Non-Contact Temperature Scanner

Touch-Free Thermo-sensor

Team 6

Maria Victoria Gromysh, Timothy Englehart, Samantha Arroyo Villanueva, Dhruv Rengarajan

Table of Contents

Project Description

The non-contact temperature sensor is an accurate system which is intended to perform the following tasks:

- Detect human presence and movement.
- Audible voice announcement alerting the user providing the instructions, and temperature measurement results.
- Capture user body temperature, and verify it exceeds 100.4°F with ± 1°F accuracy.
- Display the ambient temperature, user count, user body temperature with pass/fail result criteria.
- Record the user data with timestamp in a SD storage in csv/txt format.

Key technologies used in this project are motion detection sensor unit based on thermal imaging, accurate IR temperature sensor, speaker system with audio driver, touch color graphics tft display integrated all integrated with ATmega328P - 8-bit AVR Microcontrollers.

Writing touch display firmware driver, SD storage data logging and integrating all sensor module codes were the challenging parts of this project and considerable accomplishments.

The area of future development will be designing the system to detect human body temperature at a larger distance (greater than 6ft) but with ± 1°F accuracy.

Product Electrical Specification

- 5V, 0.5A/2A External Power Supply Required.
- Motion sensing range less than 120 degree, within 7 meters.
- Non-contact temperature sensing with +/- 0.5deg C accuracy.
- 3.5inch TFT LCD Screen Module 480x320 Resolution HD with touch
- Upto 4 GB external storage for data logging

System Overview

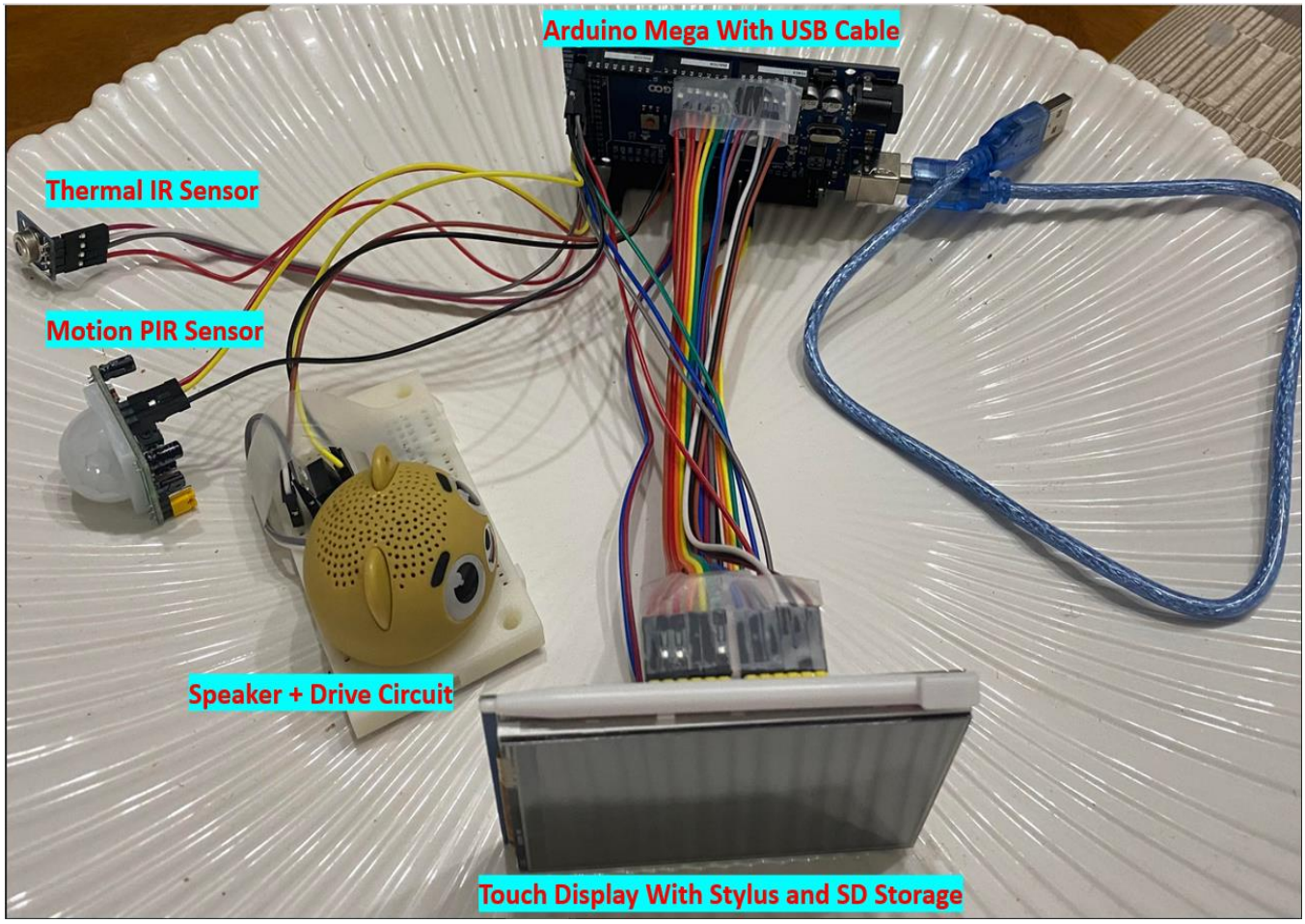


Figure 1 System Setup with all the modules

The system comprises of the following parts:

- ❖ Arduino mega 2560 with USB cable / power adapter
- ❖ Audio driver with speaker
- ❖ TFT touch display with micro-SD card slot
- ❖ IR thermal sensor
- ❖ Motion PIR sensor

Connect all the modules referring the pin connections like shown in the attached photo.

Module Connections

- ➢ Display + Micro SD Card: 24 pin connection to Arduino -> 20 signals + 2 power rails (3.3V and 5V) + 2 GND
- ➢ PIR Motion Sensor: 3 pin connection to Arduino -> 1 power rail (5V) + 1 Signal + 1 GND
- ➢ IR Thermal Sensor: 4 pin connection to Arduino -> 2 signals (SDA & SCL) + 1 power rail (5V) + 1 GND
- ➢ Speaker: 3 pin connections to Arduino -> 1 signal (PWM) + 1 power rail (5V) + 1 GND. A BJT may be required to add with speaker for high current gain.

Hardware Requirements Specifications

Table 1 HRS

SI.No	Hardware Blocks	Requirements	Definition
1	Power Supply	DC Voltage/Wall Outlet	Converts external voltage to DC and supplies power to all required interfaces
2	Enclosure	Fiber/Glass Structure	Physical casing that contains the system inside
3	Temperature Sensor	I2C Communication	Measure's temperature of nearby body
4	Motion Sensor	Digital pulse output	Determines when an object is within the minimum distance to begin measurement
5	Microcontroller	8 bit bus width	Processes incoming data and sends it out to appropriate
6	Display	3.5" TFT LCD 480x320 Screen	User interface
7	SD Storage	Micro SD Card Up to 4GB	Logs user information for the past 24 hours
8	Speaker	Audio Signal @ 1000 Hz	Announcement for user notification

Software Requirements Specifications

1. Sense user movement and notify the user with instruction to stand near the target as a voice announcement.
2. Capture, display, and record user temperature.
3. Check the user temperature >100.4 F; if so, fail else pass.
4. Store the user count, timestamp, target temperature as data logs in the SD card.
5. Provide a button to view the historical data logs with scroll feature.
6. Provide an exit button to return to the home screen.

Hardware Section

Hardware Block Diagram

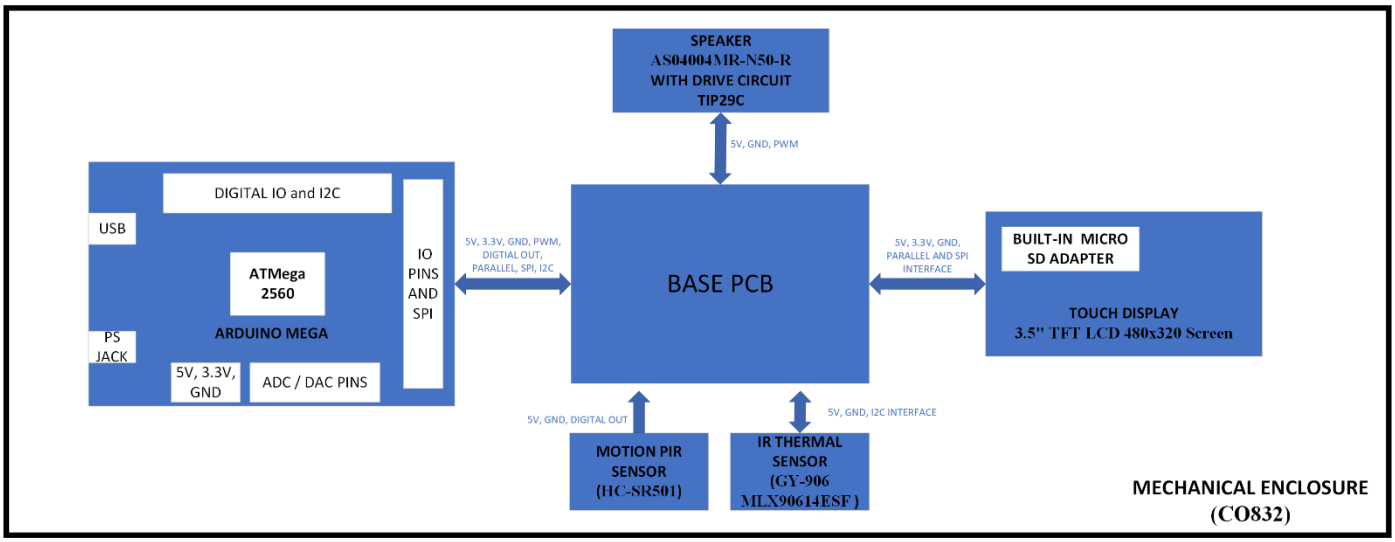


Figure 2 : Hardware block diagram with all the interfaces and power supplies in the enclosure.

Parts/Modules Detail

Arduino Mega 2560 Microcontroller

Description: The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, 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.

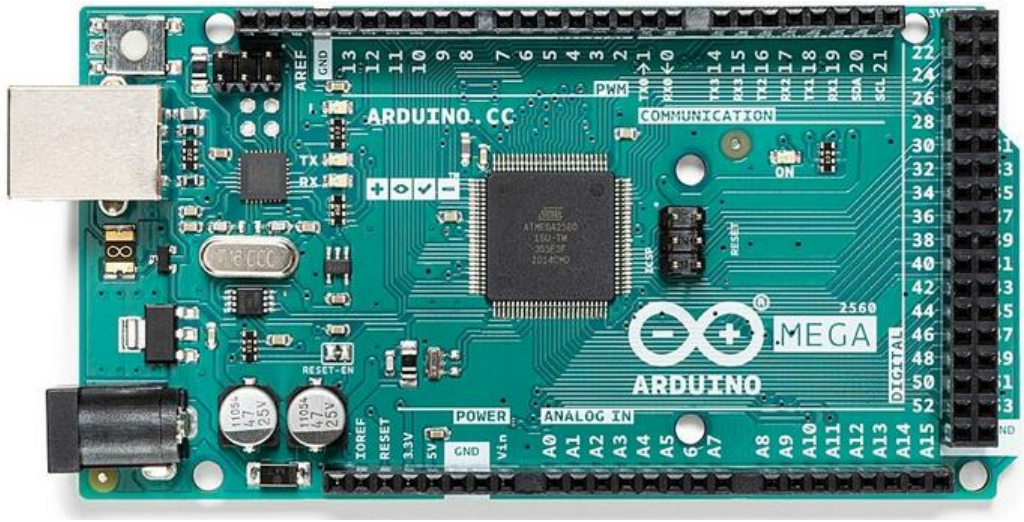


Figure 3 Arduino mega used as the computer controller unit with ATMEL Microcontroller.

