Bangladesh University of Engineering and Technology

Department of Electrical and Electronic Engineering

EEE 416 (January 2022) **A2.7**

Microprocessor and Embedded Systems Laboratory

**Final Project Report**

Biometric Attendance System: TruID

# Evaluation Form:

|  |  |  |  |
| --- | --- | --- | --- |
| STEP | DESCRIPTION | MAX | SCORE |
| 1 | Report (Format, Reference) | 10 |  |
| 2 | Design Method and Complete Design (Hardware Implementation) | 15 |  |
| 3 | Video Demonstration | 10 |  |
| 4 | Novelty of Design | 15 |  |
| 5 | Project Management and Cost Analysis | 10 |  |
| 6 | Considerations to Public Health and Safety, Environment and Cultural and Societal Needs | 10 |  |
| 7 | Assessment of Societal, Health, Safety, Legal and Cultural issues relevant to the solution | 10 |  |
| 8 | Evaluation of the sustainability and impact of designed solutions in societal and environmental contexts | 10 |  |
| 9 | Individual Contribution (Viva) | 20 |  |
| 10 | Teamwork and Diversity | 10 |  |
|  | TOTAL | 120 |  |

**Signature of Evaluator: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# Academic Honesty Statement:

**IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and Write your name in your own handwriting. *You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.***

*“In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course.”*

|  |  |  |
| --- | --- | --- |
| **Full Name: Jishnu Mahmud**  **Student ID: 1706047** | **Full Name: Raiyan Basher**  **Student ID: 1706034** |  |
| **Full Name: Tanvir Muntakim**  **Student ID: 1706048** | **Full Name: Shoaib Ahmed**  **Student ID: 1706046** |  |

Table of Contents

[Evaluation Form: 1](#_Toc110953010)

[Academic Honesty Statement: 1](#_Toc110953011)

[1 Abstract 1](#_Toc110953012)

[2 Introduction 1](#_Toc110953013)

[3 Design 2](#_Toc110953014)

[3.1 Design Method 2](#_Toc110953015)

[3.2 Circuit Diagram 4](#_Toc110953016)

[3.3 Full Source Code of Firmware 4](#_Toc110953017)

[4 Implementation 15](#_Toc110953018)

[4.1 Description 15](#_Toc110953019)

[4.2 Results 16](#_Toc110953020)

[4.3 Github Link 18](#_Toc110953021)

[4.4 YouTube Link 18](#_Toc110953022)

[5 Design Analysis and Evaluation 18](#_Toc110953023)

[5.1 Novelty 18](#_Toc110953024)

[5.2 Project Management and Cost Analysis 18](#_Toc110953025)

[5.2.1 Bill of Materials 18](#_Toc110953026)

[5.2.2 Calculation of Per Unit Cost of Prototype 19](#_Toc110953027)

[5.2.3 Calculation of Per Unit Cost of Mass-Produced Unit 19](#_Toc110953028)

[5.2.4 Timeline of Project Implementation 19](#_Toc110953029)

[5.3 Practical Considerations of the Design to Address Public Health and Safety, Environment, Cultural, and Societal Needs 20](#_Toc110953030)

[5.3.1 Considerations to public health and safety 20](#_Toc110953031)

[5.3.2 Considerations to environment 20](#_Toc110953032)

[5.3.3 Considerations to cultural and societal needs 20](#_Toc110953033)

[5.4 Assessment of the Impact of the Project on Societal, Health, Safety, Legal and Cultural Issues 21](#_Toc110953034)

[5.4.1 Assessment of Societal Issues 21](#_Toc110953035)

[5.4.2 Assessment of Health and Safety Issues 21](#_Toc110953036)

[5.4.3 Assessment of Legal Issues 21](#_Toc110953037)

[5.4.4 Assessment of Cultural Issues 21](#_Toc110953038)

[5.5 Evaluation of the Sustainability the and Impact of the Designed Solution in the Societal and Environmental Contexts 21](#_Toc110953039)

[5.5.1 Evaluation of Sustainability 21](#_Toc110953040)

[5.5.2 Evaluation of Impact of Design in Societal Context 21](#_Toc110953041)

[5.5.3 Evaluation of Impact of Design in Environmental Context 21](#_Toc110953042)

[6 Reflection on Individual and Team work 22](#_Toc110953043)

[6.1 Individual Contribution of Each Member 22](#_Toc110953044)

[6.2 Mode of TeamWork 22](#_Toc110953045)

[6.3 Diversity Statement of Team 23](#_Toc110953046)

[6.4 Log Book of Project Impelementation 23](#_Toc110953047)

[7 References 23](#_Toc110953048)

# Abstract

|  |
| --- |
| Attendance is a term very much intertwined with modern education. Attendance in simple terms is the proof of presence of an individual and in education it is the proof that a student has attended a class.  In the year of 1852, the General Court of Massachusetts passed the law “An Act Concerning the Attendance of Children at School (1852)”, which required the schools to record and evaluate based on attendance. From then onwards this idea have spread like wildfire in educational institutions all around the world.  In the 2020s when every device is being connected to the internet, unlocking its’ full potential and considerable changes have been brought out in the way we carry, depend on, and use these gadgets, the primitive concept of class attendance has somewhat been left behind. We are therefore proud to introduce our version of an automatic biometric fingerprint attendance system. |

# Introduction

# Our biometric fingerprint attendance system, or TruID as we call it gives us a simple and cost-effective solution to take and keep records of the attendance of a class. It is a very simple device, easy to operate and portable, all the factors that a device like this needs.

# Each teacher must carry each of these devices with themselves. Once plugged into the PC, the students can simply use their finger as their proof of id and enter the class. Often, mischief is done after the attendance have been given! So, our device required to scan the finger one more time once the students leave the classroom at the end of the class. Out TruID will not only provide attendance to the student and upload it to a locally hosted site, but also tell the teacher the duration for which the student was present in the class! There is another novelty idea that our TruID carries out and it is best kept secret for now!

# Design

## Design Method

## After a hefty time of debate of how we are going to start to implement our project, the one thing we could agree on was that we wanted a product that could be usable in the field. We wanted to avoid just being stuck at a prototype and wanted to push towards something which was more professional and deployable. This meant that we had to look at codes of similar projects, by professionals. Some codes were accessible to us that others and we picked a couple to help us (reference section).

## First:

## Before introducing any novelties, we wanted to make the simple form of an attendance system. This involved getting the desired parts. The parts that were initially needed were the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Name of component | Description | Quantity |
| 1 | NodeMCU | ESP8266-12E Board | 1 |
| 2 | Fingerprint Sensor | R305/R307 Module | 1 |
| 3 | OLED Display | 0.96" SSD1306 I2C OLED Display | 1 |
| 4 | Connecting Wires | Jumper Wires | 20 |
| 5 | Breadboard | -- | 4 |

## Some of the parts that were demanded by the tutorials and the codes we could find, were not available in Dhaka. So, we found out some equivalents which could be used instead. The codes were changed accordingly.

## Second

## By the end of the Mid break, we had assembled a very basic circuit to test out the fundamental working of the code. Instead of the NodeMCU we had used the simpler Arduino UNO. The OLED display was not used but rather a simpler LCD display was used. Once this was complete, we tried to incorporate the parts listed above.

## Third

## Once we could work with the advanced parts, i.e., the NodeMCU and the OLED display was incorporated we started to tweak so that the project was our very own.

## A temperature sensor was to be added. Thus, a sonar was used that senses the distance and the temperature of a source via infrared imaging. The spare pins in the NodeMCU were used to act as an input pin. The code was modified again to check the temperature of the student before they were allowed to give their attendance.

## Another change that was made was to record the duration of each student’s attendance. Once the class was over students are going to give their biometrics one more time. This causes TruID to recognize the student and record the time of departure. Thus, the whole duration was recorded.

## Fourth

## The next step was to use a locally hosted website using a third-party software to record the data we have taken. This can be downloaded later for safekeeping by the instructor, and has an interface as shown below.

## Since, none of the team members were absolute professionals in HTML, and due to shortage of time we used a third-party software to transfer the data we took to the website.

## Fifth

## The PCB design of our final circuitry was started. Done on proteus 8.8 professional, this took around a week to perfect. Lastly the outer structure of our project was made, which concluded our project.

