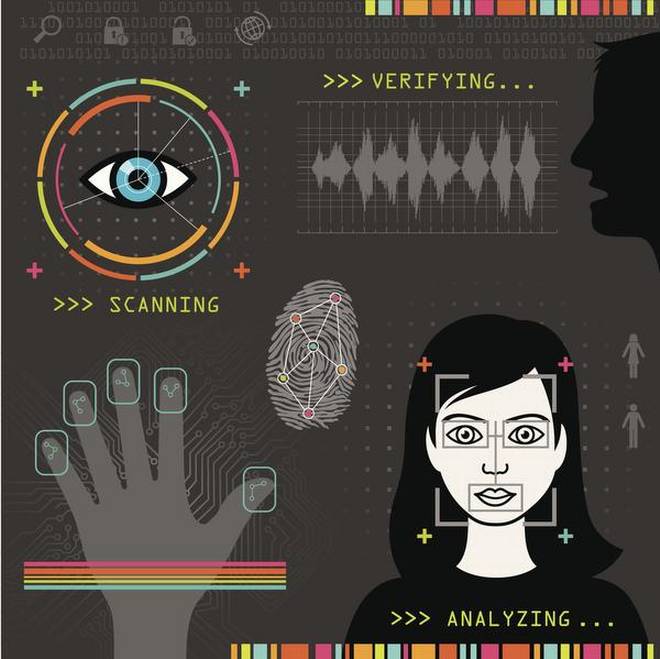
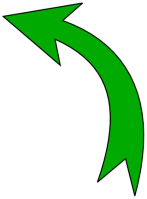
**1.3 OBJECTIVES:**

* The objective of this project is to implement a low cost, reliable, and scalable product that could stand as a support to the people in slums who are in need of food and even to remove the different problems becoming a drawback in the development of India.
* Building this communication gap between the people and government and even among the people to experience and implement the real strategies of democracy by making India a better place to live in is the **main objective of S2D** (Slums to Democratic).