Refer Appendix 1 for datasheet.

Audio System

Description: The simplest way to generate an audio signal to play on the speaker is to use a hardware PWM output. Set the PWM period to 1/frequency of the desired sound. The PWM duty cycle is set to 0.5. A lower duty cycle setting produces lower volume. The advantage of using the PWM hardware is that it takes minimal memory and no processor time to output an audio tone. Refer Appendix 3 for speaker and TIP29C datasheet.

Figure 4: Speaker component used for the speaker.

SD Card Storage

A micro-SD card up to (4GB) can be externally plugged into the card adapter inside the system for data.

Figure 5 Sd card inside the display module.

Motion PIR Sensor

Description: PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out.

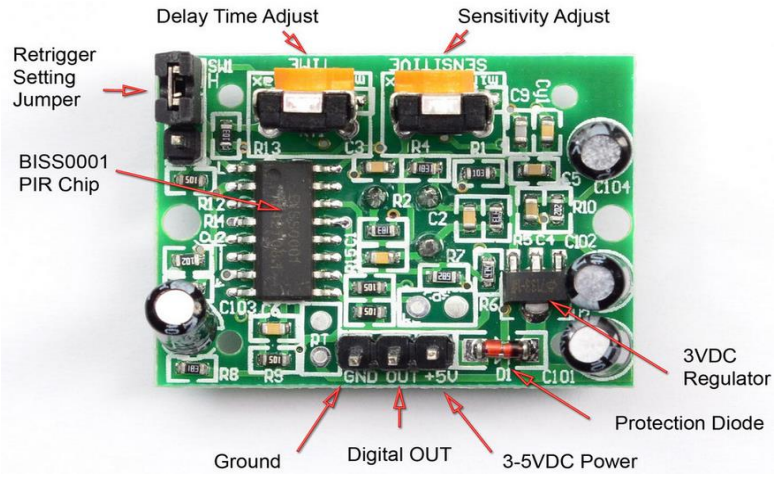


Figure 6 PIR sensor module board, bottom side

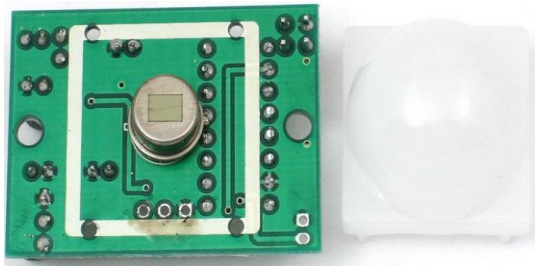


Figure 7 Top side of PIR module

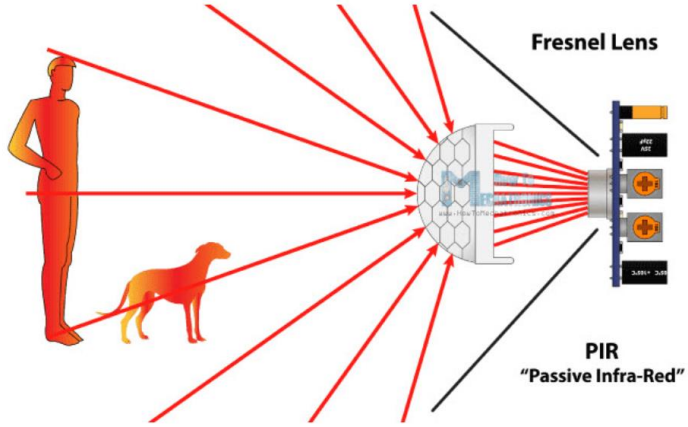


Figure 8 PIR sensor detection range

Working Principle: The PIR sensor (passive infrared sensor) detects changes in the amount of infrared radiation impinging upon it, which varies depending on the temperature and surface characteristics of the objects in front of the sensor. When an object, such as a human, passes in front of the background, such as a wall, the temperature at that point in the sensor's field of view will rise from room temperature to body temperature, and then back again. The sensor converts

the resulting change in the incoming infrared radiation into a change in the output voltage, and this triggers the detection. Objects of similar temperature but different surface characteristics may also have a different infrared emission pattern, and thus moving them with respect to the background may trigger the detector as well.

Thermal IR Sensor

Description: The GY-906 MLX90614 is a high precision infrared non-contact thermometer module with I2C interface and 5V or 3.3V operation.

- Non-contact measurement perfect for measuring temperature of moving objects.
- Temperature measurement range: -70°C to +380°C
- I2C/SMBus interface
- Optional PWM and interrupt output
- 3.3V or 5V operation

Working Principle:

The primary interface to the device is the SMBus which is basically the same as I2C and uses the same SDA (Data) and SCL (Clock) lines. The module has 4.7K pull-up resistors on these lines. The **SCL** and **SDA** pins connect to the SCL and SDA pins on the MCU.

The temperature of the object as well as the ambient temperature of MLX90614 can be read in °C and °F. The device uses a high resolution 17-bit ADC and DSP unit which provides a large range, high resolution and very good accuracy.

In addition the device can be programmed over the interface to have one of 127 I2C addresses so that up to 127 of the devices can operate on one I2C bus. A number of other settings can also be configured using the I2C interface such as using the PWM output or thermal switch mode of operation.

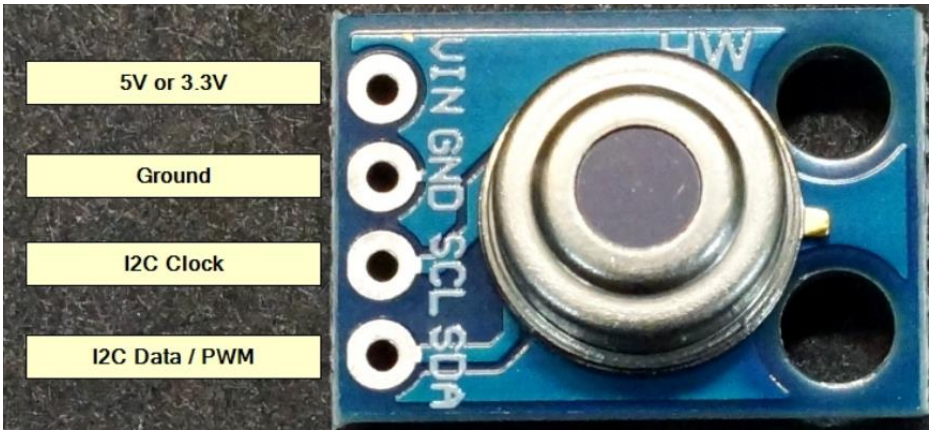


Figure 9 Temperature Sensor module board with pin connections

Base Board

A base board is designed to integrate all the modules such as IR sensor, motion sensor, Speaker, TFT touch display with the Arduino Mega board.

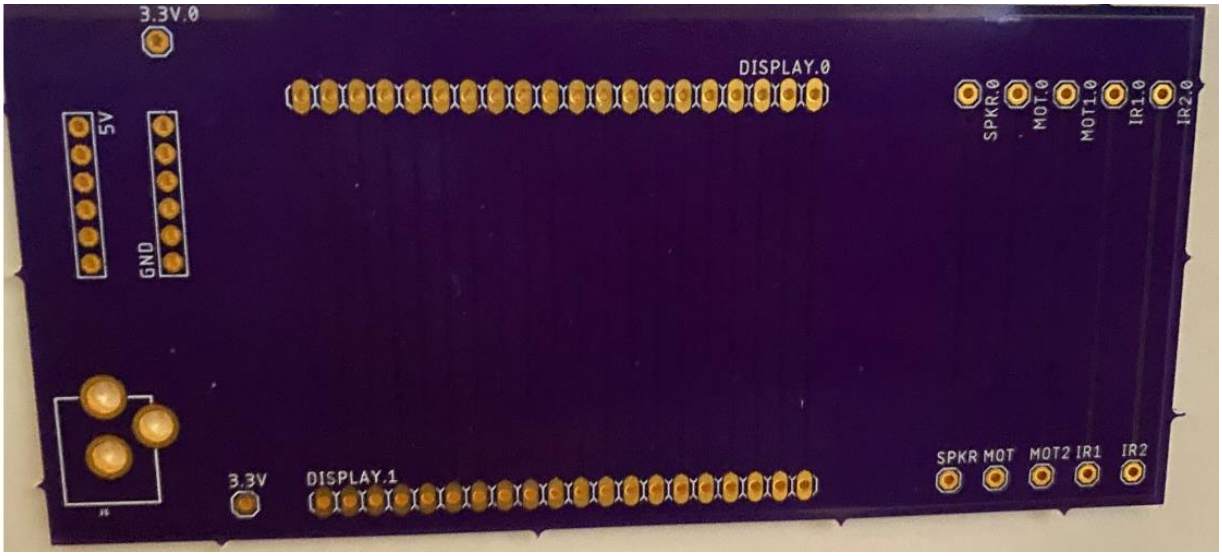


Figure 10 Top Layer of Base PCB

Figure 11 Bottom Layer of PCB board

TFT Touch Display

Description:

The 3.5-inch TFT LCD module with 320x480 resolution and 65K color display. It uses 8-bit line parallel port communication, and the driver IC is ILI9486.The module includes an LCD display, 5V~3.3V level conversion circuit, which can be directly plugged into the Arduino UNO and MEGA2560 development boards, and also supports SD card expansion function.

- Supports development boards such as Arduino UNO and Mega2560 for plug-in use without wiring.
- 480X320 resolution, clear display, support for touch function.
- Support 16-bit RGB 65K color display, display rich colors.
- 8-bit parallel bus, faster than serial SPI refresh.
- On-board 5V/3.3V level shifting IC, compatible with 5V/3.3V operating voltage.
- Easy to expand the experiment with SD card slot.
- Provides an Arduino library with a rich sample program.

Refer Appendix 2 for datasheet.