## Circuit Diagram

## 

## Full Source Code of Firmware

## The code is attached from the next page for clarity…

|  |  |
| --- | --- |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*libraries\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #include <SPI.h>  #include <Wire.h>  #include <WiFiClient.h>  #include <ESP8266WiFi.h>  #include <SoftwareSerial.h>  #include <ESP8266WebServer.h>  #include <ESP8266HTTPClient.h>  #include <Adafruit\_GFX.h> //https://github.com/adafruit/Adafruit-GFX-Library  #include <Adafruit\_SSD1306.h> //https://github.com/adafruit/Adafruit\_SSD1306  #include <Adafruit\_Fingerprint.h> //https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #define tempPin1 0 // 8 pin of the arduino as the 1st binary digit of temp sensor, sent to D3 of nodeMCU  #define tempPin2 2 // 9 pin of the arduino as the 2nd binary digit of temp sensor, sent to D4 of nodeMCU  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Fingerprint scanner Pins  #define Finger\_Rx 14 //D5, connect sensor Tx here  #define Finger\_Tx 12 //D6, connect sensor Rx here  // Declaration for SSD1306 display connected using software I2C  #define SCREEN\_WIDTH 128 // OLED display width, in pixels  #define SCREEN\_HEIGHT 64 // OLED display height, in pixels  #define OLED\_RESET 0 // Reset pin # (or -1 if sharing Arduino reset pin)  Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  SoftwareSerial mySerial(Finger\_Rx, Finger\_Tx);  Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial);  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  /\* Set these to your desired credentials. \*/  const char \*ssid = "Enma"; //ENTER YOUR WIFI SETTINGS  const char \*password = "password";  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  String postData ; // post array that will be send to the website  String link = "http://192.168.141.55/biometricattendance/getdata.php"; //computer IP or the server domain  int FingerID = 0; // The Fingerprint ID from the scanner  uint8\_t id;  int sickMeter = 0;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Biometric Icons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #define Wifi\_start\_width 54  #define Wifi\_start\_height 49  const uint8\_t PROGMEM Wifi\_start\_bits[] = {  0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x1f,0xf0,0x00,0x00,0x00  ,0x00,0x03,0xff,0xff,0x80,0x00,0x00  ,0x00,0x1f,0xf0,0x1f,0xf0,0x00,0x00  ,0x00,0x7e,0x00,0x00,0xfc,0x00,0x00  ,0x01,0xf0,0x00,0x00,0x1f,0x00,0x00  ,0x03,0xc0,0x00,0x00,0x07,0xc0,0x00  ,0x0f,0x00,0x00,0x00,0x01,0xe0,0x00  ,0x1c,0x00,0x00,0x00,0x00,0x70,0x00  ,0x38,0x00,0x07,0xc0,0x00,0x38,0x00  ,0x70,0x00,0xff,0xfe,0x00,0x1e,0x00  ,0xe0,0x03,0xfc,0x7f,0xc0,0x0e,0x00  ,0x00,0x1f,0x80,0x03,0xf0,0x00,0x00  ,0x00,0x3c,0x00,0x00,0x78,0x00,0x00  ,0x00,0xf0,0x00,0x00,0x1c,0x00,0x00  ,0x01,0xe0,0x00,0x00,0x0c,0x00,0x00  ,0x03,0x80,0x00,0x00,0x00,0x00,0x00  ,0x03,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x3f,0xf8,0x07,0x1e,0x00  ,0x00,0x00,0xff,0xfe,0x1f,0xbf,0x80  ,0x00,0x03,0xe0,0x04,0x7f,0xff,0xc0  ,0x00,0x07,0x80,0x00,0xff,0xff,0xe0  ,0x00,0x0e,0x00,0x00,0xff,0xff,0xe0  ,0x00,0x0c,0x00,0x00,0x7f,0xff,0xc0  ,0x00,0x00,0x00,0x00,0xfe,0x07,0xe0  ,0x00,0x00,0x00,0x03,0xf8,0x03,0xf8  ,0x00,0x00,0x07,0xe7,0xf9,0xf1,0xfc  ,0x00,0x00,0x1f,0xe7,0xf1,0xf9,0xfc  ,0x00,0x00,0x1f,0xe7,0xf3,0xf9,0xfc  ,0x00,0x00,0x3f,0xe7,0xf3,0xf9,0xfc  ,0x00,0x00,0x3f,0xe7,0xf1,0xf1,0xfc  ,0x00,0x00,0x3f,0xe3,0xf8,0xe3,0xfc  ,0x00,0x00,0x3f,0xf3,0xfc,0x07,0xf8  ,0x00,0x00,0x1f,0xf0,0x7f,0x0f,0xc0  ,0x00,0x00,0x0f,0xe0,0x7f,0xff,0xe0  ,0x00,0x00,0x07,0xc0,0xff,0xff,0xe0  ,0x00,0x00,0x00,0x00,0x7f,0xff,0xe0  ,0x00,0x00,0x00,0x00,0x3f,0xff,0x80  ,0x00,0x00,0x00,0x00,0x1f,0xbf,0x00  ,0x00,0x00,0x00,0x00,0x03,0x18,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00  };  #define Wifi\_connected\_width 63  #define Wifi\_connected\_height 49  const uint8\_t PROGMEM Wifi\_connected\_bits[] = {  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x03,0xff,0xff,0x80,0x00,0x00  ,0x00,0x00,0x3f,0xff,0xff,0xf8,0x00,0x00  ,0x00,0x01,0xff,0xff,0xff,0xff,0x00,0x00  ,0x00,0x0f,0xff,0xff,0xff,0xff,0xe0,0x00  ,0x00,0x3f,0xff,0xc0,0x07,0xff,0xf8,0x00  ,0x00,0xff,0xf8,0x00,0x00,0x3f,0xfe,0x00  ,0x03,0xff,0x80,0x00,0x00,0x03,0xff,0x80  ,0x07,0xfe,0x00,0x00,0x00,0x00,0xff,0xc0  ,0x1f,0xf8,0x00,0x00,0x00,0x00,0x3f,0xf0  ,0x3f,0xe0,0x01,0xff,0xff,0x00,0x0f,0xf8  ,0x7f,0x80,0x0f,0xff,0xff,0xe0,0x03,0xfc  ,0xff,0x00,0x7f,0xff,0xff,0xfc,0x01,0xfe  ,0xfc,0x01,0xff,0xff,0xff,0xff,0x00,0x7e  ,0x78,0x07,0xff,0xc0,0x07,0xff,0xc0,0x3c  ,0x00,0x0f,0xfc,0x00,0x00,0x7f,0xe0,0x00  ,0x00,0x1f,0xf0,0x00,0x00,0x1f,0xf0,0x00  ,0x00,0x3f,0xc0,0x00,0x00,0x07,0xf8,0x00  ,0x00,0x7f,0x00,0x01,0x00,0x01,0xfc,0x00  ,0x00,0x7e,0x00,0x7f,0xfc,0x00,0xfc,0x00  ,0x00,0x3c,0x03,0xff,0xff,0x80,0x78,0x00  ,0x00,0x00,0x07,0xff,0xff,0xc0,0x00,0x00  ,0x00,0x00,0x1f,0xff,0xff,0xf0,0x00,0x00  ,0x00,0x00,0x3f,0xf0,0x1f,0xf8,0x00,0x00  ,0x00,0x00,0x3f,0x80,0x03,0xf8,0x00,0x00  ,0x00,0x00,0x3f,0x00,0x01,0xf8,0x00,0x00  ,0x00,0x00,0x1c,0x00,0x00,0x70,0x00,0x00  ,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x0f,0xe0,0x00,0x00,0x00  ,0x00,0x00,0x00,0x1f,0xf0,0x00,0x00,0x00  ,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00  ,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00  ,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00  ,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00  ,0x00,0x00,0x00,0x1f,0xf0,0x00,0x00,0x00  ,0x00,0x00,0x00,0x0f,0xe0,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  };  #define FinPr\_start\_width 64  #define FinPr\_start\_height 64  const uint8\_t PROGMEM FinPr\_start\_bits[] = {  0x00,0x00,0x00,0x1f,0xe0,0x00,0x00,0x00  ,0x00,0x00,0x01,0xff,0xfe,0x00,0x00,0x00  ,0x00,0x00,0x03,0xff,0xff,0x80,0x00,0x00  ,0x00,0x00,0x0f,0xc0,0x0f,0xe0,0x00,0x00  ,0x00,0x00,0x1f,0x00,0x01,0xf8,0x00,0x00  ,0x00,0x00,0x3c,0x00,0x00,0x7c,0x00,0x00  ,0x00,0x00,0x78,0x00,0x00,0x3e,0x00,0x00  ,0x00,0x00,0xf0,0x3f,0xf8,0x0f,0x00,0x00  ,0x00,0x01,0xe0,0xff,0xfe,0x07,0x80,0x00  ,0x00,0x03,0xc3,0xff,0xff,0x03,0x80,0x00  ,0x00,0x03,0x87,0xc0,0x07,0xc3,0xc0,0x00  ,0x00,0x07,0x0f,0x00,0x03,0xe1,0xc0,0x00  ,0x00,0x0f,0x0e,0x00,0x00,0xe0,0xe0,0x00  ,0x00,0x0e,0x1c,0x00,0x00,0xf0,0xe0,0x00  ,0x00,0x0c,0x3c,0x1f,0xe0,0x70,0xe0,0x00  ,0x00,0x00,0x38,0x3f,0xf0,0x38,0x70,0x00  ,0x00,0x00,0x78,0x78,0xf8,0x38,0x70,0x00  ,0x00,0x00,0x70,0x70,0x3c,0x18,0x70,0x00  ,0x00,0x00,0xe0,0xe0,0x1e,0x1c,0x70,0x00  ,0x00,0x03,0xe1,0xe0,0x0e,0x1c,0x70,0x00  ,0x00,0x0f,0xc1,0xc3,0x0e,0x1c,0x70,0x00  ,0x00,0x3f,0x03,0xc3,0x8e,0x1c,0x70,0x00  ,0x00,0x3e,0x03,0x87,0x0e,0x1c,0x70,0x00  ,0x00,0x30,0x07,0x07,0x0e,0x18,0xe0,0x00  ,0x00,0x00,0x0e,0x0e,0x0e,0x38,0xe0,0x00  ,0x00,0x00,0x3e,0x1e,0x1e,0x38,0xe0,0x00  ,0x00,0x00,0xf8,0x1c,0x1c,0x38,0xe0,0x00  ,0x00,0x03,0xf0,0x38,0x3c,0x38,0xe0,0x00  ,0x00,0x3f,0xc0,0xf8,0x78,0x38,0xe0,0x00  ,0x00,0x7f,0x01,0xf0,0x70,0x38,0xf0,0x00  ,0x00,0x78,0x03,0xe0,0xe0,0x38,0x70,0x00  ,0x00,0x00,0x0f,0x81,0xe0,0x38,0x7c,0x00  ,0x00,0x00,0x3f,0x03,0xc0,0x38,0x3e,0x00  ,0x00,0x00,0xfc,0x0f,0x80,0x38,0x1e,0x00  ,0x00,0x07,0xf0,0x1f,0x1c,0x1c,0x04,0x00  ,0x00,0x3f,0xc0,0x3e,0x3f,0x1e,0x00,0x00  ,0x00,0x7f,0x00,0xf8,0x7f,0x0f,0x00,0x00  ,0x00,0x38,0x01,0xf0,0xf7,0x07,0xc0,0x00  ,0x00,0x00,0x07,0xe1,0xe3,0x83,0xf8,0x00  ,0x00,0x00,0x3f,0x87,0xc3,0xc0,0xfc,0x00  ,0x00,0x01,0xfe,0x0f,0x81,0xe0,0x3c,0x00  ,0x00,0x0f,0xf8,0x1f,0x00,0xf0,0x00,0x00  ,0x00,0x1f,0xc0,0x7c,0x00,0x7c,0x00,0x00  ,0x00,0x1e,0x01,0xf8,0x00,0x3f,0x00,0x00  ,0x00,0x00,0x07,0xe0,0x78,0x0f,0xc0,0x00  ,0x00,0x00,0x3f,0x81,0xfe,0x07,0xf0,0x00  ,0x00,0x01,0xfe,0x07,0xff,0x01,0xf0,0x00  ,0x00,0x07,0xf8,0x0f,0x87,0x80,0x30,0x00  ,0x00,0x07,0xc0,0x3f,0x03,0xe0,0x00,0x00  ,0x00,0x06,0x00,0xfc,0x01,0xf8,0x00,0x00  ,0x00,0x00,0x03,0xf0,0x00,0x7e,0x00,0x00  ,0x00,0x00,0x0f,0xc0,0x00,0x3f,0x80,0x00  ,0x00,0x00,0x7f,0x00,0xf8,0x0f,0x80,0x00  ,0x00,0x00,0xfc,0x03,0xfe,0x01,0x80,0x00  ,0x00,0x00,0xf0,0x1f,0xff,0x80,0x00,0x00  ,0x00,0x00,0x00,0x7f,0x07,0xe0,0x00,0x00  ,0x00,0x00,0x00,0xfc,0x03,0xf8,0x00,0x00  ,0x00,0x00,0x03,0xf0,0x00,0x78,0x00,0x00  ,0x00,0x00,0x0f,0xc0,0x00,0x18,0x00,0x00  ,0x00,0x00,0x0f,0x01,0xf8,0x00,0x00,0x00  ,0x00,0x00,0x00,0x07,0xfe,0x00,0x00,0x00  ,0x00,0x00,0x00,0x1f,0xfe,0x00,0x00,0x00  ,0x00,0x00,0x00,0x1e,0x0e,0x00,0x00,0x00  ,0x00,0x00,0x00,0x18,0x00,0x00,0x00,0x00  };  //---------------------------------------------------------------  #define FinPr\_valid\_width 64  #define FinPr\_valid\_height 64  const uint8\_t PROGMEM FinPr\_valid\_bits[] = {  0x00,0x00,0x03,0xfe,0x00,0x00,0x00,0x00  ,0x00,0x00,0x1f,0xff,0xe0,0x00,0x00,0x00  ,0x00,0x00,0x7f,0xff,0xf8,0x00,0x00,0x00  ,0x00,0x00,0xfc,0x00,0xfe,0x00,0x00,0x00  ,0x00,0x03,0xe0,0x00,0x1f,0x00,0x00,0x00  ,0x00,0x07,0xc0,0x00,0x07,0x80,0x00,0x00  ,0x00,0x0f,0x80,0x00,0x03,0xe0,0x00,0x00  ,0x00,0x0e,0x03,0xff,0x01,0xe0,0x00,0x00  ,0x00,0x1c,0x1f,0xff,0xe0,0xf0,0x00,0x00  ,0x00,0x3c,0x3f,0xff,0xf0,0x78,0x00,0x00  ,0x00,0x78,0x7c,0x00,0xf8,0x3c,0x00,0x00  ,0x00,0x70,0xf0,0x00,0x3c,0x1c,0x00,0x00  ,0x00,0xe1,0xe0,0x00,0x1e,0x1c,0x00,0x00  ,0x00,0xe1,0xc0,0x00,0x0f,0x0e,0x00,0x00  ,0x00,0xc3,0x81,0xfc,0x07,0x0e,0x00,0x00  ,0x00,0x03,0x83,0xff,0x07,0x8e,0x00,0x00  ,0x00,0x07,0x07,0x8f,0x83,0x87,0x00,0x00  ,0x00,0x0f,0x0f,0x03,0xc3,0x87,0x00,0x00  ,0x00,0x1e,0x0e,0x01,0xc3,0x87,0x00,0x00  ,0x00,0x3c,0x1c,0x00,0xe1,0x87,0x00,0x00  ,0x00,0xf8,0x1c,0x30,0xe1,0x87,0x00,0x00  ,0x07,0xf0,0x38,0x70,0xe1,0x86,0x00,0x00  ,0x07,0xc0,0x78,0x70,0xe3,0x8e,0x00,0x00  ,0x02,0x00,0xf0,0xf0,0xe3,0x8e,0x00,0x00  ,0x00,0x01,0xe0,0xe0,0xe3,0x8e,0x00,0x00  ,0x00,0x03,0xc1,0xe1,0xc3,0x8e,0x00,0x00  ,0x00,0x0f,0x83,0xc3,0xc3,0x8e,0x00,0x00  ,0x00,0x7f,0x07,0x83,0x83,0x0e,0x00,0x00  ,0x07,0xfc,0x0f,0x07,0x83,0x0e,0x00,0x00  ,0x07,0xf0,0x1e,0x0f,0x03,0x0e,0x00,0x00  ,0x07,0x80,0x7c,0x1e,0x03,0x07,0x00,0x00  ,0x00,0x00,0xf8,0x3c,0x03,0x87,0x80,0x00  ,0x00,0x03,0xf0,0x78,0x03,0x83,0xc0,0x00  ,0x00,0x1f,0xc0,0xf0,0x02,0x00,0x00,0x00  ,0x00,0xff,0x01,0xe1,0xc0,0x0c,0x00,0x00  ,0x07,0xfc,0x03,0xc3,0xe1,0xff,0xc0,0x00  ,0x07,0xe0,0x0f,0x87,0xc7,0xff,0xf0,0x00  ,0x07,0x00,0x3f,0x0f,0x0f,0xff,0xfc,0x00  ,0x00,0x00,0x7c,0x3e,0x3f,0xff,0xfe,0x00  ,0x00,0x03,0xf8,0x7c,0x3f,0xff,0xff,0x00  ,0x00,0x1f,0xe0,0xf0,0x7f,0xff,0xff,0x80  ,0x00,0xff,0x83,0xe0,0xff,0xff,0xff,0x80  ,0x01,0xfc,0x07,0xc1,0xff,0xff,0xe3,0xc0  ,0x01,0xe0,0x1f,0x01,0xff,0xff,0xc3,0xc0  ,0x00,0x00,0xfe,0x01,0xff,0xff,0x87,0xe0  ,0x00,0x03,0xf8,0x13,0xff,0xff,0x0f,0xe0  ,0x00,0x1f,0xe0,0x73,0xff,0xfe,0x1f,0xe0  ,0x00,0x7f,0x81,0xf3,0xff,0xfc,0x1f,0xe0  ,0x00,0xfc,0x03,0xe3,0xef,0xf8,0x3f,0xe0  ,0x00,0x60,0x0f,0xc3,0xc7,0xf0,0x7f,0xe0  ,0x00,0x00,0x3f,0x03,0xc3,0xe0,0xff,0xe0  ,0x00,0x00,0xfc,0x03,0xc1,0xc1,0xff,0xe0  ,0x00,0x07,0xf0,0x13,0xe0,0x83,0xff,0xe0  ,0x00,0x0f,0xc0,0x7b,0xf8,0x07,0xff,0xe0  ,0x00,0x0f,0x01,0xf9,0xfc,0x0f,0xff,0xc0  ,0x00,0x00,0x07,0xf1,0xfe,0x1f,0xff,0xc0  ,0x00,0x00,0x1f,0xc0,0xff,0x3f,0xff,0x80  ,0x00,0x00,0x7e,0x00,0xff,0xff,0xff,0x80  ,0x00,0x00,0xfc,0x00,0x7f,0xff,0xff,0x00  ,0x00,0x00,0xf0,0x1f,0x3f,0xff,0xfe,0x00  ,0x00,0x00,0x00,0x7f,0x1f,0xff,0xfc,0x00  ,0x00,0x00,0x01,0xff,0x8f,0xff,0xf8,0x00  ,0x00,0x00,0x03,0xe0,0xe3,0xff,0xe0,0x00  ,0x00,0x00,0x01,0x80,0x00,0x7f,0x00,0x00  };  //---------------------------------------------------------------  #define FinPr\_invalid\_width 64  #define FinPr\_invalid\_height 64  const uint8\_t PROGMEM FinPr\_invalid\_bits[] = {  0x00,0x00,0x03,0xfe,0x00,0x00,0x00,0x00  ,0x00,0x00,0x1f,0xff,0xe0,0x00,0x00,0x00  ,0x00,0x00,0x7f,0xff,0xf8,0x00,0x00,0x00  ,0x00,0x00,0xfc,0x00,0xfe,0x00,0x00,0x00  ,0x00,0x03,0xe0,0x00,0x1f,0x00,0x00,0x00  ,0x00,0x07,0xc0,0x00,0x07,0x80,0x00,0x00  ,0x00,0x0f,0x80,0x00,0x03,0xe0,0x00,0x00  ,0x00,0x0e,0x03,0xff,0x01,0xe0,0x00,0x00  ,0x00,0x1c,0x1f,0xff,0xe0,0xf0,0x00,0x00  ,0x00,0x3c,0x3f,0xff,0xf0,0x78,0x00,0x00  ,0x00,0x78,0x7c,0x00,0xf8,0x3c,0x00,0x00  ,0x00,0x70,0xf0,0x00,0x3c,0x1c,0x00,0x00  ,0x00,0xe1,0xe0,0x00,0x1e,0x1c,0x00,0x00  ,0x00,0xe1,0xc0,0x00,0x0f,0x0e,0x00,0x00  ,0x00,0xc3,0x81,0xfc,0x07,0x0e,0x00,0x00  ,0x00,0x03,0x83,0xff,0x07,0x8e,0x00,0x00  ,0x00,0x07,0x07,0x8f,0x83,0x87,0x00,0x00  ,0x00,0x0f,0x0f,0x03,0xc3,0x87,0x00,0x00  ,0x00,0x1e,0x0e,0x01,0xc3,0x87,0x00,0x00  ,0x00,0x3c,0x1c,0x00,0xe1,0x87,0x00,0x00  ,0x00,0xf8,0x1c,0x30,0xe1,0x87,0x00,0x00  ,0x07,0xf0,0x38,0x70,0xe1,0x86,0x00,0x00  ,0x07,0xc0,0x78,0x70,0xe3,0x8e,0x00,0x00  ,0x02,0x00,0xf0,0xf0,0xe3,0x8e,0x00,0x00  ,0x00,0x01,0xe0,0xe0,0xe3,0x8e,0x00,0x00  ,0x00,0x03,0xc1,0xe1,0xc3,0x8e,0x00,0x00  ,0x00,0x0f,0x83,0xc3,0xc3,0x8e,0x00,0x00  ,0x00,0x7f,0x07,0x83,0x83,0x0e,0x00,0x00  ,0x07,0xfc,0x0f,0x07,0x83,0x0e,0x00,0x00  ,0x07,0xf0,0x1e,0x0f,0x03,0x0e,0x00,0x00  ,0x07,0x80,0x7c,0x1e,0x03,0x07,0x00,0x00  ,0x00,0x00,0xf8,0x3c,0x03,0x87,0x80,0x00  ,0x00,0x03,0xf0,0x78,0x03,0x83,0xc0,0x00  ,0x00,0x1f,0xc0,0xf0,0x02,0x00,0x00,0x00  ,0x00,0xff,0x01,0xe1,0xc0,0x00,0x00,0x00  ,0x07,0xfc,0x03,0xc3,0xe1,0xff,0xc0,0x00  ,0x07,0xe0,0x0f,0x87,0xc7,0xff,0xf0,0x00  ,0x07,0x00,0x3f,0x0f,0x0f,0xff,0xf8,0x00  ,0x00,0x00,0x7c,0x3e,0x1f,0xff,0xfe,0x00  ,0x00,0x03,0xf8,0x7c,0x3f,0xff,0xff,0x00  ,0x00,0x1f,0xe0,0xf0,0x7f,0xff,0xff,0x00  ,0x00,0xff,0x83,0xe0,0xfe,0xff,0xbf,0x80  ,0x01,0xfc,0x07,0xc0,0xfc,0x7f,0x1f,0xc0  ,0x01,0xe0,0x1f,0x01,0xf8,0x3e,0x0f,0xc0  ,0x00,0x00,0xfe,0x01,0xf8,0x1c,0x07,0xe0  ,0x00,0x03,0xf8,0x13,0xf8,0x00,0x0f,0xe0  ,0x00,0x1f,0xe0,0x73,0xfc,0x00,0x1f,0xe0  ,0x00,0x7f,0x81,0xf3,0xfe,0x00,0x3f,0xe0  ,0x00,0xfc,0x03,0xe3,0xff,0x00,0x7f,0xe0  ,0x00,0x60,0x0f,0xc3,0xff,0x80,0xff,0xe0  ,0x00,0x00,0x3f,0x03,0xff,0x00,0x7f,0xe0  ,0x00,0x00,0xfc,0x03,0xfe,0x00,0x3f,0xe0  ,0x00,0x07,0xf0,0x13,0xfc,0x00,0x1f,0xe0  ,0x00,0x0f,0xc0,0x79,0xf8,0x08,0x0f,0xe0  ,0x00,0x0f,0x01,0xf9,0xf8,0x1c,0x0f,0xc0  ,0x00,0x00,0x07,0xf1,0xfc,0x3e,0x1f,0xc0  ,0x00,0x00,0x1f,0xc0,0xfe,0x7f,0x3f,0x80  ,0x00,0x00,0x7e,0x00,0xff,0xff,0xff,0x80  ,0x00,0x00,0xfc,0x00,0x7f,0xff,0xff,0x00  ,0x00,0x00,0xf0,0x1f,0x3f,0xff,0xfe,0x00  ,0x00,0x00,0x00,0x7f,0x1f,0xff,0xfc,0x00  ,0x00,0x00,0x01,0xff,0x8f,0xff,0xf8,0x00  ,0x00,0x00,0x03,0xe0,0xe3,0xff,0xe0,0x00  ,0x00,0x00,0x01,0x80,0x00,0x7f,0x00,0x00  };  //---------------------------------------------------------------  #define FinPr\_failed\_width 64  #define FinPr\_failed\_height 64  const uint8\_t PROGMEM FinPr\_failed\_bits[] = {  0x00,0x00,0x3f,0xe0,0x00,0x00,0x00,0x00  ,0x00,0x01,0xff,0xfe,0x00,0x00,0x00,0x00  ,0x00,0x0f,0xc0,0x1f,0x80,0x00,0x00,0x00  ,0x00,0x1e,0x00,0x03,0xc0,0x00,0x00,0x00  ,0x00,0x78,0x00,0x00,0xf0,0x00,0x00,0x00  ,0x00,0xe0,0x00,0x00,0x38,0x00,0x00,0x00  ,0x01,0xc0,0x00,0x00,0x1c,0x00,0x00,0x00  ,0x03,0x80,0x00,0x00,0x0e,0x00,0x00,0x00  ,0x07,0x00,0x7f,0xe0,0x07,0x00,0x00,0x00  ,0x06,0x01,0xff,0xf8,0x03,0x00,0x00,0x00  ,0x0c,0x03,0xc0,0x3c,0x03,0x80,0x00,0x00  ,0x1c,0x0f,0x00,0x0e,0x01,0x80,0x00,0x00  ,0x18,0x0c,0x00,0x03,0x00,0xc0,0x00,0x00  ,0x18,0x18,0x00,0x01,0x80,0xc0,0x00,0x00  ,0x30,0x38,0x00,0x01,0xc0,0xe0,0x00,0x00  ,0x30,0x30,0x0f,0x00,0xc0,0x60,0x00,0x00  ,0x30,0x30,0x3f,0xc0,0xe0,0x60,0x00,0x00  ,0x70,0x60,0x78,0xe0,0x60,0x60,0x00,0x00  ,0x60,0x60,0x60,0x60,0x60,0x70,0x00,0x00  ,0x60,0x60,0x60,0x60,0x60,0x30,0x00,0x00  ,0x60,0x60,0x60,0x60,0x30,0x30,0x00,0x00  ,0x60,0x60,0x60,0x30,0x30,0x20,0x00,0x00  ,0x60,0x60,0x60,0x30,0x30,0x01,0xe0,0x00  ,0x60,0x60,0x60,0x30,0x30,0x0f,0xfc,0x00  ,0x60,0x60,0x60,0x30,0x30,0x3f,0xff,0x00  ,0x60,0x60,0x60,0x30,0x18,0x78,0x03,0x80  ,0x60,0x60,0x60,0x30,0x1c,0x60,0x01,0x80  ,0x60,0x60,0x30,0x38,0x0c,0xc0,0x00,0xc0  ,0x00,0x60,0x30,0x18,0x00,0xc0,0x00,0xc0  ,0x00,0x60,0x30,0x18,0x00,0xc0,0x00,0xc0  ,0x00,0xe0,0x30,0x0c,0x01,0xc0,0x00,0xe0  ,0x00,0xc0,0x18,0x0e,0x01,0xc0,0x00,0xe0  ,0x60,0xc0,0x18,0x07,0x01,0xc0,0x00,0xe0  ,0x01,0xc0,0x1c,0x03,0x81,0xc0,0x00,0xe0  ,0x01,0x80,0x0c,0x01,0xc1,0xc0,0x00,0xe0  ,0x03,0x80,0x0e,0x00,0xf1,0xc0,0x00,0xe0  ,0x0f,0x00,0x06,0x00,0x01,0xc0,0x00,0xe0  ,0x3e,0x01,0x03,0x00,0x01,0xc0,0x00,0xe0  ,0x30,0x03,0x83,0x80,0x1f,0xff,0xff,0xfe  ,0x00,0x03,0x81,0xc0,0x3f,0xff,0xff,0xff  ,0x00,0x07,0xc0,0xe0,0x30,0x00,0x00,0x03  ,0x00,0x0e,0xc0,0x78,0x30,0x00,0x00,0x03  ,0x00,0x3c,0x60,0x1e,0x30,0x00,0x00,0x03  ,0x00,0x78,0x70,0x0f,0x30,0x00,0x00,0x03  ,0x03,0xe0,0x38,0x03,0x30,0x00,0x00,0x03  ,0x07,0x80,0x1c,0x00,0x30,0x00,0x00,0x03  ,0xc0,0x00,0x0f,0x00,0x30,0x00,0x00,0x03  ,0xc0,0x00,0x03,0x80,0x30,0x01,0xe0,0x03  ,0x00,0x18,0x01,0xe0,0x30,0x03,0xf0,0x03  ,0x00,0x18,0x00,0x7c,0x30,0x07,0x38,0x03  ,0x00,0x0c,0x00,0x1f,0x30,0x06,0x18,0x03  ,0x18,0x0e,0x00,0x07,0x30,0x06,0x18,0x03  ,0x0c,0x07,0x80,0x00,0x30,0x07,0x38,0x03  ,0x0e,0x03,0xc0,0x00,0x30,0x03,0x30,0x03  ,0x07,0x00,0xf0,0x00,0x30,0x03,0x30,0x03  ,0x03,0x00,0x7e,0x00,0x30,0x03,0x30,0x03  ,0x01,0x80,0x1f,0xc0,0x30,0x03,0x30,0x03  ,0x01,0xc0,0x03,0xe1,0x30,0x07,0xf8,0x03  ,0x00,0xf0,0x00,0x01,0x30,0x03,0xf0,0x03  ,0x00,0x38,0x00,0x00,0x30,0x00,0x00,0x03  ,0x00,0x1e,0x00,0x00,0x30,0x00,0x00,0x03  ,0x00,0x07,0xc0,0x00,0x30,0x00,0x00,0x03  ,0x00,0x01,0xff,0x80,0x3f,0xff,0xff,0xff  ,0x00,0x00,0x3f,0x80,0x1f,0xff,0xff,0xfe  };  //---------------------------------------------------------------  #define FinPr\_scan\_width 64  #define FinPr\_scan\_height 64  const uint8\_t PROGMEM FinPr\_scan\_bits[] = {  0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x1f,0xf8,0x00,0x00,0x00  ,0x00,0x00,0x00,0x7f,0xff,0x00,0x00,0x00  ,0x00,0x00,0x01,0xfc,0x7f,0xc0,0x00,0x00  ,0x00,0x00,0x03,0xc0,0x03,0xe0,0x00,0x00  ,0x00,0x00,0x07,0x80,0x00,0xf0,0x00,0x00  ,0x00,0x00,0x0e,0x00,0x00,0x3c,0x00,0x00  ,0x00,0x00,0x1c,0x1f,0xfc,0x1c,0x00,0x00  ,0x00,0x00,0x38,0x7f,0xfe,0x0e,0x00,0x00  ,0x00,0x00,0x78,0xf8,0x0f,0x87,0x00,0x00  ,0x00,0x00,0x71,0xe0,0x03,0xc7,0x00,0x00  ,0x00,0x00,0xe3,0x80,0x01,0xc3,0x80,0x00  ,0x00,0x00,0xc3,0x83,0xc0,0xe3,0x80,0x00  ,0x00,0x00,0xc7,0x0f,0xf0,0x71,0x80,0x00  ,0x00,0x00,0x06,0x1f,0xf8,0x71,0xc0,0x00  ,0x00,0x00,0x0e,0x1c,0x3c,0x31,0xc0,0x00  ,0x00,0x00,0x1c,0x38,0x1c,0x31,0xc0,0x00  ,0x00,0x00,0x38,0x70,0x0e,0x39,0xc0,0x00  ,0x00,0x01,0xf0,0x71,0x8e,0x39,0xc0,0x00  ,0x00,0x03,0xe0,0xe1,0x86,0x31,0xc0,0x00  ,0x00,0x03,0x81,0xe3,0x8e,0x31,0x80,0x00  ,0x00,0x00,0x03,0xc3,0x8e,0x33,0x80,0x00  ,0x00,0x00,0x07,0x87,0x0c,0x73,0x80,0x00  ,0x00,0x00,0x1f,0x0e,0x1c,0x73,0x80,0x00  ,0x7f,0xff,0xff,0xff,0xff,0xff,0xff,0xfe  ,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff  ,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff  ,0x7f,0xff,0xff,0xff,0xff,0xff,0xff,0xfe  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x03,0xf0,0x1e,0x3e,0x1c,0x00,0x00  ,0x00,0x03,0x80,0x7c,0x77,0x0f,0x00,0x00  ,0x00,0x00,0x01,0xf0,0xe3,0x07,0xc0,0x00  ,0x00,0x00,0x07,0xe3,0xc3,0x81,0xf0,0x00  ,0x00,0x00,0x3f,0x87,0x81,0xc0,0x60,0x00  ,0x00,0x01,0xfc,0x1f,0x00,0xf0,0x00,0x00  ,0x00,0x01,0xe0,0x3c,0x00,0x7c,0x00,0x00  ,0x00,0x00,0x00,0xf8,0x78,0x1f,0x00,0x00  ,0x00,0x00,0x07,0xe0,0xfc,0x0f,0xc0,0x00  ,0x00,0x00,0x3f,0x83,0xef,0x03,0xc0,0x00  ,0x00,0x00,0xfc,0x0f,0x87,0x80,0x00,0x00  ,0x00,0x00,0x70,0x1f,0x03,0xe0,0x00,0x00  ,0x00,0x00,0x00,0x7c,0x00,0xf8,0x00,0x00  ,0x00,0x00,0x01,0xf0,0x00,0x3e,0x00,0x00  ,0x00,0x00,0x0f,0xc0,0xf8,0x0f,0x00,0x00  ,0x00,0x00,0x1f,0x03,0xfe,0x02,0x00,0x00  ,0x00,0x00,0x0c,0x0f,0x8f,0x80,0x00,0x00  ,0x00,0x00,0x00,0x3f,0x03,0xe0,0x00,0x00  ,0x00,0x00,0x00,0xf8,0x00,0xf0,0x00,0x00  ,0x00,0x00,0x01,0xe0,0x00,0x30,0x00,0x00  ,0x00,0x00,0x01,0xc0,0xf8,0x00,0x00,0x00  ,0x00,0x00,0x00,0x07,0xfe,0x00,0x00,0x00  ,0x00,0x00,0x00,0x0f,0x8e,0x00,0x00,0x00  ,0x00,0x00,0x00,0x06,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  };  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | void setup() {    Serial.begin(115200);    //-----------initiate OLED display-------------    // SSD1306\_SWITCHCAPVCC = generate display voltage from 3.3V internally  if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64  Serial.println(F("SSD1306 allocation failed"));  for(;;); // Don't proceed, loop forever  }  // Show initial display buffer contents on the screen --  // the library initializes this with an Adafruit splash screen.  // you can delet these three lines if you don't want to get the Adfruit logo appear  display.display();  delay(2000); // Pause for 2 seconds  display.clearDisplay();    //---------------------------------------------    connectToWiFi();    //---------------------------------------------    // set the data rate for the sensor serial port  finger.begin(57600);  Serial.println("\n\nAdafruit finger detect test");    if (finger.verifyPassword()) {  Serial.println("Found fingerprint sensor!");  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_valid\_bits, FinPr\_valid\_width, FinPr\_valid\_height, WHITE);  display.display();  } else {  Serial.println("Did not find fingerprint sensor :(");  display.clearDisplay();  display.drawBitmap( 32, 0, FinPr\_failed\_bits, FinPr\_failed\_width, FinPr\_failed\_height, WHITE);  display.display();  while (1) { delay(1); }  }  //---------------------------------------------    finger.getTemplateCount();  Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");  Serial.println("Waiting for valid finger...");    //------------\*test the connection\*------------    //SendFingerprintID( FingerID );  // The outputs from the thermal sensor defined as input  pinMode(tempPin1, INPUT);  pinMode(tempPin2, INPUT);  // The outputs from the thermal sensor defined as input    }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void loop() {  int flag = 0;    //check if there's a connection to WiFi or not  if(WiFi.status() != WL\_CONNECTED){  connectToWiFi();  }  // Get body temperature from the thermal sensor  int D0 = 0;  int D1 = 0;  D0 = digitalRead(tempPin1);  D1 = digitalRead(tempPin2);  Serial.println("First Digital Bit:");  Serial.println(D0);  Serial.println(D1);  // [D1, D0] is the two bit binary data indicating temperature reading cases.  // 01 - good temperature, student healthy, 10 - high temperature, don't take fingerprint, 11 - temperature reading not yet taken  if(D0 == 1 && D1 == 1){ // no temperature reading  display.clearDisplay();  display.setTextSize(2);  display.setCursor(0,0);  display.print("Please get your temperature reading!");  display.display();  flag = 1;  }  if(D0 == 0 && D1 == 1){ // high temperature  flag = 1;  display.clearDisplay();  display.setTextSize(2);  display.setCursor(0,0);  display.print("Your temperature is too high!");  display.display();  delay(2000);    }  if(flag == 0){  // Get body temperature from the thermal sensor  //---------------------------------------------  //If there no fingerprint has been scanned return -1 or -2 if there an error or 0 if there nothing, The ID start form 1 to 127  FingerID = getFingerprintID(); // Get the Fingerprint ID from the Scanner  delay(50); //don't need to run this at full speed.    //---------------------------------------------    DisplayFingerprintID();    //---------------------------------------------    ChecktoAddID();    //---------------------------------------------    ChecktoDeleteID();    //---------------------------------------------  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*Display the fingerprint ID state on the OLED\*\*\*\*\*\*\*\*\*\*\*\*\*  void DisplayFingerprintID(){  //Fingerprint has been detected  if (FingerID > 0){  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_valid\_bits, FinPr\_valid\_width, FinPr\_valid\_height, WHITE);  display.display();    SendFingerprintID( FingerID ); // Send the Fingerprint ID to the website.    }  //---------------------------------------------  //No finger detected  else if (FingerID == 0){  display.clearDisplay();  display.drawBitmap( 32, 0, FinPr\_start\_bits, FinPr\_start\_width, FinPr\_start\_height, WHITE);  display.display();  }  //---------------------------------------------  //Didn't find a match  else if (FingerID == -1){  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_invalid\_bits, FinPr\_invalid\_width, FinPr\_invalid\_height, WHITE);  display.display();  }  //---------------------------------------------  //Didn't find the scanner or there an error  else if (FingerID == -2){  display.clearDisplay();  display.drawBitmap( 32, 0, FinPr\_failed\_bits, FinPr\_failed\_width, FinPr\_failed\_height, WHITE);  display.display();  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*send the fingerprint ID to the website\*\*\*\*\*\*\*\*\*\*\*\*\*  void SendFingerprintID( int finger ){    HTTPClient http; //Declare object of class HTTPClient  //Post Data  postData = "FingerID=" + String(finger); // Add the Fingerprint ID to the Post array in order to send it  // Post methode    http.begin(link); //initiate HTTP request, put your Website URL or Your Computer IP  http.addHeader("Content-Type", "application/x-www-form-urlencoded"); //Specify content-type header    int httpCode = http.POST(postData); //Send the request  String payload = http.getString(); //Get the response payload    Serial.println(httpCode); //Print HTTP return code  Serial.println(payload); //Print request response payload  Serial.println(postData); //Post Data  Serial.println(finger); //Print fingerprint ID    if (payload.substring(0, 5) == "login") {  String user\_name = payload.substring(5);  // Serial.println(user\_name);    display.clearDisplay();  display.setTextSize(2); // Normal 2:2 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(15,0); // Start at top-left corner  display.print(F("Welcome"));  display.setCursor(0,20);  display.print(user\_name);  display.display();  }  else if (payload.substring(0, 6) == "logout") {  String user\_name = payload.substring(6);  // Serial.println(user\_name);    display.clearDisplay();  display.setTextSize(2); // Normal 2:2 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(10,0); // Start at top-left corner  display.print(F("Good Bye"));  display.setCursor(0,20);  display.print(user\_name);  display.display();  }  delay(1000);    postData = "";  http.end(); //Close connection  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Get the Fingerprint ID\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  int 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 0;  case FINGERPRINT\_PACKETRECIEVEERR:  //Serial.println("Communication error");  return -2;  case FINGERPRINT\_IMAGEFAIL:  //Serial.println("Imaging error");  return -2;  default:  //Serial.println("Unknown error");  return -2;  }  // OK success!  p = finger.image2Tz();  switch (p) {  case FINGERPRINT\_OK:  //Serial.println("Image converted");  break;  case FINGERPRINT\_IMAGEMESS:  //Serial.println("Image too messy");  return -1;  case FINGERPRINT\_PACKETRECIEVEERR:  //Serial.println("Communication error");  return -2;  case FINGERPRINT\_FEATUREFAIL:  //Serial.println("Could not find fingerprint features");  return -2;  case FINGERPRINT\_INVALIDIMAGE:  //Serial.println("Could not find fingerprint features");  return -2;  default:  //Serial.println("Unknown error");  return -2;  }  // 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 -2;  } else if (p == FINGERPRINT\_NOTFOUND) {  //Serial.println("Did not find a match");  return -1;  } else {  //Serial.println("Unknown error");  return -2;  }  // found a match!  //Serial.print("Found ID #"); Serial.print(finger.fingerID);  //Serial.print(" with confidence of "); Serial.println(finger.confidence);    return finger.fingerID;  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Check if there a Fingerprint ID to delete\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void ChecktoDeleteID(){    HTTPClient http; //Declare object of class HTTPClient  //Post Data  postData = "DeleteID=check"; // Add the Fingerprint ID to the Post array in order to send it  // Post methode    http.begin(link); //initiate HTTP request, put your Website URL or Your Computer IP  http.addHeader("Content-Type", "application/x-www-form-urlencoded"); //Specify content-type header    int httpCode = http.POST(postData); //Send the request  String payload = http.getString(); //Get the response payload    if (payload.substring(0, 6) == "del-id") {  String del\_id = payload.substring(6);  Serial.println(del\_id);  deleteFingerprint( del\_id.toInt() );  }    http.end(); //Close connection  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Delete Finpgerprint ID\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  uint8\_t deleteFingerprint( int id) {  uint8\_t p = -1;    p = finger.deleteModel(id);    if (p == FINGERPRINT\_OK) {  //Serial.println("Deleted!");  display.clearDisplay();  display.setTextSize(2); // Normal 2:2 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0,0); // Start at top-left corner  display.print(F("Deleted!\n"));  display.display();  } else if (p == FINGERPRINT\_PACKETRECIEVEERR) {  //Serial.println("Communication error");  display.clearDisplay();  display.setTextSize(1); // Normal 1:1 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0,0); // Start at top-left corner  display.print(F("Communication error!\n"));  display.display();  return p;  } else if (p == FINGERPRINT\_BADLOCATION) {  //Serial.println("Could not delete in that location");  display.clearDisplay();  display.setTextSize(1); // Normal 1:1 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0,0); // Start at top-left corner  display.print(F("Could not delete in that location!\n"));  display.display();  return p;  } else if (p == FINGERPRINT\_FLASHERR) {  //Serial.println("Error writing to flash");  display.clearDisplay();  display.setTextSize(1); // Normal 1:1 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0,0); // Start at top-left corner  display.print(F("Error writing to flash!\n"));  display.display();  return p;  } else {  //Serial.print("Unknown error: 0x"); Serial.println(p, HEX);  display.clearDisplay();  display.setTextSize(2); // Normal 2:2 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0,0); // Start at top-left corner  display.print(F("Unknown error:\n"));  display.display();  return p;  }  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Check if there a Fingerprint ID to add\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void ChecktoAddID(){    HTTPClient http; //Declare object of class HTTPClient  //Post Data  postData = "Get\_Fingerid=get\_id"; // Add the Fingerprint ID to the Post array in order to send it  // Post methode    http.begin(link); //initiate HTTP request, put your Website URL or Your Computer IP  http.addHeader("Content-Type", "application/x-www-form-urlencoded"); //Specify content-type header    int httpCode = http.POST(postData); //Send the request  String payload = http.getString(); //Get the response payload    if (payload.substring(0, 6) == "add-id") {  String add\_id = payload.substring(6);  Serial.println(add\_id);  id = add\_id.toInt();  getFingerprintEnroll();  }  http.end(); //Close connection  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enroll a Finpgerprint ID\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  uint8\_t getFingerprintEnroll() {    int p = -1;  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_scan\_bits, FinPr\_scan\_width, FinPr\_scan\_height, WHITE);  display.display();  while (p != FINGERPRINT\_OK) {  p = finger.getImage();  switch (p) {  case FINGERPRINT\_OK:  //Serial.println("Image taken");  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_valid\_bits, FinPr\_valid\_width, FinPr\_valid\_height, WHITE);  display.display();  break;  case FINGERPRINT\_NOFINGER:  //Serial.println(".");  display.setTextSize(1); // Normal 2:2 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0,0); // Start at top-left corner  display.print(F("scanning"));  display.display();  break;  case FINGERPRINT\_PACKETRECIEVEERR:  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_invalid\_bits, FinPr\_invalid\_width, FinPr\_invalid\_height, WHITE);  display.display();  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:  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_valid\_bits, FinPr\_valid\_width, FinPr\_valid\_height, WHITE);  display.display();  break;  case FINGERPRINT\_IMAGEMESS:  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_invalid\_bits, FinPr\_invalid\_width, FinPr\_invalid\_height, WHITE);  display.display();  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;  }  display.clearDisplay();  display.setTextSize(2); // Normal 2:2 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0,0); // Start at top-left corner  display.print(F("Remove"));  display.setCursor(0,20);  display.print(F("finger"));  display.display();  //Serial.println("Remove finger");  delay(2000);  p = 0;  while (p != FINGERPRINT\_NOFINGER) {  p = finger.getImage();  }  Serial.print("ID "); Serial.println(id);  p = -1;  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_scan\_bits, FinPr\_scan\_width, FinPr\_scan\_height, WHITE);  display.display();  while (p != FINGERPRINT\_OK) {  p = finger.getImage();  switch (p) {  case FINGERPRINT\_OK:  //Serial.println("Image taken");  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_valid\_bits, FinPr\_valid\_width, FinPr\_valid\_height, WHITE);  display.display();  break;  case FINGERPRINT\_NOFINGER:  //Serial.println(".");  display.setTextSize(1); // Normal 2:2 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0,0); // Start at top-left corner  display.print(F("scanning"));  display.display();  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");  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_valid\_bits, FinPr\_valid\_width, FinPr\_valid\_height, WHITE);  display.display();  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!  Serial.print("Creating model for #"); Serial.println(id);    p = finger.createModel();  if (p == FINGERPRINT\_OK) {  //Serial.println("Prints matched!");  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_valid\_bits, FinPr\_valid\_width, FinPr\_valid\_height, WHITE);  display.display();  } 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!");  display.clearDisplay();  display.drawBitmap( 34, 0, FinPr\_valid\_bits, FinPr\_valid\_width, FinPr\_valid\_height, WHITE);  display.display();  confirmAdding();  } 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 if there a Fingerprint ID to add\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void confirmAdding(){    HTTPClient http; //Declare object of class HTTPClient  //Post Data  postData = "confirm\_id=" + String(id); // Add the Fingerprint ID to the Post array in order to send it  // Post methode    http.begin(link); //initiate HTTP request, put your Website URL or Your Computer IP  http.addHeader("Content-Type", "application/x-www-form-urlencoded"); //Specify content-type header    int httpCode = http.POST(postData); //Send the request  String payload = http.getString(); //Get the response payload    display.clearDisplay();  display.setTextSize(1.5); // Normal 1:1 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0,0); // Start at top-left corner  display.print(payload);  display.display();  delay(1000);  Serial.println(payload);    http.end(); //Close connection  }  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*connect to the WiFi\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void connectToWiFi(){  WiFi.mode(WIFI\_OFF); //Prevents reconnection issue (taking too long to connect)  delay(1000);  WiFi.mode(WIFI\_STA);  Serial.print("Connecting to ");  Serial.println(ssid);  WiFi.begin(ssid, password);    display.clearDisplay();  display.setTextSize(1); // Normal 1:1 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(0, 0); // Start at top-left corner  display.print(F("Connecting to \n"));  display.setCursor(0, 50);  display.setTextSize(2);  display.print(ssid);  display.drawBitmap( 73, 10, Wifi\_start\_bits, Wifi\_start\_width, Wifi\_start\_height, WHITE);  display.display();    while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println("");  Serial.println("Connected");    display.clearDisplay();  display.setTextSize(2); // Normal 1:1 pixel scale  display.setTextColor(WHITE); // Draw white text  display.setCursor(8, 0); // Start at top-left corner  display.print(F("Connected \n"));  display.drawBitmap( 33, 15, Wifi\_connected\_bits, Wifi\_connected\_width, Wifi\_connected\_height, WHITE);  display.display();    Serial.print("IP address: ");  Serial.println(WiFi.localIP() |