**1.3** **REAL-TIME PROCESS OF THIS PROJECT IN A NUTSHELL**

  **NGO’s**

****

****

**CHAPTER 2**

**PROJECT DESCRIPTION**

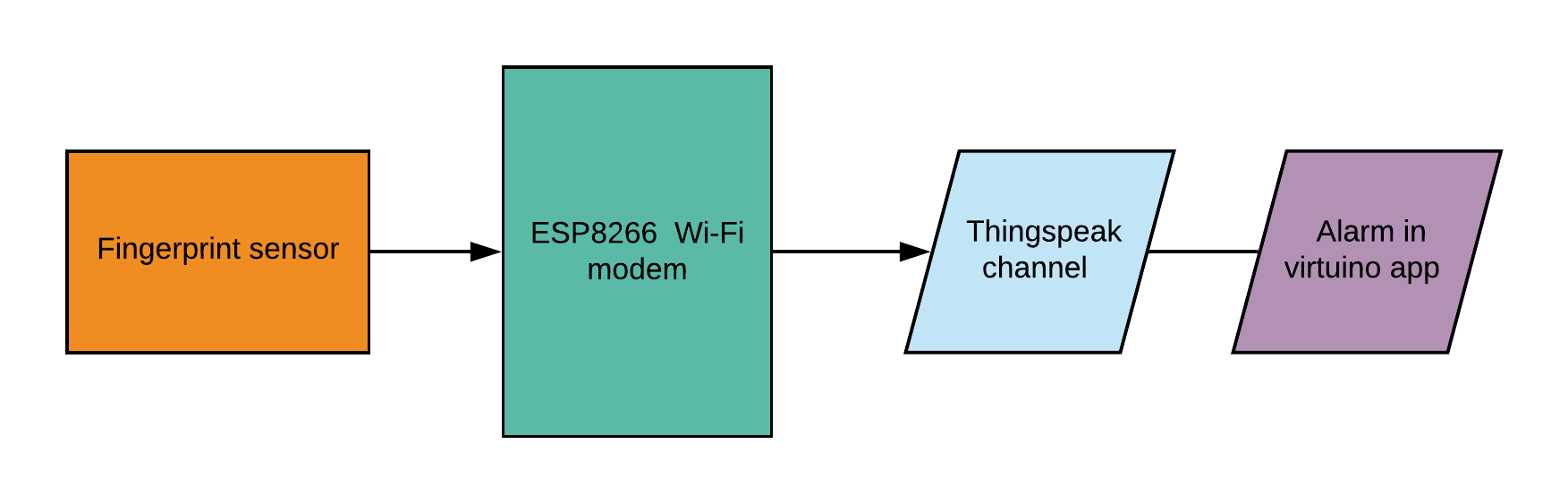
**2.1 BLOCK DIAGRAM OF THE PROJECT**

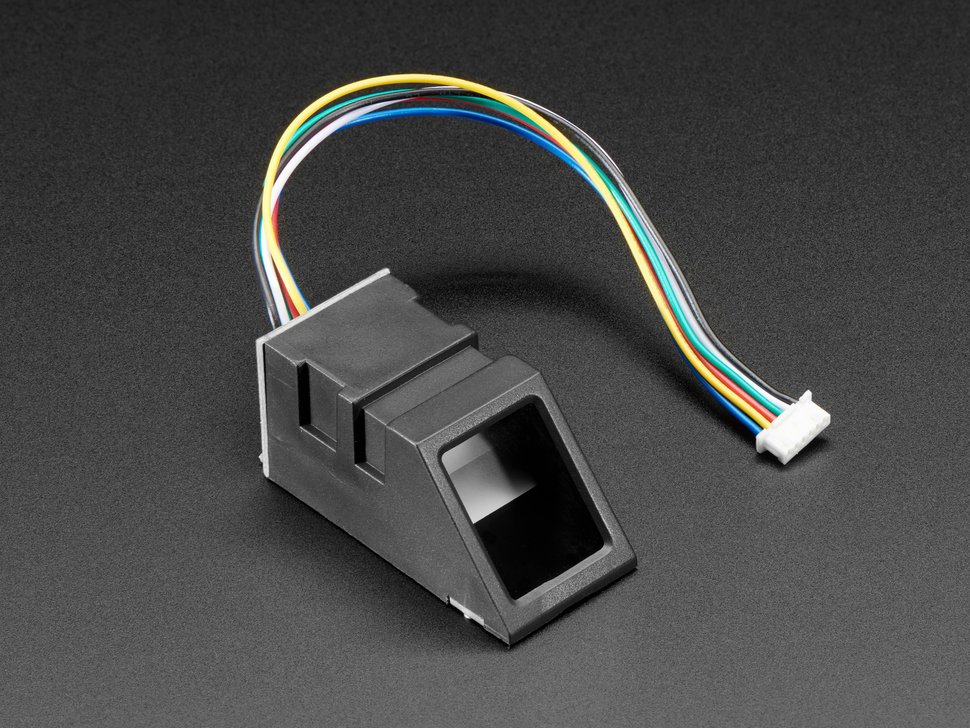
Fig.2.1 Block Diagram

As shown in the above schematic diagram it mainly consists of involvement of the above concepts.

Whenever a person who is in need of food approaches to this smart helping booth and scan his finger through fingerprint sensor. If the finger is detected and then once the network is set up, through the ESP8266 modem, the data of finger ID is sent to the thingspeak channel. From there in the thingspeak channel, the graph is mapped based on the data that is transferred into channel. Once the data is sent into the channel, a alarm rings through Virtuino app so as to intimate and caution the NGO person.

**2.3HARDWARE DESCRIPTION**

**2.3.1FINGERPRINT SENSOR:**



**DESCRIPTION**

Secure your project with biometrics - this all-in-one optical fingerprint sensor will make adding fingerprint detection and verification super simple. These modules are typically used in safes - there's a high powered DSP chip that does the image rendering, calculation, feature-finding and searching. Connect to any microcontroller or system with TTL serial, and send packets of data to take photos, detect prints, hash and search. You can also enroll new fingers directly - up to 162 finger prints can be stored in the onboard FLASH memory.  
  
We like this particular sensor because not only is it easy to use, it also comes with fairly straight-forward Windows software that makes testing the module simple - you can even enroll using the software and see an image of the fingerprint on your computer screen. But, of course, we wouldn't leave you a datasheet and a "good luck!" - we've written both a Arduino library and Circuit Python library, so that you can get running in under 10 minutes. The library can enroll and search so its perfect for any project.

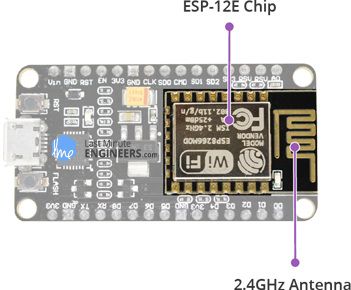
TECHNICAL DETAILS

* Supply voltage: 3.6 - 6.0VDC
* Operating current: 120mA max
* Peak current: 150mA max
* Fingerprint imaging time: <1.0 seconds
* Window area: 14mm x 18mm
* Signature file: 256 bytes
* Template file: 512 bytes
* Storage capacity: 162 templates
* Safety ratings (1-5 low to high safety)
* False Acceptance Rate: <0.001% (Security level 3)
* False Reject Rate: <1.0% (Security level 3)
* Interface: TTL Serial
* Baud rate: 9600, 19200, 28800, 38400, 57600 (default is 57600)
* Working temperature rating: -20C to +50C
* Working humidity: 40%-85% RH
* Full Dimensions: 56 x 20 x 21.5mm
* Exposed Dimensions (when placed in box): 21mm x 21mm x 21mm triangular
* Weight: 20 grams

## 2.3.2 ESP8266 Module

The development board equips the ESP-12E module containing ESP8266 chip having **Tensilica Xtensa® 32-bit LX106 RISC microprocessor** which operates at **80 to 160 MHz** adjustable clock frequency and supports **RTOS**.