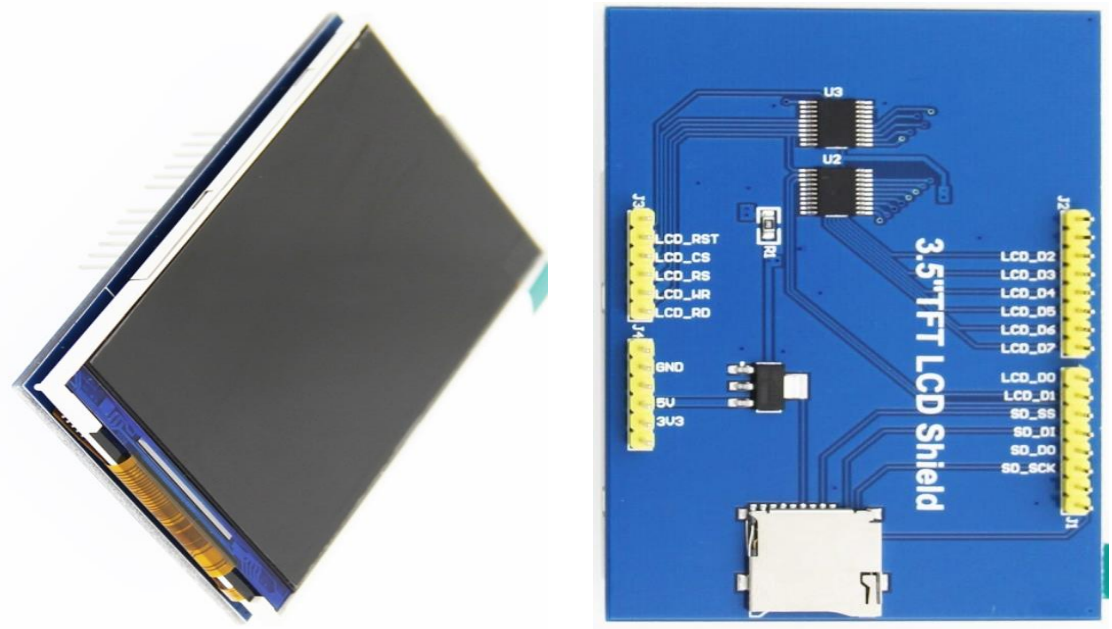


Figure 12 Top and bottom side of the TFT display module.