Figure: the left-hand side column denotes the library imports and the display setup; the right-hand column contains the logic and the dead loop

|  |  |
| --- | --- |
| #include <NewPing.h>  #include <Adafruit\_MLX90614.h>  #define BUZZER\_PIN 5  #define CUT\_OFF\_TEMPERATURE 42 //This is temperature in degreeC  #define DISTANCE\_TO\_CHECK 10 //This is in cm  #define tempPin1 6 // logic that tells us whether  #define tempPin2 7 // logic  NewPing ultrasonicSensor(2,3,400);  Adafruit\_MLX90614 mlx = Adafruit\_MLX90614();  void setup()  {  lcd.begin(16, 2);  lcd.clear();  Serial.begin(9600);  analogWrite(BUZZER\_PIN, 0);  pinMode(tempPin1, OUTPUT);  pinMode(tempPin2, OUTPUT);  mlx.begin();  }  void tempReadTone()  {  for (int i = 0; i < 5; i++)  {  analogWrite(BUZZER\_PIN,100);  delay(100);  analogWrite(BUZZER\_PIN,0);  delay(100);  }  }  void tempHighTone()  {  for (int i = 0; i < 10; i++)  {  analogWrite(BUZZER\_PIN,100);  delay(100);  analogWrite(BUZZER\_PIN,0);  delay(100);  }  }  void tempOkTone()  {  analogWrite(BUZZER\_PIN, 100);  delay(2000);  analogWrite(BUZZER\_PIN, 0);  } | void loop()  {  String displayString;  lcd.setCursor(0, 0);  lcd.print("Check Temp");    int distance = ultrasonicSensor.ping\_cm();  Serial.println("Distance of object from sonar:");  Serial.println(distance);  if (distance > 6 && distance < DISTANCE\_TO\_CHECK)  {  tempReadTone();  delay(2000);  float sum = 0;  float temperature = 0;  int i = 0;  for(i=0;i<=100;i++){  temperature = mlx.readObjectTempC();  sum = sum + temperature;  Serial.println(temperature);  }  temperature = sum / i;  Serial.println("The average temperature is:");  Serial.println(temperature);  //temperature = mlx.readObjectTempC();  //Serial.println(temperature);  distance = ultrasonicSensor.ping\_cm();  if (distance > 6 && distance < DISTANCE\_TO\_CHECK && temperature <= CUT\_OFF\_TEMPERATURE)  {  tempOkTone();  delay(2000);  digitalWrite(tempPin1, HIGH);  digitalWrite(tempPin2, LOW);  }  else if (distance > 6 && distance < DISTANCE\_TO\_CHECK && temperature > CUT\_OFF\_TEMPERATURE){  tempHighTone();  delay(2000);  digitalWrite(tempPin1, LOW);  digitalWrite(tempPin2, HIGH);  }  else  {  delay(2000);  }  }  else{  delay(2000);  digitalWrite(tempPin1, HIGH);  digitalWrite(tempPin2, HIGH);    }  delay(5000);  } |