* Tensilica Xtensa R 32-bit LX106
* 80 to 160 MHz clock Freq
* 128kB internal RAM
* 4MB external flash
* 802.11b/g/n Wi-Fi transcieve



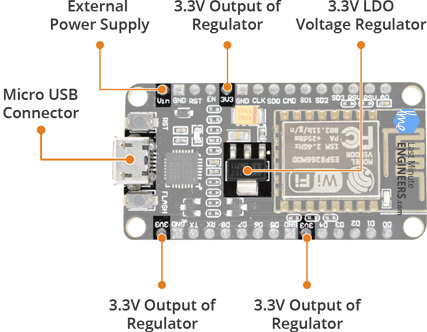
There’s also **128 KB RAM and 4MB of Flash memory** (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IOT devices nowadays.

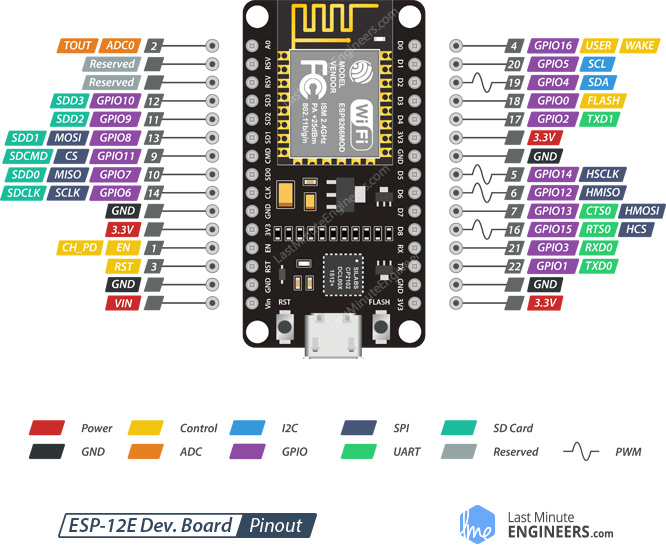
The ESP8266 Integrates **802.11b/g/n HT40 Wi-Fi transceiver**, so it can not only connect to a Wi-Fi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 Node MCU even more versatile

## Power Requirement

As the operating voltage range of ESP8266 is **3V to 3.6V**, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as **80mA during RF transmissions**. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

* Operating voltage:2.5V to 3.6V
* On-board 3.3V 600mA regulator
* 80mA operating current20 µA during Sleep Mode
* 20µA during sleep mode





### ESP8266-01 Features

* Low cost, compact and powerful Wi-Fi Module
* Power Supply: +3.3V only
* Current Consumption: 100mA
* I/O Voltage:  3.6V (max)
* I/O source current: 12mA (max)
* Built-in low power 32-bit MCU @ 80MHz
* 512kB Flash Memory
* Can be used as Station or Access Point or both combined
* Supports Deep sleep (<10uA)
* Supports serial communication hence compatible with many development platform like Arduino
* Can be programmed using Arduino IDE or AT-commands or Lua Script

**2.3.3 ARDUINO UNO**

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features: 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes. Stronger RESET circuit. Atmega 16U2 replace the 8U2. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)



Fig. 2.2 Arduino Uno

**Applications:**

* Xoscillo, an open-source [oscilloscope](https://en.wikipedia.org/wiki/Oscilloscope)
* [Arduinome](https://en.wikipedia.org/wiki/Arduinome), a [MIDI controller](https://en.wikipedia.org/wiki/MIDI_controller) device that mimics the [Monome](https://en.wikipedia.org/wiki/Monome" \o "Monome)
* [OBDuino](https://en.wikipedia.org/wiki/OBDuino), a [trip computer](https://en.wikipedia.org/wiki/Trip_computer) that uses the [on-board diagnostics](https://en.wikipedia.org/wiki/On-board_diagnostics) interface found in most modern cars
* Gameduino, an Arduino shield to create retro 2D video games
* ArduinoPhone, a do-it-yourself cellphone
* Water quality testing platform
* Automatic titration system based on Arduino and stepper motor
* Low cost data glove for virtual reality applications
* Impedance sensor system to detect bovine milk adulteration
* Homemade CNC using Arduino and DC motors with close loop control by Homofaciens
* DC motor control using Arduino and H-Bridge