Schematics

Touch Display Circuit

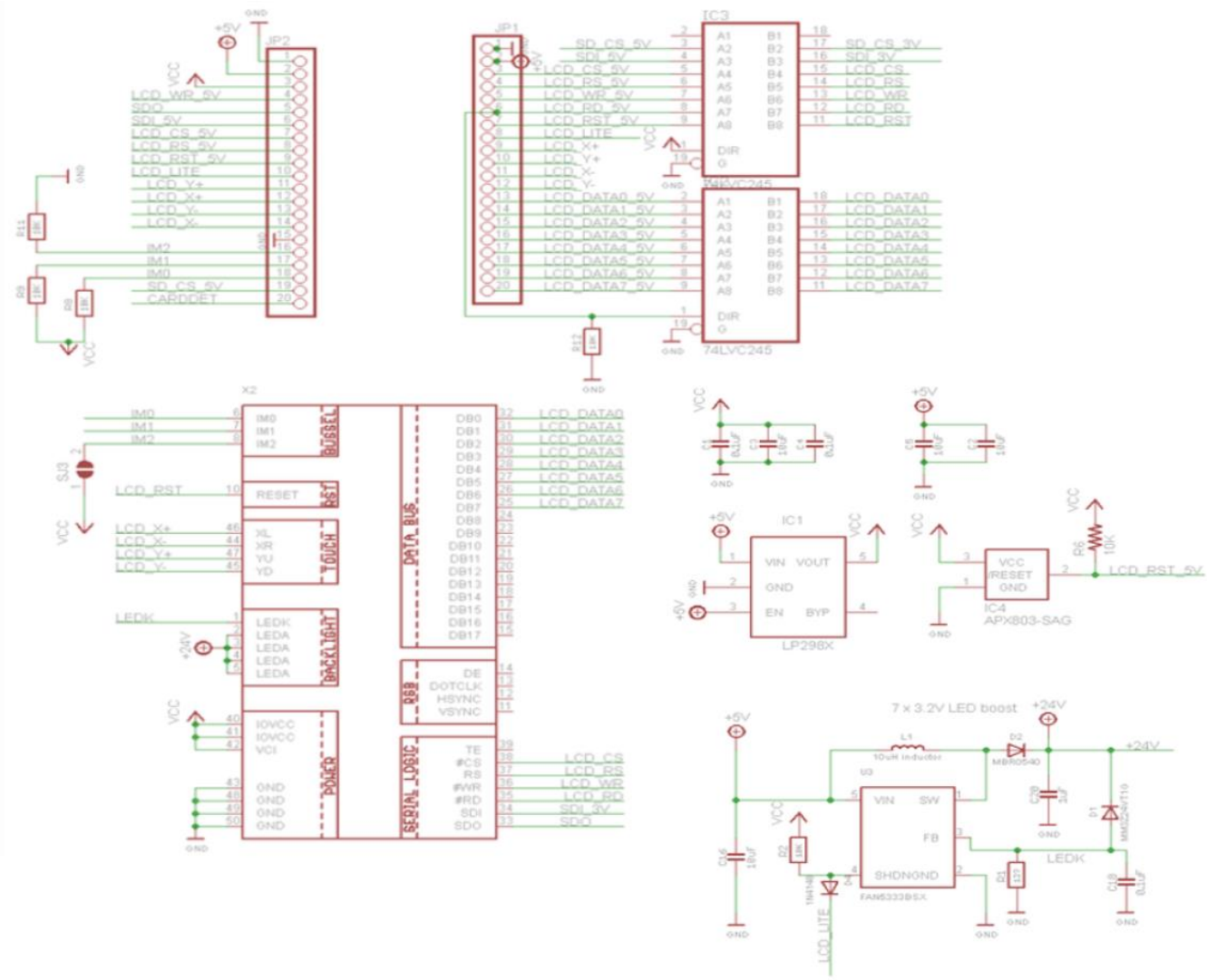


Figure 13 TFT Touch Display Interface and Power Supply Circuit

ATmega2560 Microcontroller Circuit

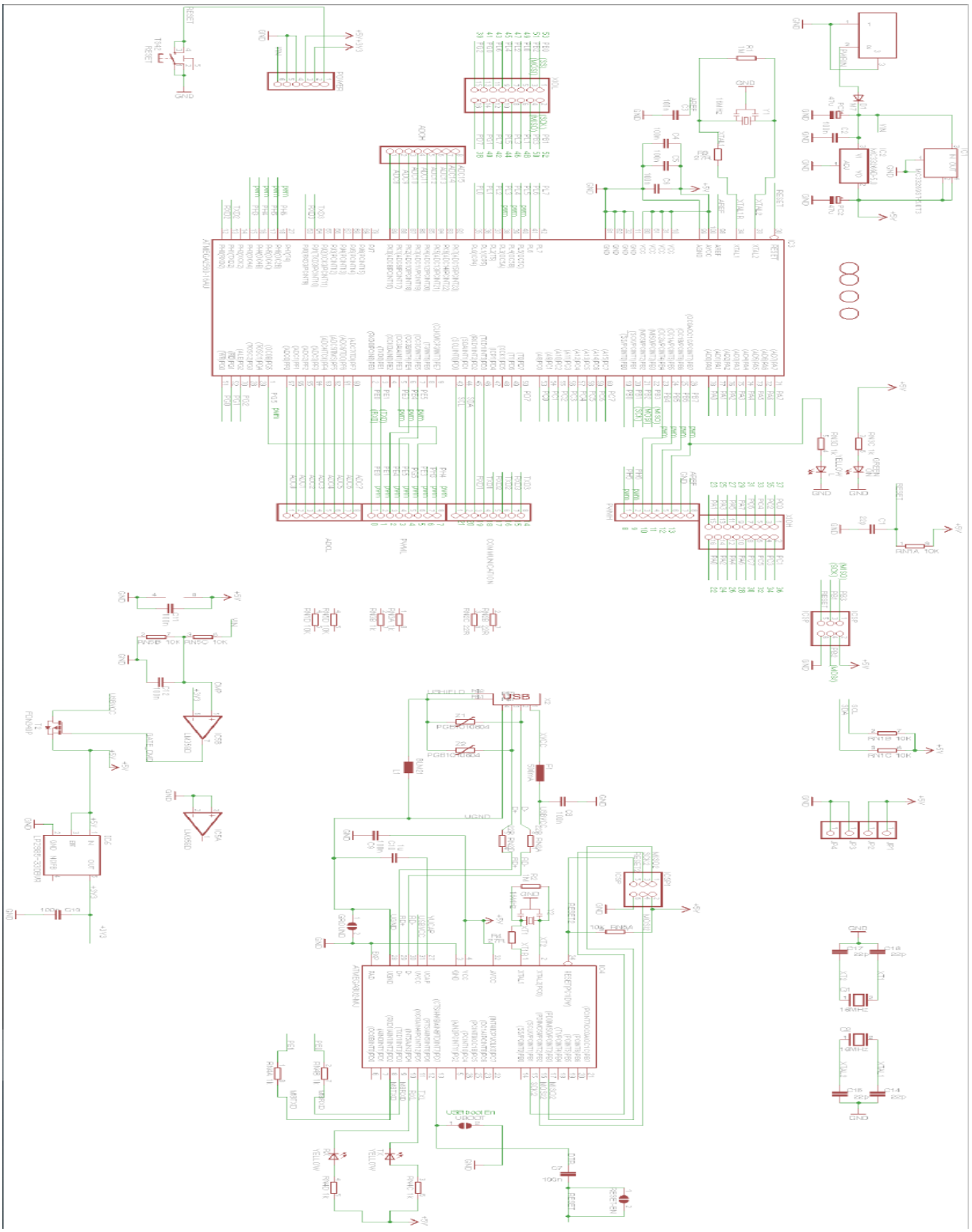


Figure 14 Microcontroller schematic

Motion Sensor Circuit

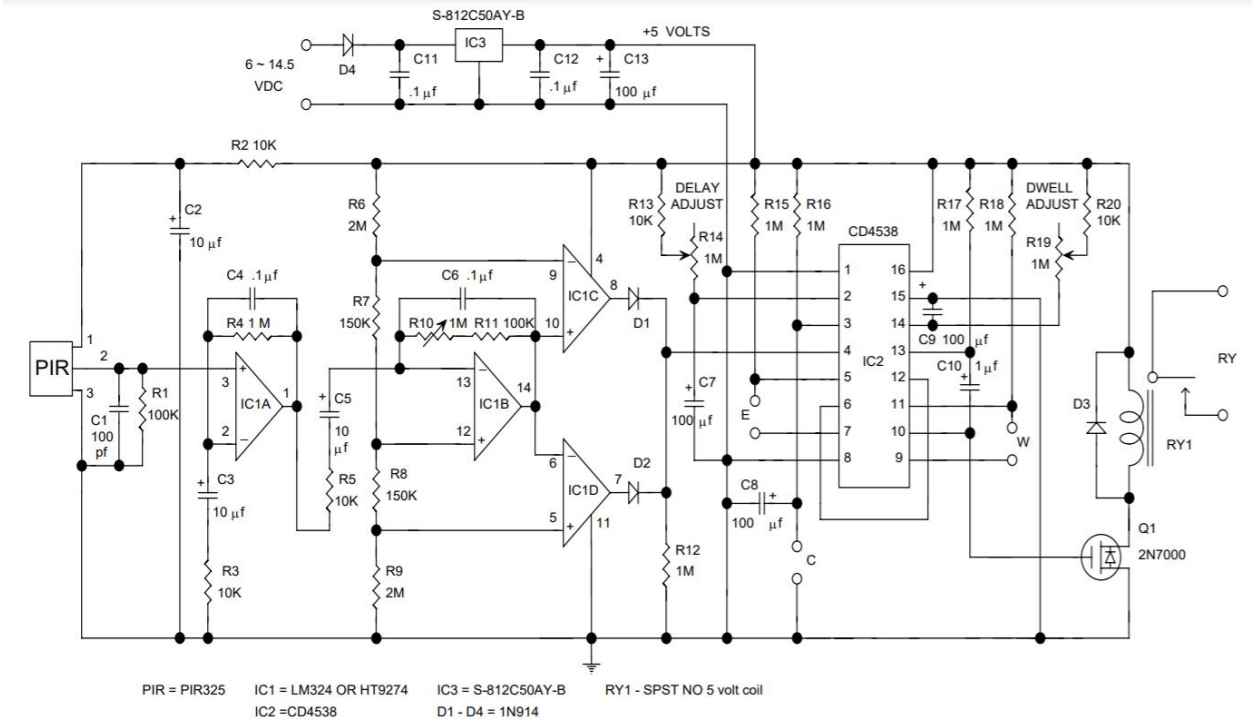


Figure 15 Schematic of motion sensor module

IR Temperature Sensor Circuit

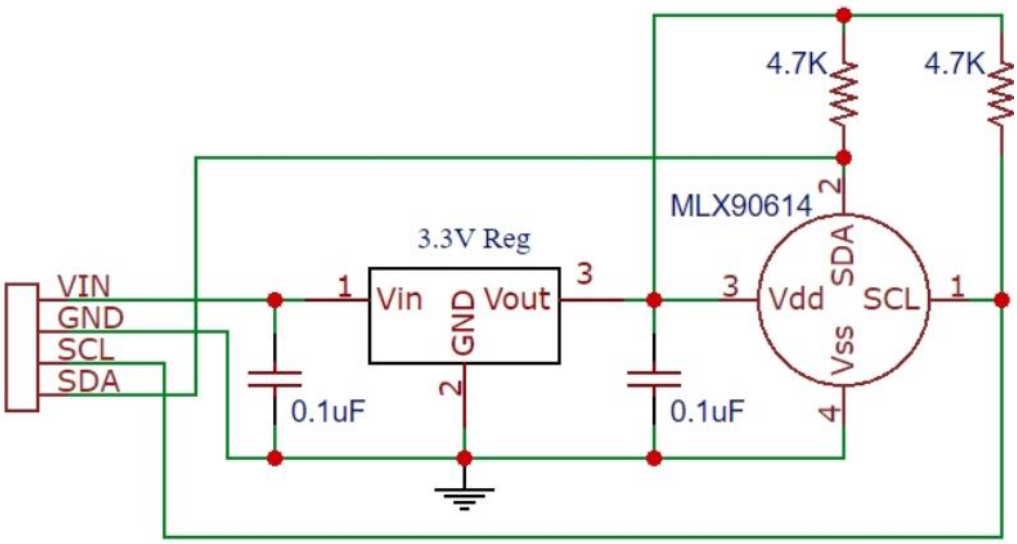


Figure 16 Schematic of the IR temperature module

Micro SD Card Adapter Interface Circuit

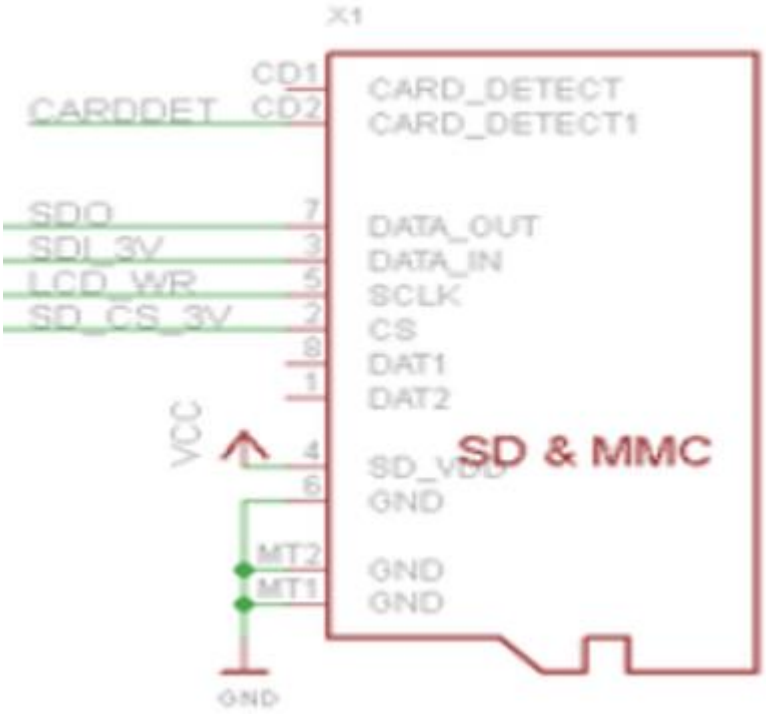


Figure 17 SD card Schematic

Audio Driver Circuit

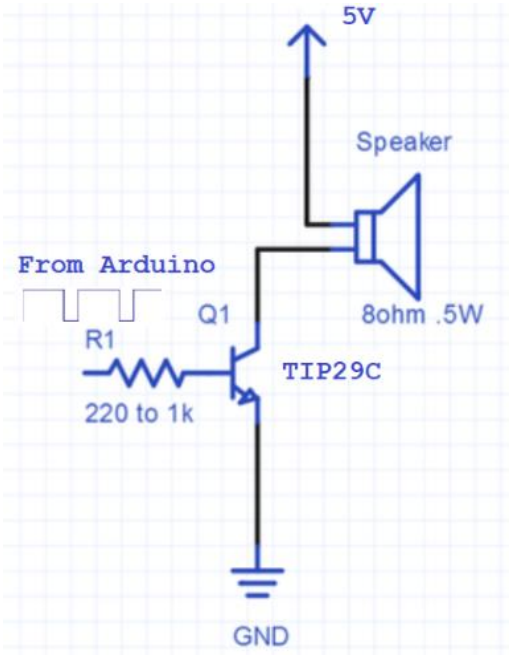


Figure 18 Schematic of the Speaker module

Base Board Circuit

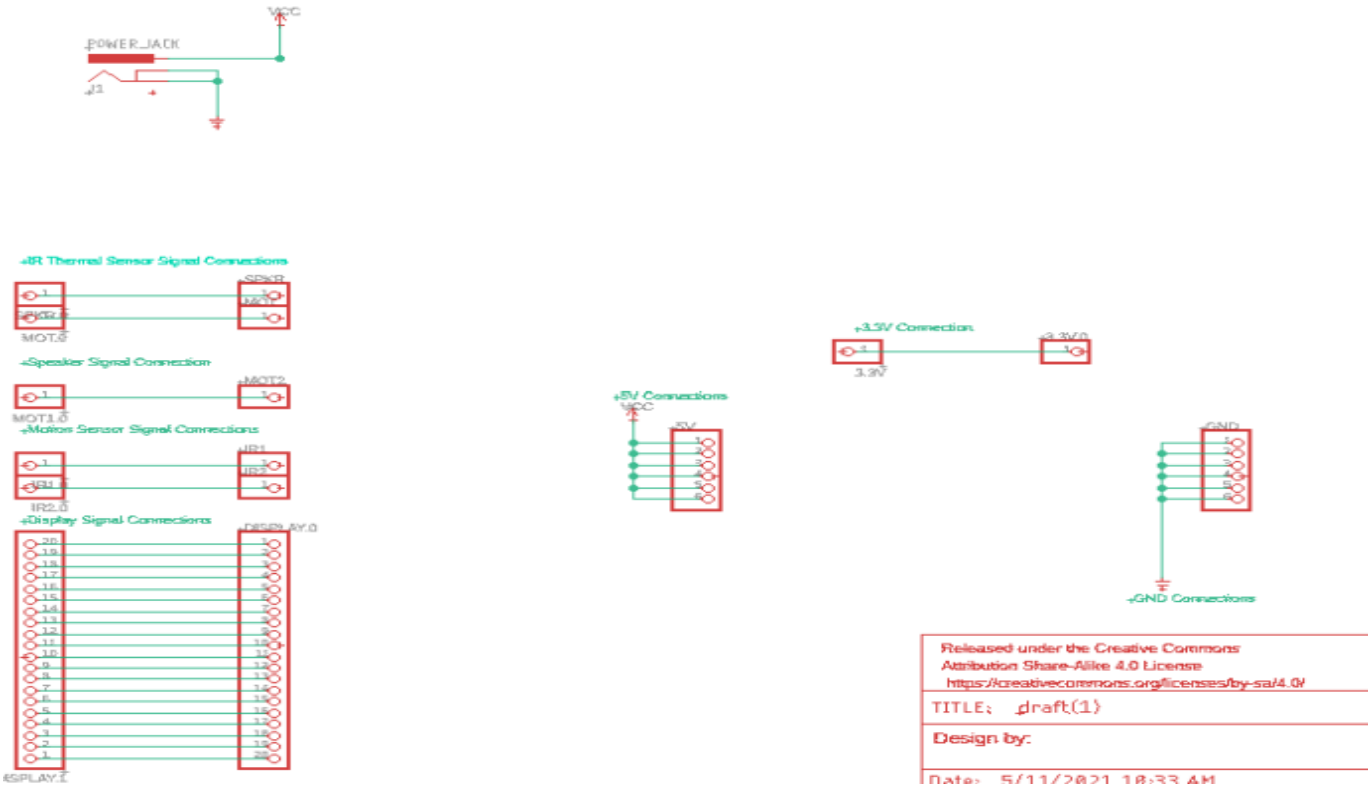


Figure 19 Base board to interface all the modules and Arduino

PCB Layout

A two layer 4.5” x 2.5” PCB designed to integrate all the modules such as IR sensor, motion sensor, Speaker, TFT touch display with the Arduino Mega board.

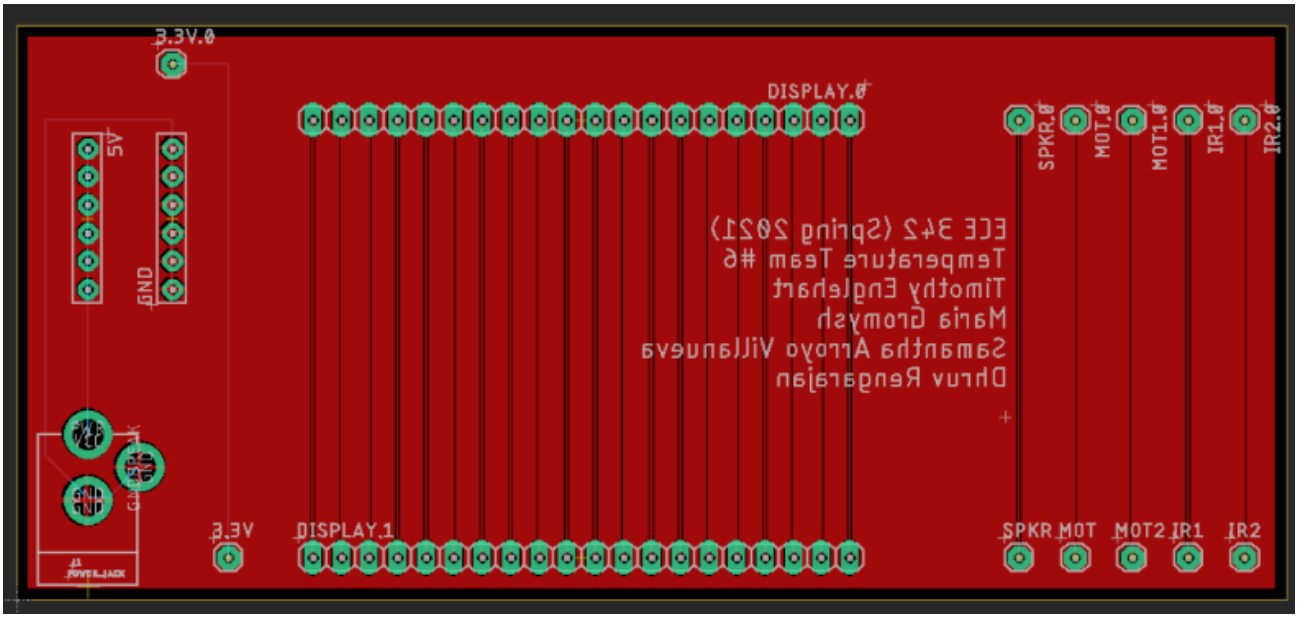


Figure 20 Top layer of PCB

Figure 21 Bottom layer of PCB

Mechanical Enclosure:

Drawing:

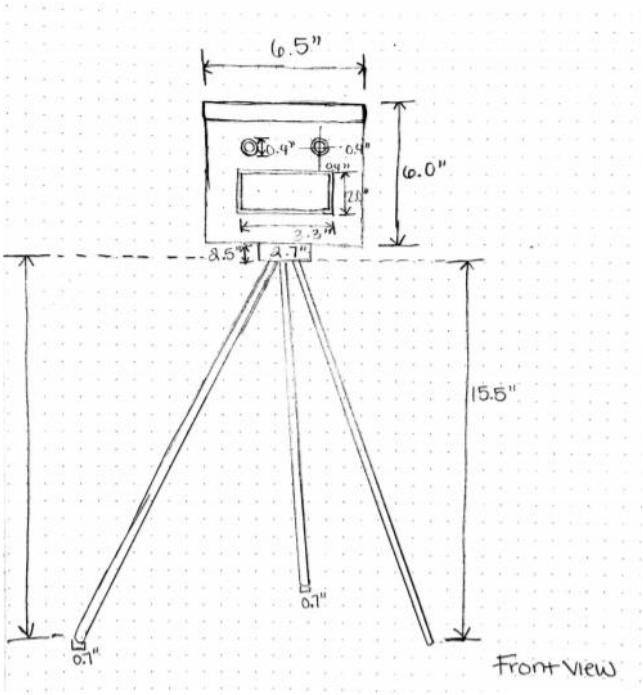


Figure 22 Front View of enclosure

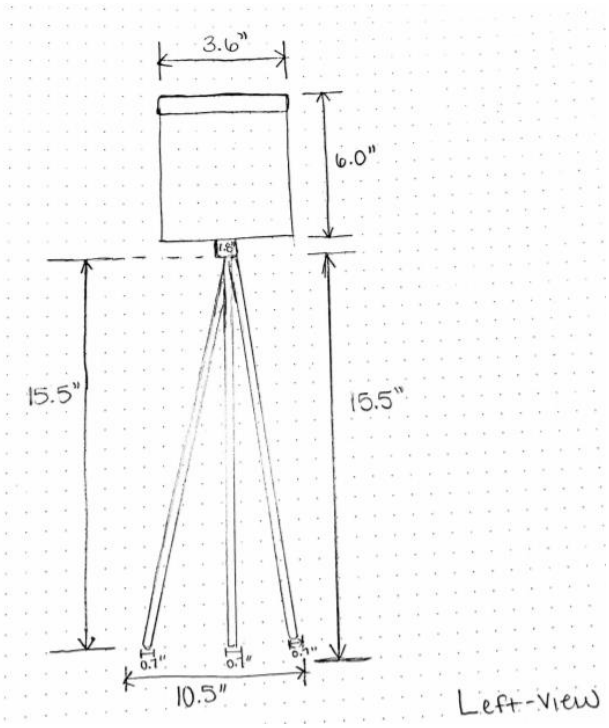


Figure 23 Left view of the enclosure

Datasheet: Refer appendix 4.

Bill of Materials (BOM)

Item	Description	Quantity	Cost Per Unit	Total Cost
HC-SR501 PIR Sensor	Motion Sensor Module	1	10.99 USD	10.99 USD
Arduino Mega	Microcontroller Module	1	17.47 USD	17.47 USD
3.5" TFT LCD 480x320 Screen	Touch Display Module + Micro SD Card Adapter	1	19.49 USD	19.49 USD
Transcend 2 GB SD Flash	Memory Card	1	7.98 USD	7.98 USD
GY-906 MLX90614ESF Infrared	Temperature Sensor Module	1	15.16 USD	15.16 USD
CSS-66668N	8 Ohms General Purpose Speaker 3 W 160 Hz ~ 15.0 kHz Top Round, Square Frame	1	3.87 USD	7.04 USD
PCB	Base printed circuit board	1	4.99 USD	4.99 USD
TIP29C	NPN power transistors	1	0.88 USD	0.88 USD
Jumper wires	Male to Female connector	36	0.0582 USD	2.094 USD
Enclosure and tripod	Made with sheet metal and welded	1	24.09 USD	24.09 USD
TOTAL COST				110.19 USD

Note: The above BoM is based on purchase quantities used for prototype. The total cost will come down in volume manufacturing.