Figure: The code for the added temperature sensor module

# Implementation

## Description

## The PCB design has been implemented via proteus software. The circuit was not too complex and straightforward, but too many overlapping lines called for multilayer autorouting algorithmic solutions.

## The NodeMCU model was found, and incorporated, but many modules are difficult to obtain and at some times, not even seen in any library of proteus. To loop around these perils, the Conn-SIL has been used so that pin headers could be soldered later and used to connect to male pins from the different modules.

## The solution had both top layer of copper and bottom layer, crossing the height dimension using 4 via junctions. Some extra pins had been kept from NodeMCU so that more female headers could be soldered, if required (which turned out to be not required). The double layering and designing process usually takes 10 taka per square inch and some increased expense for added beautification.

## The NodeMCU sits in the middle, surrounded by OLED, fingerprint, Arduino junction and extended connection headers for the SONAR, temperature sensor and buzzer in the master sensor module. We kept 5V and GND pins (3 each) at hand, and 4 resistors provided an adequate potential divider voltage cut. All this has been put on resin, and soldered cautiously to avoid shorting, to finally create a working version

Diagram, schematic

Description automatically generated Diagram, schematic

Description automatically generated

Figure 2: PCB layout Bottom view and Top view respectively

## Results

## The results were exciting!

## A picture containing electronics, computer Description automatically generated

Figure: the bread board panel inside-> outside

A picture containing text, indoor, electronics

Description automatically generated

Figure: the sonar and the temperature sensor

A picture containing indoor, cluttered

Description automatically generated

Figure: The project as a whole

## The goals of the project were perfectly attained with all the parts working together perfectly.

## The simple procedure of use through which our result is best displayed:

## A teacher walks into a classroom. The teacher plugs in the device to their laptop. The device is powered. Once the students walk in, they will have to stand in front of the sonar/temperature sensor to verify the temperature. If they record normal temperature, upon placing their finger on the fingerprint sensor the attendance will be provided. This records the time, and ID to the pointed website specified by the instructor.

## Once the class is over the students leaves by providing their biometric one more time so that the instructor can know the duration of time they were in the class. Even if someone goes for a washroom break for example, they will have to place their finger on the system and again when they come back so that the perfect duration for which they were in the class can be recorder. (This may sound strenuous and so the instructor can decide to be lenient and excuse the student to a bathroom break without the information of how much time they were outside the class!)

## Graphical user interface, text, website Description automatically generated

## A screenshot of a computer Description automatically generated with medium confidenceFigure: website sample (only used for demonstration)

Figure: some of the additional features that the teacher can access. They can select dates to view and export attendance of their students.