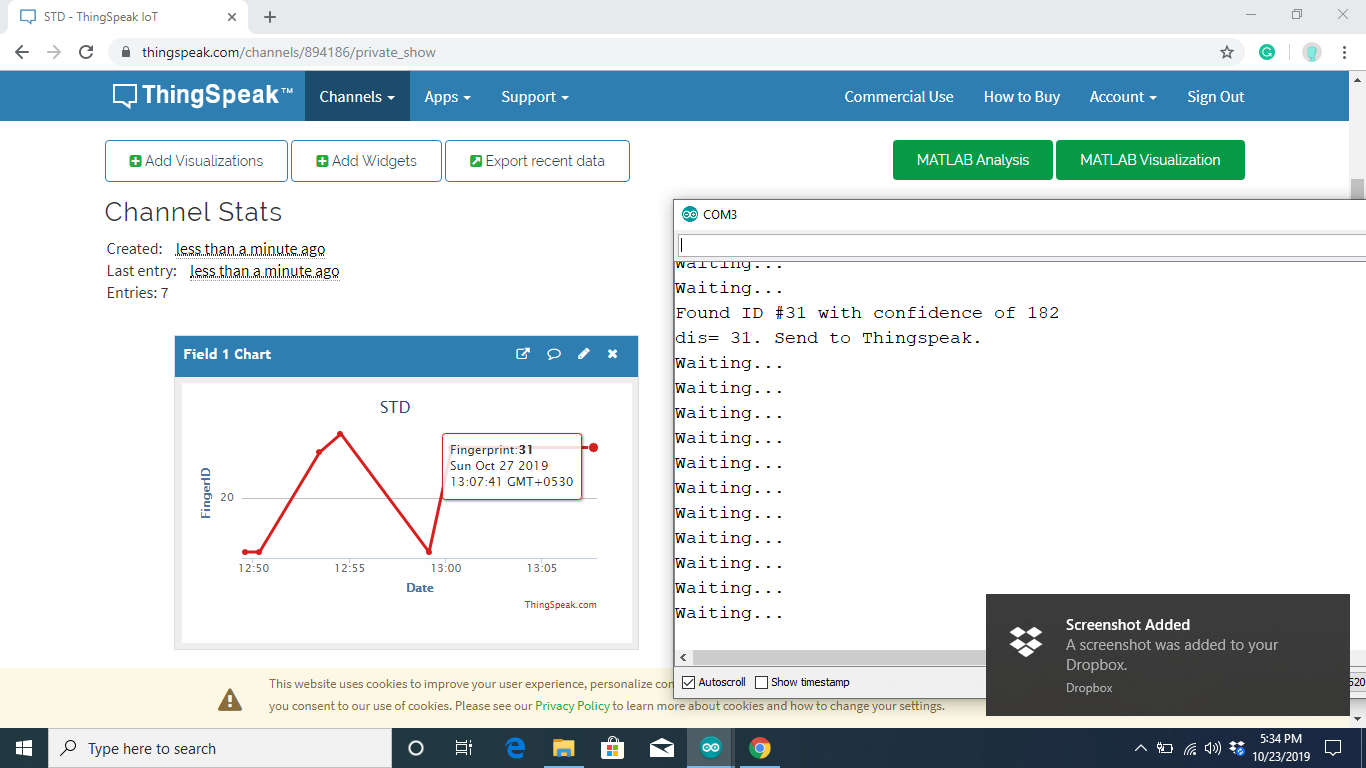
**2.3.4 THINGSPEAK CHANNEL**

ThingSpeak™ is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics. ThingSpeak has integrated support from the numerical computing software [MATLAB](https://en.wikipedia.org/wiki/MATLAB) from [Math Works](https://en.wikipedia.org/wiki/MathWorks),allowing ThingSpeak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks.

**ThingSpeak Key Features**

ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. Some of the key capabilities of ThingSpeak include the ability to:

* Easily configure devices to send data to ThingSpeak using popular IoT protocols.
* Visualize your sensor data in real-time.
* Aggregate data on-demand from third-party sources.
* Use the power of MATLAB to make sense of your IoT data.
* Run your IoT analytics automatically based on schedules or events.
* Prototype and build IoT systems without setting up servers or developing web software.
* Automatically act on your data and communicate using third-party services like Twilio® or Twitter®



**2.3.4 VIRTUINO APP:**

**Create visual interfaces with widgets like LEDs, buttons, switches, value displays, instruments, regulators etc.**



**2.4 SOFTWARE DESCRIPTION**

**The software used here is ARDUINO SOFTWARE:**

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

**Writing Sketches:**

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

**NB:**

Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the ino extension on save.

|  |  |  |
| --- | --- | --- |
| https://www.arduino.cc/en/uploads/Guide/play.png | | ***Verify*** Checks your code for errors compiling it. |
| https://www.arduino.cc/en/uploads/Guide/export.png | | ***Upload*** Compiles your code and uploads it to the configured board. See [uploading](https://www.arduino.cc/en/Guide/Environment#uploading) below for details.  **Note:** If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer" |
| https://www.arduino.cc/en/uploads/Guide/new.png | ***New*** Creates a new sketch. | |
| https://www.arduino.cc/en/uploads/Guide/open.png | ***Open***  Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.  **Note:** due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead. | |
| https://www.arduino.cc/en/uploads/Guide/save.png | ***Save*** Saves your sketch. | |
| https://www.arduino.cc/en/uploads/Guide/serial_monitor.png | ***Serial Monitor*** Opens the [serial monitor](https://www.arduino.cc/en/Guide/Environment#serialmonitor). | |

Additional commands are found within the five menus: File, Edit, Sketch, Tools,and help.