Software Section

Firmware Flow Chart

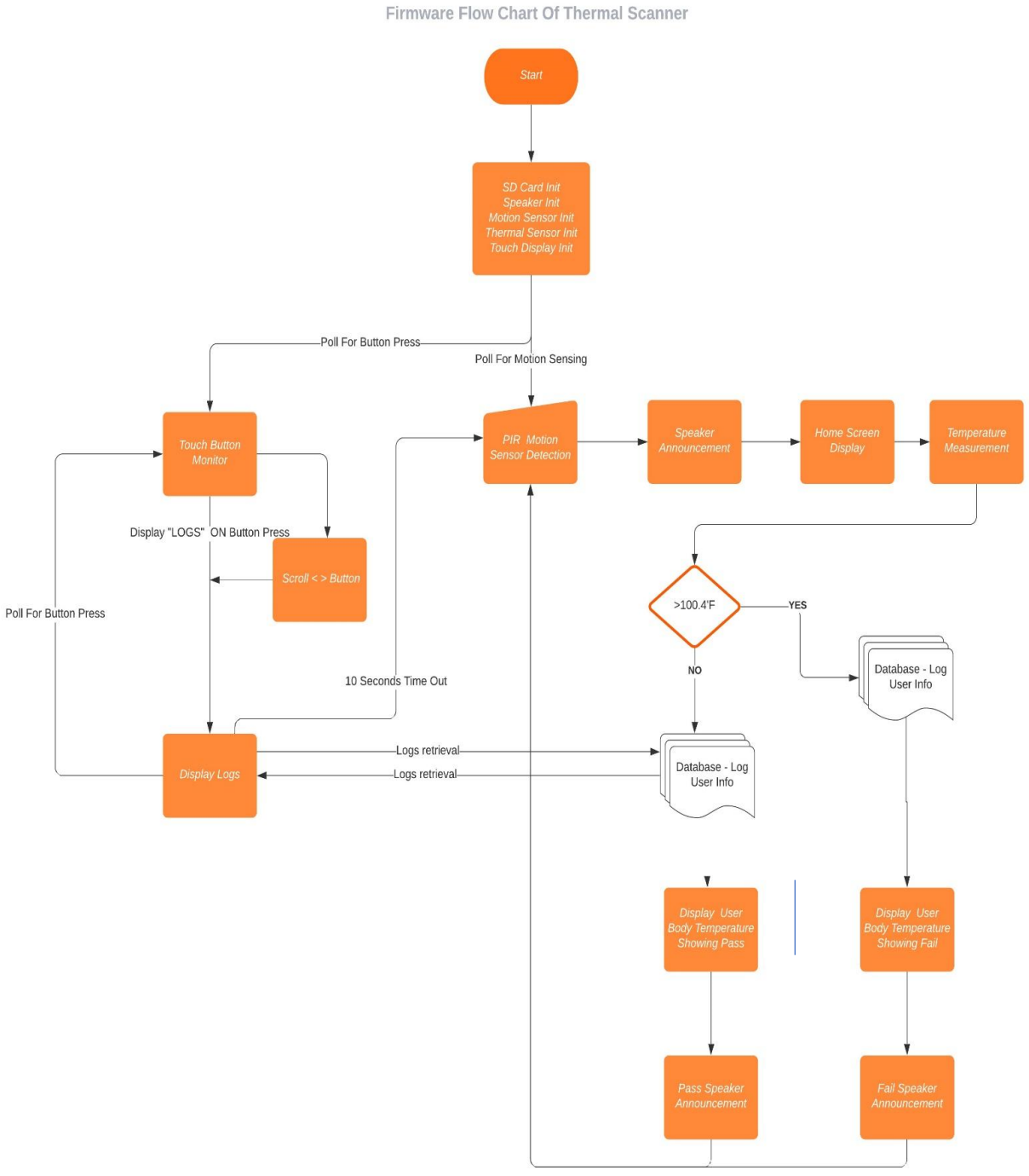


Figure 24 Flowchart of the firmware design

Full Integrated Code: Thermal_Scanner_Code_RevAs.INO

```
#include <Wire.h>
#include <display_init.h>
#include <PIR_Sensor_Init.h>
#include <Speaker_Init.h>
#include <Temp_IR_Init.h>
int count=0;
bool log_stat = false;
//************************************SD Card Setup**********************************
void sdcard_setup()
{
//  Serial.begin(9600);
  if(!SD.begin(SD_ChipSelectPin))
  {
    Serial.println("SD Card Init failed");
    return;
  }
  Serial.println("SD Card Init Done.");
  File dataFile = SD.open("datalog.txt",   FILE_WRITE);
  // if the file is available, write to it:
  if (dataFile) {
    dataFile.close();
    Serial.println("Log File Detected!");
    // print to the serial port too:
```

```
  }
}

//************************************Display
Section*********************************

void showmsgXY(int x, int y, int sz, const GFXfont *f, String msg, const char *color)
{
    int16_t x1, y1;
    uint16_t wid, ht;
    tft.setFont(f);
    tft.setCursor(x, y);
    tft.setTextColor(color);
    tft.setTextSize(sz);
    tft.print(msg);
    Serial.println(msg);
//    delay(1000);
}



void log_screen()
{
    showmsgXY(70, 20, 1, &FreeSans9pt7b, msg[0],WHITE); //Project Name
    showmsgXY(190, 40, 1, &FreeSans9pt7b, msg[1],WHITE);// Team Name
    tft.drawFastHLine(0, 44, tft.width(), WHITE); // Line
    left_btn.initButton(&tft,  160, 280,  60,40, BLACK, CYAN, WHITE, "<", 1);
```

```
    right_btn.initButton(&tft, 240,  280, 60,40, BLACK, CYAN, WHITE, ">", 1);
    exit_btn.initButton(&tft,  320, 280,  60,40, BLACK, CYAN, WHITE, "EXIT", 1);
    left_btn.drawButton(false);
    right_btn.drawButton(false);s
    exit_btn.drawButton(false);
    Serial.println("Log Screen Displayed");

}

void home_screen()
{
    tft.fillScreen(WHITE);
    showmsgXY(70, 20, 1, &FreeSans9pt7b, msg[0],BLACK); //Project Name
    showmsgXY(190, 40, 1, &FreeSans9pt7b, msg[1],BLACK);// Team Name
    tft.drawFastHLine(0, 44, tft.width(), BLACK); // Line
    showmsgXY(10, 80, 1, &FreeSans9pt7b, msg[2],BLACK); // Ambient temp
    showmsgXY(10, 120, 1, &FreeSans9pt7b, msg[3],BLACK); // Target temp
    showmsgXY(10, 160, 1, &FreeSans9pt7b, msg[4],BLACK); // Result
    showmsgXY(10, 200, 1, &FreeSans9pt7b, "COUNT                    :",BLACK); //
Person Count
    left_btn.initButton(&tft,  80, 280,  60,40, BLACK, CYAN, BLACK, "<", 1);
    right_btn.initButton(&tft, 160, 280, 60,40, BLACK, CYAN, BLACK, ">", 1);
    exit_btn.initButton(&tft,  320, 280,  60,40, BLACK, CYAN, BLACK, "EXIT", 1);
    log_btn.initButton(&tft,   240, 280,  60,40, BLACK, CYAN, BLACK, "LOGS", 1);
    log_btn.drawButton(false);
    left_btn.drawButton(false);
```

```
    right_btn.drawButton(false);
    exit_btn.drawButton(false);
    Serial.println("Home Screen Displayed");

}

void clear_value()
{
   tft.drawRect(250, 60, 150, 120, WHITE);
   tft.fillRect(250, 60, 150, 120, WHITE);
   Serial.println("Past Values Cleared");

}
bool Touch_getXY(void)
{
    TSPoint p = ts.getPoint();
    pinMode(YP, OUTPUT);      //restore shared pins
    pinMode(XM, OUTPUT);
    digitalWrite(YP, HIGH);   //because TFT control pins
    digitalWrite(XM, HIGH);
    bool pressed = (p.z > MINPRESSURE && p.z < MAXPRESSURE);
    if (pressed) {
        pixel_x = map(p.y, TS_BOT, TS_TOP, 0, tft.width());
        pixel_y = map(p.x, TS_LEFT, TS_RT, 0, tft.height());
    }
```

```
    Serial.println("Pixel X Value");
    Serial.println(pixel_x);
    Serial.println("Button XY touch coordinates received");
    Serial.println("Pixel Y Value");
    Serial.println(pixel_y);
    return pressed;
}

void log_button_scan()
{
    bool down = Touch_getXY();
    log_btn.press(down && log_btn.contains(pixel_x, pixel_y));
    if (log_btn.justPressed()){
        Serial.println("Log Button Pressed");
        retrievelog();
        log_btn.drawButton(false);
        }
     exit_btn.press(down && exit_btn.contains(pixel_x, pixel_y));
      if (exit_btn.justPressed()){
        Serial.println("Exit Button Pressed");
        home_screen();
        log_stat = false;
        exit_btn.drawButton(false);
        }
}
```

```
void touch_display_setup()
{
//   Serial.begin(9600);
    tft.reset();
    uint16_t ID = tft.readID();
    ID = 0x9486; // write-only shield
    tft.begin(ID);
    tft.setRotation(aspect);
   Serial.println("Touch Display Init Done!");
    home_screen();
}


//************************************Speaker Section*********************************
void speaker_setup()
{
  tmrpcm.speakerPin=46;
  tmrpcm.setVolume(5);
  Serial.println("Speaker Init Done!");
}

//************************************Data Logging Section*****************************
void datalog(String dataString)
```

```
{
 Serial.println("Data Log Started");
 File dataFile = SD.open("datalog.txt", FILE_WRITE);
 if (dataFile) {
    Serial.println("Data Log Done!");
    dataFile.println(dataString);
    dataFile.close();
  }
  // if the file isn't open, pop up an error:
  else {
    Serial.println("Error opening datalog.txt");
  }
}
void retrievelog()
{
  Serial.println("Data Retrieve Started");
  int i = 0;
  tft.drawRect(10, 60, 400, 180, WHITE);
  tft.fillRect(10, 60, 400, 180, WHITE);
  File dataFile = SD.open("datalog.txt");
  if (dataFile) {
      while (dataFile.available()) { //execute while file is available
      String line = dataFile.readStringUntil('\n');
      i = i+20;
      showmsgXY(10, 80+i, 1, &FreeSans9pt7b, line, BLACK); //Project Name
```

```
      Serial.println(line);
      if(i == 140)
      {
        Serial.println("break");
        Serial.println("Data Retrieve Successful");
        log_stat = true;
        break;
      }
      }
//      String letter = dataFile.read(); //read next character from file
    dataFile.close(); //close file
  }
}

//************************************Temperature Sensor Section*****************************
void thermal_ir_sensor_setup()
{
 mlx.begin();//initialize temp sensor
 Serial.println("Thermal IR Init Successful");

}

void temp_scan()
{
  Serial.println("Temperature Scan Started");
```

```
  count = count + 1;

  tft.drawRect(250, 180, 40, 30, WHITE);

  tft.fillRect(250, 180, 40, 30, WHITE);

  showmsgXY(250, 200, 1, &FreeSans9pt7b, String(count),BLACK); // Pass

  String log_value;

  ambient = mlx.readAmbientTempC();

  target = mlx.readObjectTempC();

  dtostrf(ambient, 6, 2, amb); // Converting float to char

  dtostrf(target, 6, 2, tar); //  Converting float to char

  showmsgXY(250, 80, 1, &FreeSans9pt7b, amb,BLACK); // Ambient value

  showmsgXY(320, 80, 1, &FreeSans9pt7b, msg[7],BLACK); // Unit Celcius for ambient
value

  showmsgXY(250, 120, 1, &FreeSans9pt7b, tar,BLACK); // Target value

  showmsgXY(320, 120, 1, &FreeSans9pt7b, msg[7],BLACK);// Unit Celcius for target
value

  Serial.println("Ambient Temp in deg C!");

  Serial.println(ambient);

  Serial.println("Target Temp in deg C!");

  Serial.println(target);

  if (target > threshold){

    showmsgXY(250, 160, 1, &FreeSans9pt7b, msg[6],RED); // Fail

    tmrpcm.play("Audio_4.wav");

    Serial.println("Playing Announcement 4");

    Serial.println("Target temperature is higher than 38degC - Failed");

    String result = msg[6];

    delay(7000);
```

```
  }
  else{
  showmsgXY(250, 160, 1, &FreeSans9pt7b, msg[5],GREEN); // Pass
     Serial.println("Playing Announcement 3");

    tmrpcm.play("Audio_3.wav");
    Serial.println("Target temperature is less than 38degC - Passed");
    String result = msg[5];
    delay(7000);
  }
  log_value = String("Time in mS:") + String(millis()) + String(" Count=") + String(count) +
String(" Target_Temp=")+ String(target) + String("C ") + result;
  Serial.println("Data Logged");
  Serial.println(log_value);
  datalog(log_value);
  clear_value();
}

//************************************Motion Sensor
Section*****************************

void motion_sensor_setup()
{
  pinMode(pir_led, OUTPUT);   // initalize LED as an output
  pinMode(pir_input, INPUT);    // initialize sensor as an input
  Serial.println("Motion Sensor Init Done!");
```

```
}

void motion_scan()
{
    Serial.println("Motion Scan Started");
    if (digitalRead(pir_input) == HIGH) { // check if the sensor is HIGH
    digitalWrite(pir_led, HIGH);      // turn LED ON
    Serial.println("Motion detected!");
    Serial.println("Playing Announcement 1");
    tmrpcm.play("Audio_1.wav");
    delay(2000);
    Serial.println("Playing Announcement 2");
    tmrpcm.play("Audio_2.wav");
    delay(7000);
    temp_scan();
  }
  else {
    digitalWrite(pir_led, LOW);       // turn LED OFF
    Serial.println("No Motion Detected");
//    delay(1000);                  // delay 100 milliseconds
  }
}
```

```
//************************************Main Setup
Section*****************************


void setup()

{

  Serial.begin(9600);     // initialize serial

  Serial.println("Started All Modules Init");

  sdcard_setup();

  speaker_setup();

  motion_sensor_setup(); // initialize PIR motion sensor

  thermal_ir_sensor_setup();// initialize thermal IR motion sensor

  touch_display_setup();// initialize touch display

  home_screen(); //Shows the home screen with history button to view past records (person
count, time stamp, temperature)

}


//************************************Main Polling Loop
Section*****************************


void loop()

{

  Serial.println("Motion Scan In Progress");

  if (log_stat==false)

  {

     motion_scan(); // Check PIR sensor and takes decision based on its output

   }

  Serial.println("Button Scan In Progress");
```

```
 log_button_scan();

 //delay(15000);

}
```

Header File: Display_init.h

```
#include <Adafruit_GFX.h> // Hardware-specific library
#include <MCUFRIEND_kbv.h>
#include <Fonts/FreeSans9pt7b.h>
#include <Fonts/FreeSans12pt7b.h>
#include <Fonts/FreeSerif12pt7b.h>
#include <FreeDefaultFonts.h>
#define BLACK   0x0000
#define BLUE    0x001F
#define RED     0xF800
#define GREEN   0x07E0
#define CYAN    0x07FF
#define MAGENTA 0xF81F
#define YELLOW  0xFFE0
#define WHITE   0xFFFF
MCUFRIEND_kbv tft;
char *msg[] = {"NON-CONTACT TEMPERATURE SCANNER", "TEAM 6",
"AMBIENT TEMPERATURE :", "TARGET TEMPERATURE  :", "RESULT
:", "PASS", "FAIL", "C"};
uint8_t aspect = 3;
#include <TouchScreen.h>
#define MINPRESSURE 200
#define MAXPRESSURE 1000
char *name = "Please Calibrate.";  //edit name of shield
const int XP=6,XM=56,YP=55,YM=7; //ID=0x9341
const int TS_LEFT=907,TS_RT=136,TS_TOP=942,TS_BOT=139;
Adafruit_GFX_Button log_btn, left_btn, right_btn, exit_btn;
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);
TSPoint tp;
int pixel_x, pixel_y;     //Touch_getXY() updates global vars
```

Header File: PIR_Sensor_Init.h

```
int pir_led = 13;            // the pin that the LED is atteched for visual indicator
int pir_input = 22;              // the pin that the PIR sensor is atteched to
```

Header File: Speaker_Init.h

```
#include "SD.h"
#define SD_ChipSelectPin 53
#include "TMRpcm.h"
```

```
#include "SPI.h"
TMRpcm tmrpcm;
```

Header File: Temp_IR_Init.h

```
#include <Adafruit_MLX90614.h>
//#define I2C_ADDR 0x27 //I2C adress, you should use the code to scan the temp sensor address first (0x27) here
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
float ambient;
float target;
float threshold = 38;
char amb[8]; // Buffer big enough for 7-character float
char tar[8]; // Buffer big enough for 7-character float
char result[8]; // Buffer big enough for 7-character float
```

Product Testing

Black box testing and system verification were performed as part of the product testing.

Black Box Testing

Home Screen Verification

The home screen shows the banner, ambient, target, count, and result. This also includes 4 buttons as follows:

“<” Scroll Left During Logs Screen.

“>” Scroll Right During Logs Screen.

“LOGS” Display the past logs.