Graphical user interface

Description automatically generatedFigure: The panel for adding new students

## GitHub Link

## YouTube Link

## <https://youtube.com/watch?v=eVSP1x1BxRs&feature=share>

# Design Analysis and Evaluation

## Novelty

## The basic biometric fingerprint attendance system is available on the internet. So, what is it that makes this one special?

## Firstly, we have used the Wi-Fi module in the NodeMCU, to upload our data to a local host. The data transfer is wireless and is non tamperable, which is paramount in cases that require transparency, like attendance systems. The data is easily accessed by the teacher only and downloaded in PC to keep soft copy of attendance without any sort of hassle.

## At recent times, we have heard time and again that we must learn to combat and live with recurring pandemics, especially coronavirus. So, to reduce the spread during spikes, it is of utmost importance to filter out healthy students from those who are potentially ill, and a quick test can be, of course, of body temperature; and this is a totally separate feature that our project encapsulates with the attendance logic system. This gives a better protection algorithm against pandemics that raise temperature, and though it may not be totally fault free, but it is a bold stand.

## Project Management and Cost Analysis

### Bill of Materials

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of component** | **Description** | **quantity** | **Price** |
| NodeMCU | ESP8266-12E Board | 1 | 220 |
| Fingerprint Sensor | R305/R307 Module | 1 | 1100 |
| Sonar | HCSR04 | 1 | 90 |
| OLED Display | 0.96" SSD1306 I2C OLED Display | 1 | 250 |
| Connecting wires | --------------- | -------- |  |
| Arduino UNO | SNDedition | 1 | 650 |
| Temperature sensor | MLX 90614 | 1 | 1080 |
| Buzzer | -------------- | 1 | 40 |
| Commutation and other | -------------- | --------- | 500 |

### Calculation of Per Unit Cost of Prototype

Per Unit cost of the device will be around: TK 3,930

However, if more units are made the fixed cost will be spread out and the cost will be much lower!

### Calculation of Per Unit Cost of Mass-Produced Unit

Since the commutation and miscellaneous costs will not increase with units, they are roughly fixed costs:

Roughly the variable cost for 100 units will be: 100\*(3,430) = Tk 343,000

The fixed cost (assuming fixed until major increase in costing is to be done) = Tk 500

Per unit cost if around a hundred units are produced: Tk 3,434.

This is an overstated amount as the discounts upon bulk purchases of raw materials from suppliers are not considered. Thus, the cost per unit will be under Tk. 3,434.

\*\*Note that this does not include the cost of a domain name/website which the user must have of their own

### Timeline of Project Implementation

The table below shows some of the important dates that marked some milestones in our project:

Table

Description automatically generated

## Practical Considerations of the Design to Address Public Health and Safety, Environment, Cultural, and Societal Needs

### Considerations to public health and safety

The whole framework of TruID is run by only 5 volts taken via the USB port from the user computer. It doesn’t draw any current. Thus, the elements are nor hazardous to human health and is fully safe if the gadgets are used by taking appropriate measures.

### Considerations to environment

Our project has no issues with environmental pollution upon use. However, we understand that the use of non-recycled plastic is bad for the environment. Some of our project components did contain plastics whose sources are unknown. These parts were unavoidable and hence were used. The parts where we did have control, such as the outer structure of the device, we used biodegradable cardboard and wooden plates which are more environment friendly and have no negative impacts on human life.

### Considerations to cultural and societal needs

The idea of class attendance has been culturally an area of mischief for the students and a source of mistrust that can hamper a healthy teacher-student relationship. In the four years of undergad, and 12 years of school we have often seen some untoward and unnecessary problems due to mischievous intents. This not only hampers the student-pupil relationships but may leave either party disinterested in the course/class after an unfortunate event. Our project makes this factor redundant by giving a simple work around!

## Assessment of the Impact of the Project on Societal, Health, Safety, Legal and Cultural Issues

### Assessment of Societal Issues

Assessment of Societal Issues. The overall maintenance of the school attendance system will be improved. Students will now be held accountable for their mischief in dodging attendances and will be penalized since their digital signature will be recorded. This will help combat children indiscipline as one of the first forms of indiscipline is a student wrongfully giving his/her attendance.

### Assessment of Health and Safety Issues

Ill students can be isolated easily by preventing them from registering their attendance and thus, they will have no choice but to contact their respective class teachers for further action. This is important in health monitoring and classroom health management for students.

### Assessment of Legal Issues

We emulated code from various sources. This might cause some consideration from copyright, though we tried to innovate the code properly with our own ideas and implementations.

### Assessment of Cultural Issues

The project can be used in many official settings where many staff members work on a common theme/goal in an institute. These institutes may keep track of their staff attendance by use of our project. The same can be said about political meetings/discussions, societal projects, non-profit work etc.

## Evaluation of the Sustainability and Impact of the Designed Solution in the Societal and Environmental Contexts

### Evaluation of Sustainability

The project draws a lot less power and is very stable and deterministic. Simple parts and modules are there that are readily produced in large scale, and thus should be sustainable and replaceable if need be.

### Evaluation of Impact of Design in Societal Context

The possible implications of the project in societal context have already been looked into and proper evaluations can already be found in smart classrooms in developed countries who emulated the project in real life

### 

### Evaluation of Impact of Design in Environmental Context

The project is deemed to be very environmentally friendly as it is contained, sustainable, and puts very low pressure on environment. There are no drawbacks like carbon emission and warming effects, etc.

# Reflection on Individual and Teamwork

## Individual Contribution of Each Member

## The project was done with the efforts off all the team-mates. The responsibilities varied throughout the journey. At the starting we met twice a week to discuss ideas and talked about how the end result of the project was supposed to look like.

## Next, we looked at repositories and tutorials to learn more about fingerprint sensors and how to incorporate them into larger circuits.

## ID 47 and 48, tried to understand the codes of that were found in the professional repositories. ID 34 and 46 tried to put together the hardware of the circuits. Often things went south in both departments, and everyone would help to debug once explained what the problem at hand was.

## ID 34 learnt and design the PCB once the hardware was complete. Others incorporated the novelties that was decided to be added. (The local host setup for IOT based system and the temperature sensor).

## Lastly when all the things were complete, the external structure of the device was made by ID 34, 46, 48. ID 47, due to being diagnosed with COVID, isolated home this week and started to prepare the project report and slide

## Mode of Teamwork

## There are many ways a team can work. We, the members of the team have known and worked with each other for quite some time now. This helped us to move forward without much friction. Our team transitioned between the 3 of the 4 modes seamlessly. Initially, when deciding the end result and discussing novelties our team was in COULD DO mode. Once the GitHub repositories and tutorials were gathered we moved towards the SHOULD DO mode. Finally, as the problems and bugs were starting to be fixed, we moved towards the DOING mode.

## Our team never felt the need of a clear leader with everyone respecting each other opinion and skills. Thus, the WILL DO mode was never needed. Diagram Description automatically generatedFig: The 4 modes of Teamwork

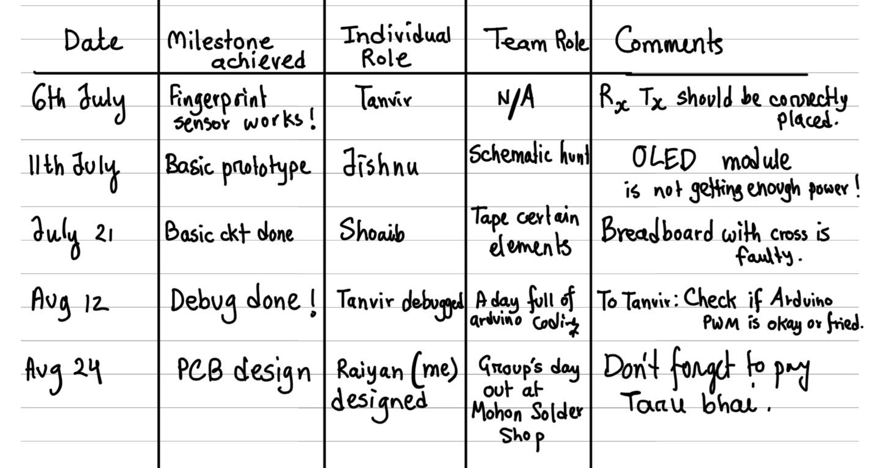
## Diversity Statement of Team

## Our team was a true representation of diversity, with different educational board backgrounds. Two of us were from an IGCSE Edexcel educational background whereas the others were from English version and Bangla medium background.

## Geographically, our teammates originated from the different areas of Bangladesh, ranging from Pabna, to Dhaka to Rajshahi to Kurigram. One of our mates is now located at the dormitory of BUET the others live in 3 corners of Dhaka city.

## Logbook of Project Implementation

## Some basic recollection of events that took place!



# References

https://github.com/adafruit/Adafruit-GFX-Library

//https://github.com/adafruit/Adafruit\_SSD1306

//https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library

https://www.youtube.com/watch?v=DZHhMR3ViFg&t=548s&ab\_channel=InnovateYourself

https://www.apachefriends.org/download.html

https://how2electronics.com/fingerprint-biometric-attendance-system-arduino/

https://www.youtube.com/watch?v=PSo3m7euzo0&list=PL\_zvrXFdKgZpCgI1ZgfbJUcuGTKLL8pMT