##### **Programming on arduino**

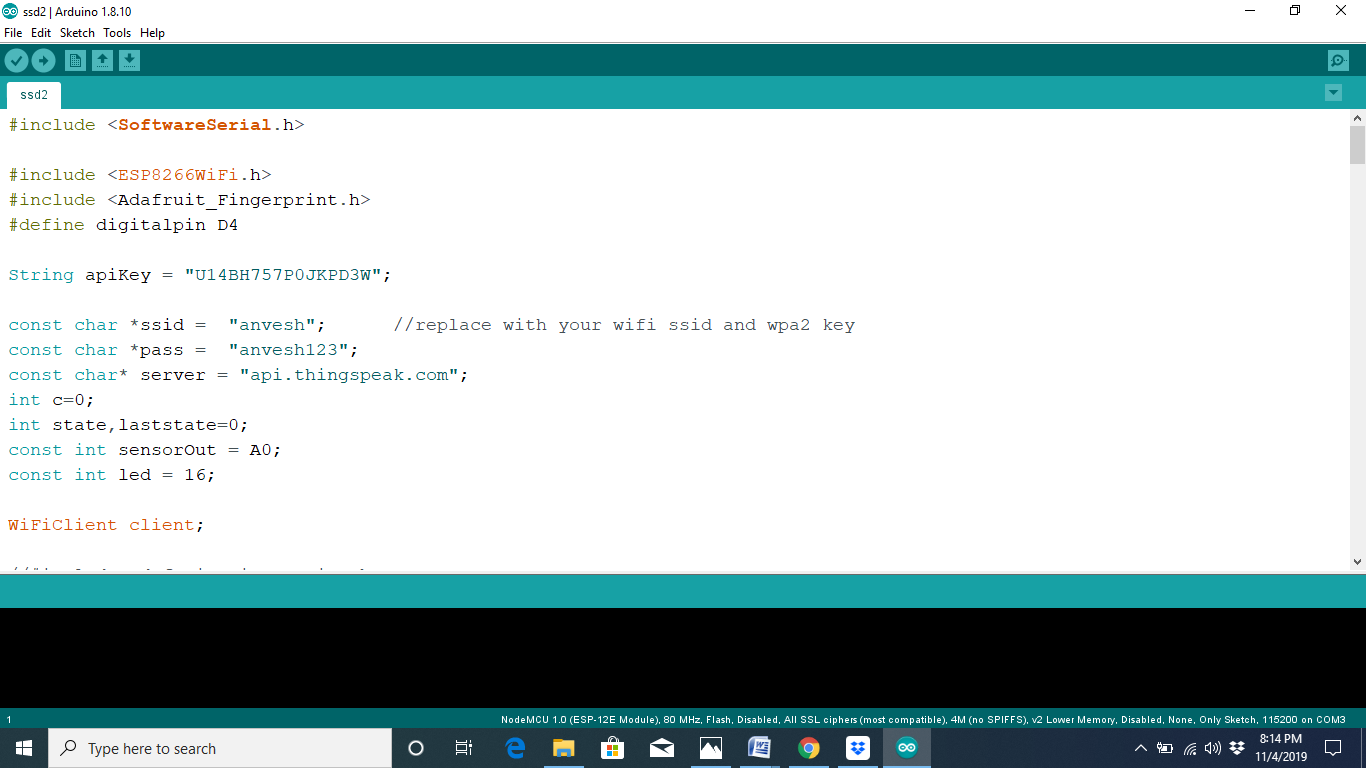
****

Fig.2.6 Software IDE

In order for the Arduino-Uno board to be able to interact with the application used in this project certain program (code) needs to be uploaded to the Arduino-Uno.

Arduino Company provides user friendly software which allows writing any code for any function wanted to be performed by the Arduino-Uno and upload it to the board. Refer to appendix A for the full source code of the Arduino-Uno board.

**CHAPTER 3**

**CIRCUIT DIAGRAM AND DESCRIPTION**

**3.1 Working**

Whenever a person who is in need of food approaches to this smart helping booth and scan his finger through fingerprint sensor. If the finger is detected and then once the network is set up, through the ESP8266 modem, the data of finger ID is sent to the thingspeak channel. From there in the thingspeak channel, the graph is mapped based on the data that is transferred into channel. Once the data is sent into the channel, a alarm rings through Virtuino app so as to intimate and caution the NGO person.

As the supply of volt doesn’t be sufficient to the fingerprint sensor through the ESP8266, connect to 5V of Arduino and the connections can be depicted as below in fig 3.1.