“EXIT” Takes back to home screen always.

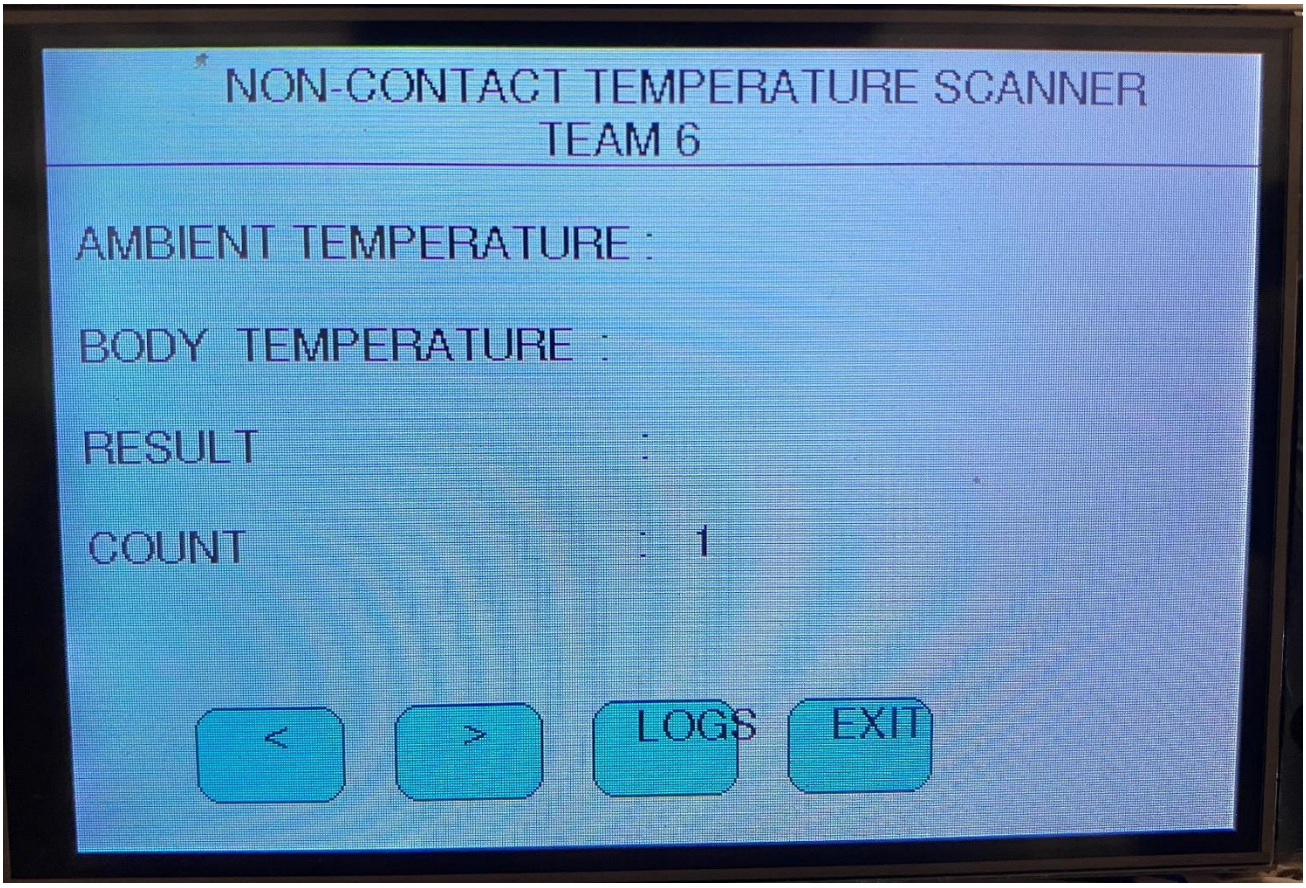


Figure 25 Home screen for the GUI based Touch Display

Pass Screen Verification

The pass screen shows the banner, ambient, target, count, and result as “Pass” in green color if target temperature is <100.4 F.

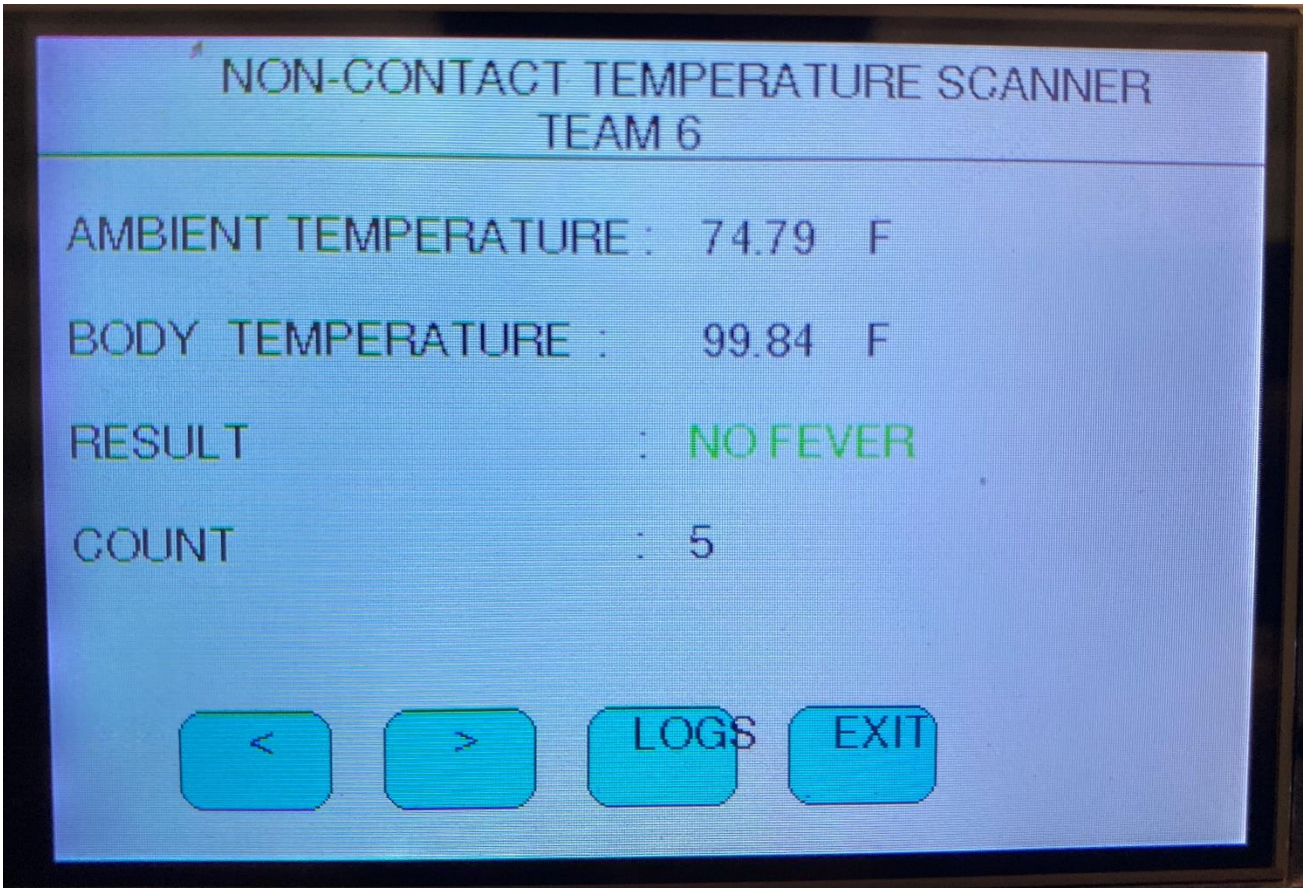


Figure 26 Pass Screen displayed to user

Fail Screen Verification

The failure screen shows the banner, ambient, target, count, and result as “Fail” in red color if target temperature is >100.4 F.

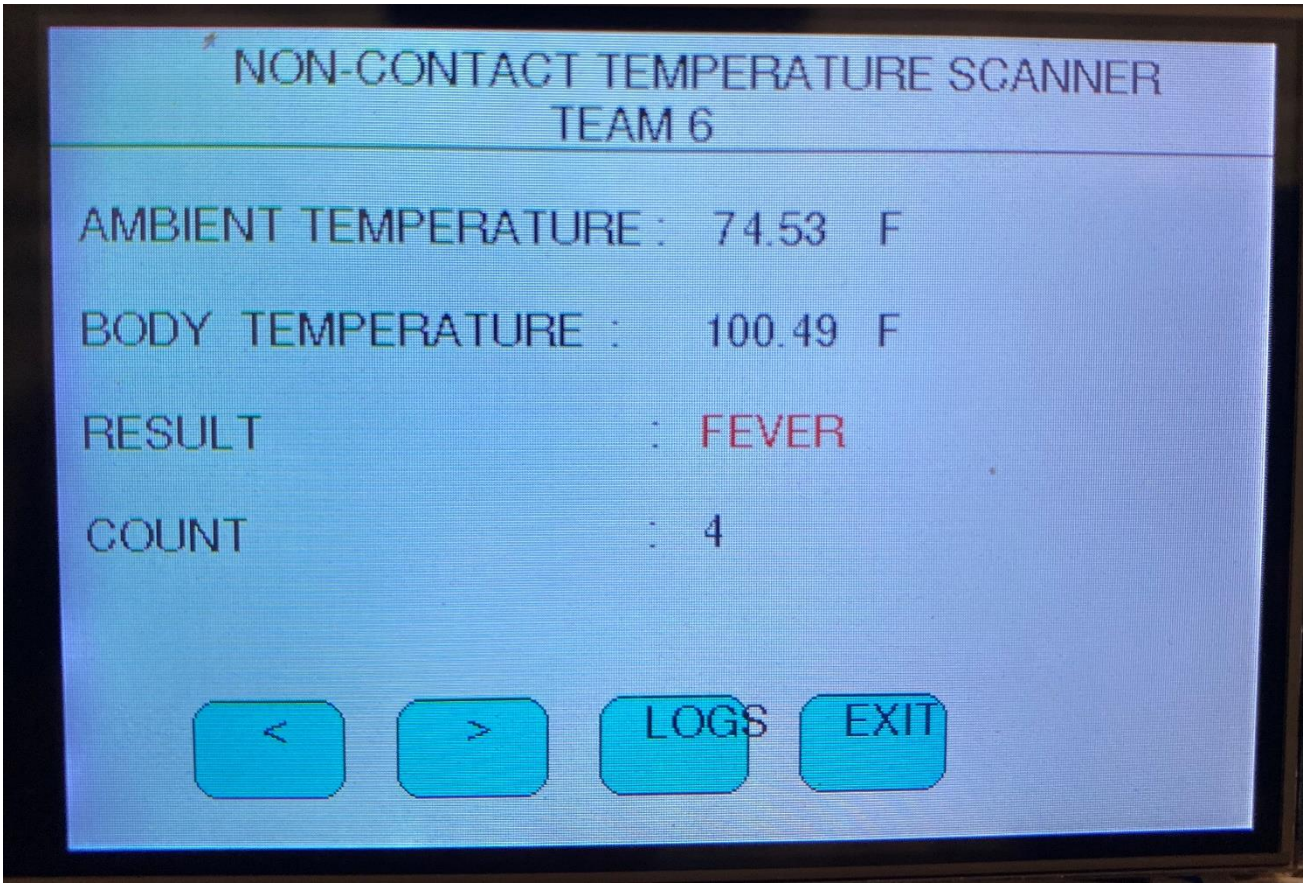


Figure 27 Fail screen displayed to user

Log History Retrieval Screen Verification

- When “LOGS” button is pressed, it shows the banner and past logs in the format as:
  Time Stamp in mS: XXXXX Count=XX Target Temp: XX C
- If user did not exit, it automatically goes back to home screen. During this screen, motion sensing loop will not work.