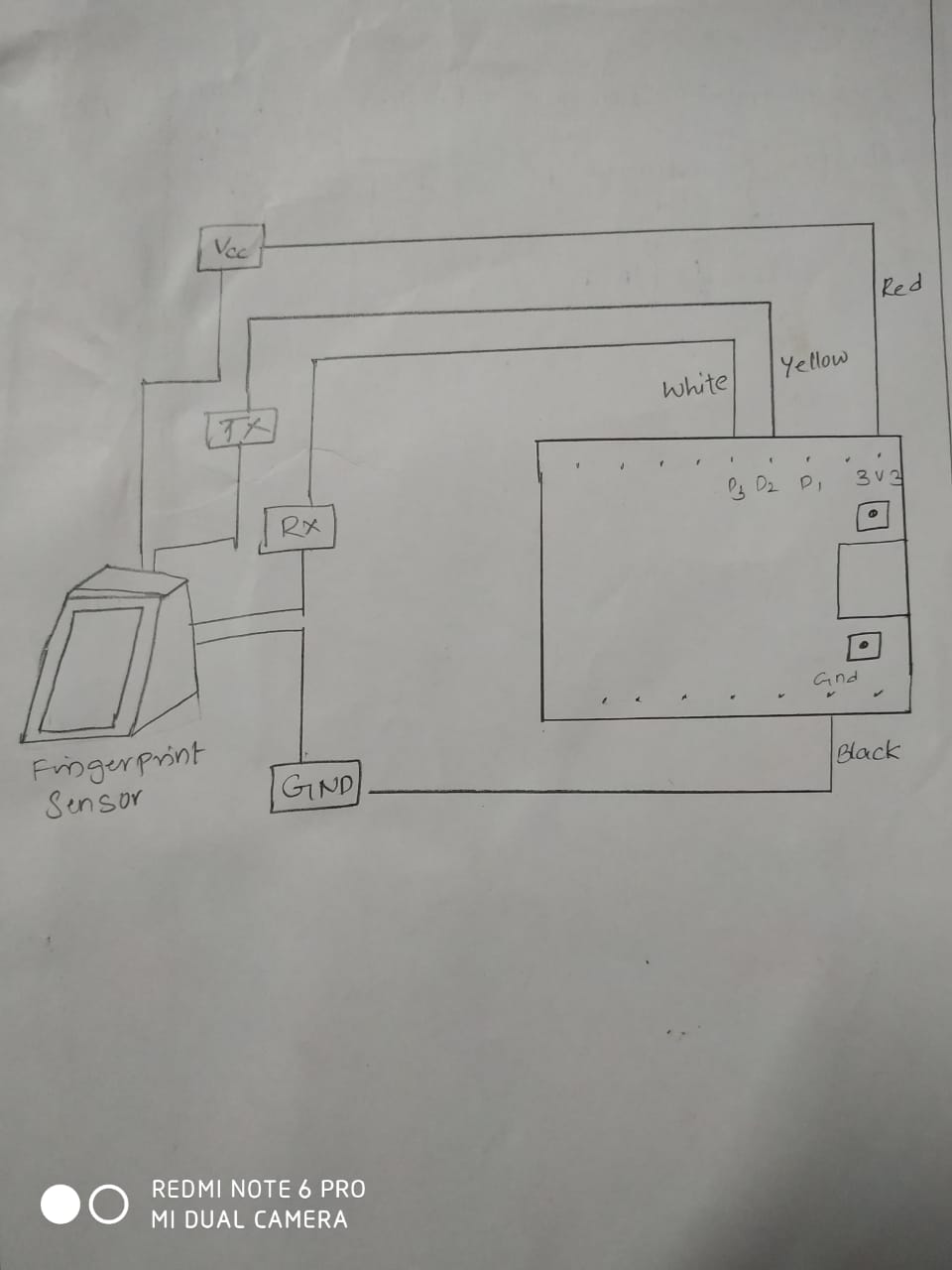


Fig 3.1 Schematic diagram

**3.2 RESULTS**

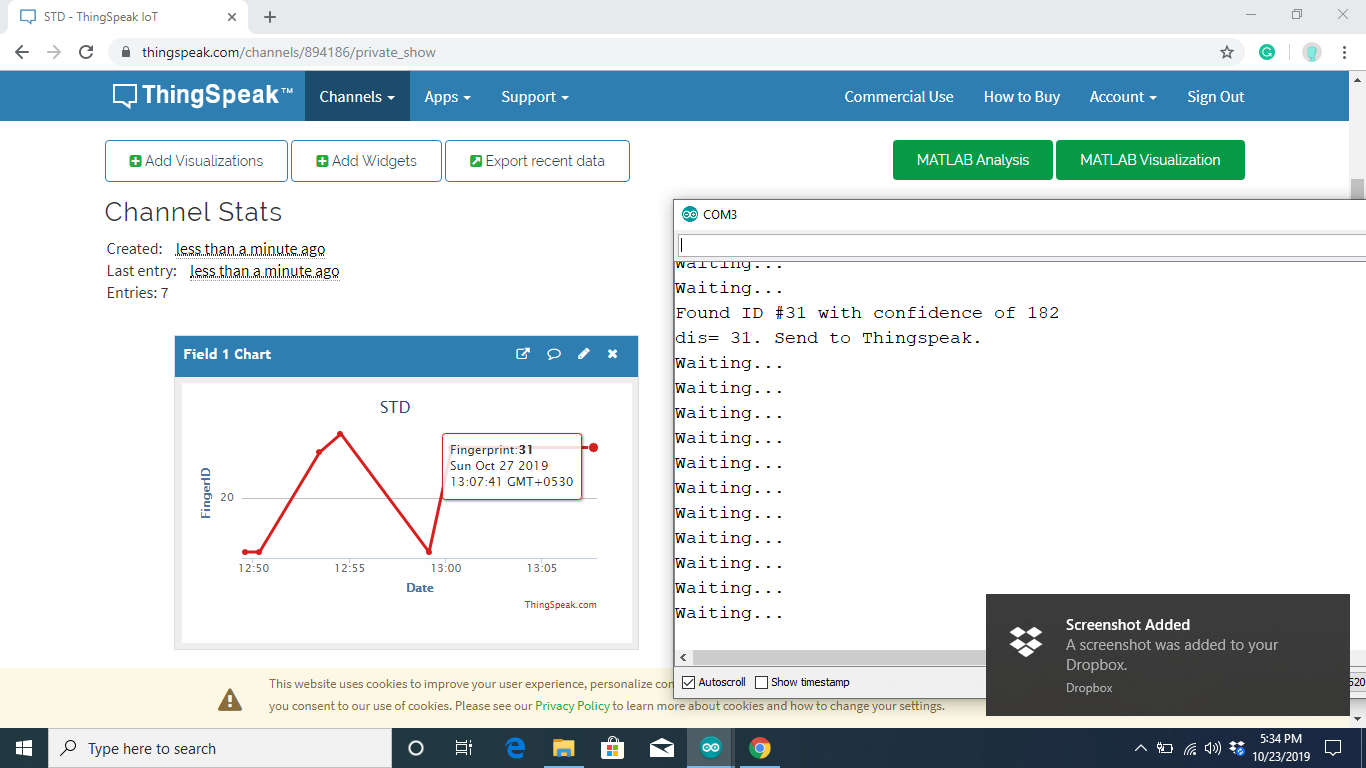
The experimental result is as shown in below.



Fig.3.2.1 Whole setup and the functioning of fingerprint sensor (ready to scan the finger)

Fig. 3.2.2 Functioning of ESP8266 and Arduino

Transfer of data through thingspeak channel is depicted in the fig. 3.2.3

Fig 3.2.3 Transfer of data through channel

**TO CAUTION THE NGO PERSON**

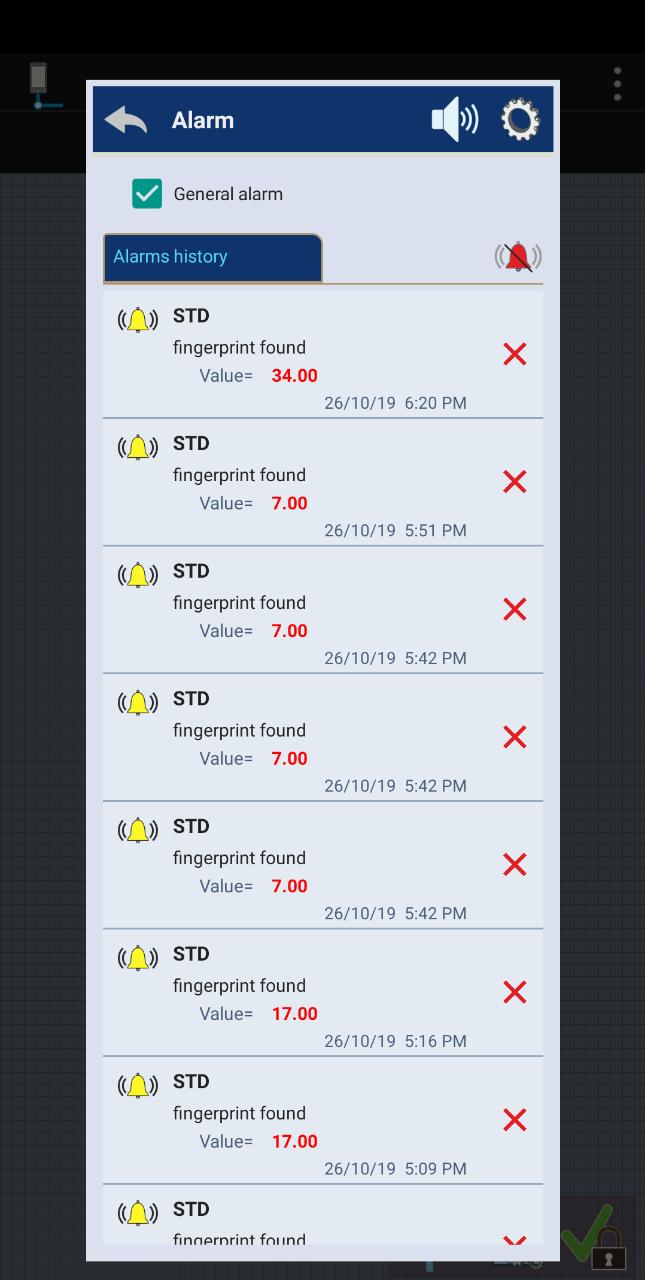


Fig 3.2.4.Alarm rings in Virtuino app and display Finger ID, date and time

**3.3 ADVANTAGES**

* Easy to handle and affordable to implement in slums.
* Proper communication established between people in need and people who are ready to help.
* Most major problem of food scarcity can be solved.
* Digitalize slums.
* Employment generated.
* Helps in survey and data analysis of poverty levels and support can be established.
* Contributes in nation’s development.

**3.4 DISADVANTAGES**

* Network issues due to weather conditions.
* Human errors.
* Reliability.

**CHAPTER 4**

**CONCLUSION**

**4.1 CONCLUSION**

At last, we would conclude that this would be even a great start-up that would really bring changes in the present circumstances as it is related to the nation’s development and aims to **transform millions of lives.**

India is being ruled by the people’s government and this power of unity is one of the greatest powers. Let’s utilize this power to the maximum extent to tribute our mother nation by our contribution. Everyone has the right to live and we have the right to contribute to transforming the lives by uplifting them by S2D (Slums to democratic).

**4.2 FUTURE ADVANCEMENTS:**

We believe in interpreting technology as an implementation tool not only the learning tool. Thus whatever we learn, we will definitely implement for good cause.

With the help of technologies like- Machine Learning, AI, Deep Learning and even IOT to the most extent so as to advance the model to overcome the security issues and even in the field of the health care system in slums.

When the person in the slums share the issue through the walkie talkie, the audio clip must be evaluated through AI and estimate the person’s need levels and to know the person whether he belongs to that region or not and even used further for issues like security.

Translation of the audio acknowledgment from the NGO’s to the regional language of slums. Build a model to speed up disease detection and can be implemented in the **health care system.**

Thus, different services can be provided through one smart helping booth.

**BIBLIOGRAPHY**

# [1]. [ESP8266 NodeMCU – Fingerprint Optical Sensor (JM-101)](https://www.geekering.com/?p=241) through Geekering site.

[2]. Thingspeak Virtuino data transfer, Journal of Networks - IoT, Vol. 3, No. 2.

[3]. Interfacing Fingerprint sensor – libraries to install – Adafruit –through Arduino software manage libraries.

**APPENDIX**

#include <SoftwareSerial.h>

#include <ESP8266WiFi.h>

#include <Adafruit\_Fingerprint.h>

#define digitalpin D4

String apiKey = "U14BH757P0JKPD3W";

const char \*ssid = "\*\*\*"; //replace with your wifi ssid and wpa2 key

const char \*pass = "\*\*\*\*\*";

const char\* server = "api.thingspeak.com";

int c=0;

int state,laststate=0;

const int sensorOut = A0;

const int led = 16;

WiFiClient client;

//#include Adafruit\_Fingerprint.h

// On LeonardoMicro or others with hardware serial, use those! #0 is green wire, #1 is white

//uncomment this line

//#define mySerial Serial1

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

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

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

//comment these two lines if using hardware serial

SoftwareSerial mySerial(D2, D3);

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

void setup()

{

Serial.begin(115200);

while (!Serial); //For YunLeoMicroZero...

delay(100);

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

//set the data rate for the sensor serial port

finger.begin(57600);

delay(5);

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...");

pinMode(D4, OUTPUT);

Serial.println("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println(" ");

Serial.println("WiFi connected");

}

void loop() // run over and over again

{

int x;

x=getFingerprintIDez();

delay(50); //don't ned to run this at full speed.

if (x==0)

{

Serial.println("Failed to read from sensor!");

}

else if (x>0 && x<127)

if (client.connect(server,80))

{

String postStr = apiKey;

postStr +="&field1=";

postStr += String(x);

postStr += "\r\n\r\n";

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

Serial.print("dis= ");

Serial.print(x);

Serial.println(". Send to Thingspeak.");

}

client.stop();

Serial.println("Waiting...");

// thingspeak needs minimum 15 sec delay between updates, i've set it to 30 seconds

delay(5000);

}

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!");

digitalWrite(digitalpin,HIGH);

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

Serial.println("Communication error");

digitalWrite(digitalpin,LOW);

return p;

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

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

digitalWrite(digitalpin,LOW);

return p;

} else {

Serial.println("Unknown error");

digitalWrite(digitalpin,LOW);

return p;

}

// found a match!

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

if(finger.fingerID>0 && finger.fingerID<127)

{

digitalWrite(D4,HIGH);

}

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

return finger.fingerID;

}

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

int getFingerprintIDez() {

uint8\_t p = finger.getImage();

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

p = finger.image2Tz();

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

p = finger.fingerFastSearch();

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

// found a match!

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

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

return finger.fingerID;

// Upload to thingspeak

}