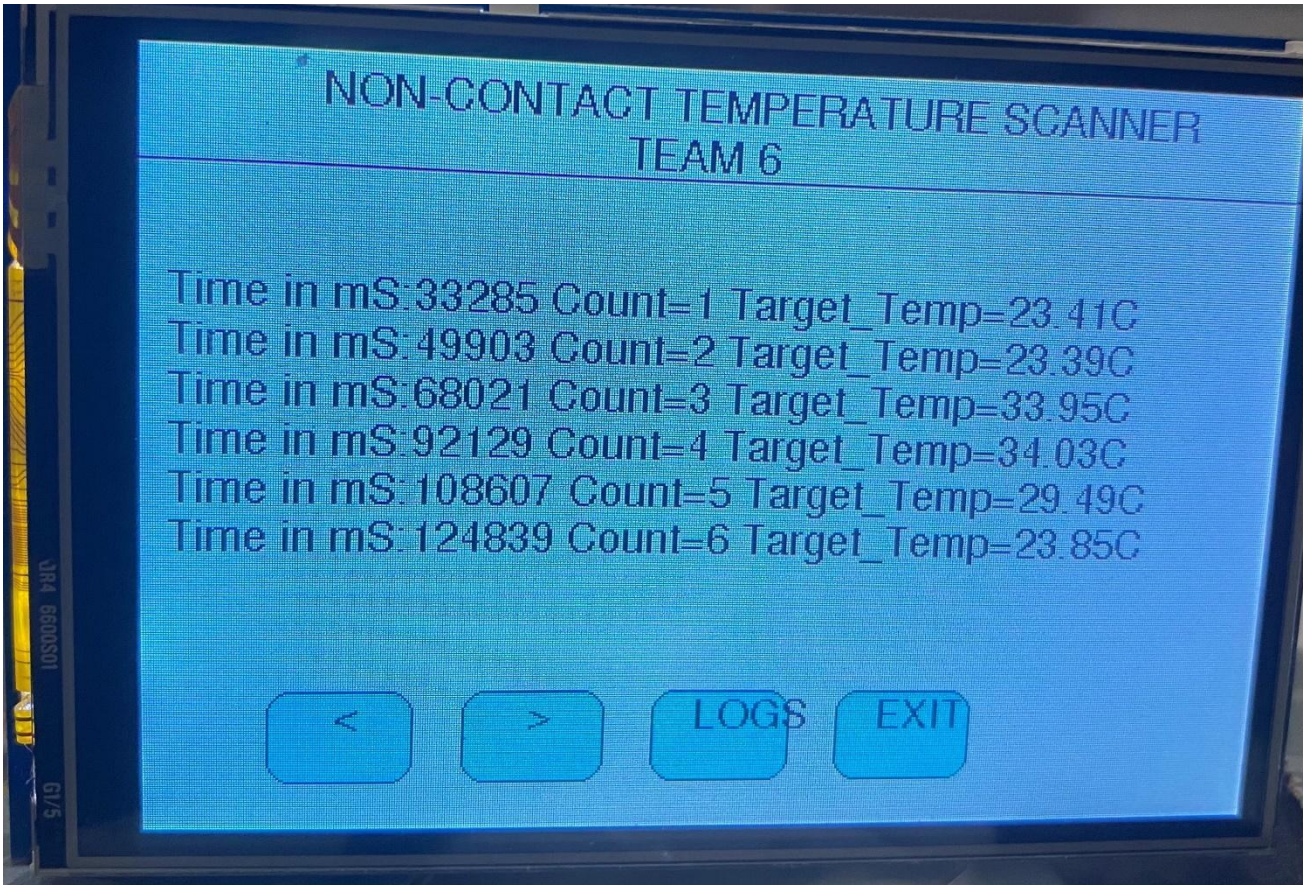


Figure 28 User logs with time, count, and temperature.

Log History SD Card storage verification.

Verified all the data logs stored in a csv format in the SD card. Attached the sample data logs collected in the SD storage.

logs.csv

Audio Announcement Verification:

Following voice phrases recorded in wav format and stored in SD card. These announcements are retrieved at the specified conditions and verified against the test cases.

Speaker Announcement 1: “Person detected.”

Speaker Announcement 2: “Please stand at the target location for body temperature scan.”

Speaker Announcement 3: “Scan complete. You have no fever, access granted.”

Speaker Announcement 4: “Scan complete. You have a fever, access denied.”

System Verification

The system was verified by analyzing the logs printed on the serial terminal.

Test Debug Logs Printed on Serial

Started All Modules Init

SD Card Init Done.

Log File Detected!

Speaker Init Done!

Motion Sensor Init Done!

Thermal IR Init Successful

Touch Display Init Done!

NON-CONTACT TEMPERATURE SCANNER

TEAM 6

AMBIENT TEMPERATURE :

TARGET TEMPERATURE :

RESULT :

COUNT :

Home Screen Displayed

Motion Scan In Progress

Motion Scan Started

No Motion Detected

Button Scan In Progress

Pixel X Value

0

Button XY touch coordinates received

Pixel Y Value

0

Motion Scan In Progress

Motion Scan Started

No Motion Detected

Button Scan In Progress

Pixel X Value

0

Button XY touch coordinates received

Pixel Y Value

0

Motion Scan In Progress

Motion Scan Started

No Motion Detected

Button Scan In Progress

Pixel X Value

0

Button XY touch coordinates received

Pixel Y Value

0

Motion Scan In Progress

Motion Scan Started

Motion detected!

Playing Announcement 1

Playing Announcement 2

Temperature Scan Started!

24.91

C

25.29

C

Ambient Temp in deg C!

24.91

Target Temp in deg C!

25.29

PASS

Playing Announcement 3

Target temperature is less than 38degC - Passed

Data Logged

Time in mS:37453 Count=1 Target_Temp=25.29C

Data Log Started

Data Log Done!

Past Values Cleared

Button Scan In Progress

Pixel X Value

0

Button XY touch coordinates received

Pixel Y Value

0

Motion Scan In Progress

Motion Scan Started

No Motion Detected

Button Scan In Progress

Pixel X Value

0

Button XY touch coordinates received

Pixel Y Value

0

Motion Scan In Progress

Motion Scan Started

No Motion Detected

Motion Scan In Progress

Motion Scan Started

Motion detected!

Playing Announcement 1

Playing Announcement 2

Temperature Scan Started

2

24.91

C

24.41

C

Ambient Temp in deg C!

24.91

Target Temp in deg C!

24.41

PASS

Playing Announcement 3

Target temperature is less than 38degC - Passed

Data Logged

Time in mS:69695 Count=2 Target_Temp=24.41C

Data Log Started

Data Log Done!

Past Values Cleared

Button Scan In Progress

Pixel X Value

0

Button XY touch coordinates received

Pixel Y Value

0

Motion Scan In Progress

Motion Scan Started

No Motion Detected

Button Scan In Progress

Pixel X Value

245

Button XY touch coordinates received

Pixel Y Value

276

Log Button Pressed

Data Retrieve Started

Time in mS:33285 Count=1 Target_Temp=23.41C

Time in mS:33285 Count=1 Target_Temp=23.41C

Time in mS:49903 Count=2 Target_Temp=23.39C

Time in mS:49903 Count=2 Target_Temp=23.39C

Time in mS:68021 Count=3 Target_Temp=33.95C

Time in mS:68021 Count=3 Target_Temp=33.95C

Time in mS:92129 Count=4 Target_Temp=34.03C

Time in mS:92129 Count=4 Target_Temp=34.03C

Time in mS:108607 Count=5 Target_Temp=29.49C

Time in mS:108607 Count=5 Target_Temp=29.49C

Time in mS:124839 Count=6 Target_Temp=23.85C

Time in mS:124839 Count=6 Target_Temp=23.85C

Time in mS:87549 Count=1 Target_Temp=23.65C

Time in mS:87549 Count=1 Target_Temp=23.65C

break

Data Retrieve Successful

Motion Scan In Progress

Button Scan In Progress

Pixel X Value

245

Button XY touch coordinates received.

Pixel Y Value

276

Motion Scan In Progress

Button Scan In Progress

Pixel X Value

323

Button XY touch coordinates received.

Pixel Y Value

273

Exit Button Pressed

NON-CONTACT TEMPERATURE SCANNER

TEAM 6

AMBIENT TEMPERATURE :

TARGET TEMPERATURE :

RESULT :

COUNT :

Home Screen Displayed

Motion Scan In Progress

Motion Scan Started

No Motion Detected

Appendix

Appendix 1: Atmega2560 Datasheet

Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V

8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash

DATASHEET

Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 × 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16MHz
  - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 64K/128K/256KBytes of In-System Self-Programmable Flash
  - 4Kbytes EEPROM
  - 8Kbytes Internal SRAM
  - Write/Erase Cycles:10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/ 100 years at 25°C
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
    - Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 64 sense channels
- JTAG (IEEE® std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
  - Output Compare Modulator
  - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
  - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
  - Master/Slave SPI Serial Interface
  - Byte Oriented 2-wire Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
  - 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)
  - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
  - RoHS/Fully Green
- Temperature Range:
  - -40°C to 85°C Industrial

Figure 29 Datasheet for the ATmega microcontroller

Appendix 2: Touch display and SD Adapter, Pin Outs, and Product Specs.

Product Parameters:

Name	Parameter
Display Color	RGB 65K color
SKU	MAR3501（have touch screen）/MAR3502(have no touch screen)
Screen Size	3.5(inch)
Type	TFT
Driver IC	ILI9486
Resolution	480*320 (Pixel)
Module Interface	8-bit parallel interface
Active Area	73.44*48.96(mm)
Module PCB Size	85.49*55.63(mm)
Operating Temperature	-20℃~70℃
Storage Temperature	-40℃~70℃
Operating Voltage	5V/3.3V
Power Consumption	TBD
Product Weight(Package containing)	44g(have touch screen)，55g(have no touch screen)

Figure 30 Table for the product Specs

Display and SD Card Pin Details:

Number	Pin Label	Pin Description
1	LCD_RST	LCD bus reset signal, low level reset
2	LCD_CS	LCD bus chip select signal, low level enables
3	LCD_RS	LCD bus command / data selection signal, low level: command, high level: data
4	LCD_WR	LCD bus write signal
5	LCD_RD	LCD bus read signal
6	GND	Power ground
7	5V	5V power input
8	3V3	3.3V power input, this pin can be disconnected
9	LCD_D0	LCD 8-bit data Bit0
10	LCD_D1	LCD 8-bit data Bit1
11	LCD_D2	LCD 8-bit data Bit2
12	LCD_D3	LCD 8-bit data Bit3
13	LCD_D4	LCD 8-bit data Bit4
14	LCD_D5	LCD 8-bit data Bit5
15	LCD_D6	LCD 8-bit data Bit6
16	LCD_D7	LCD 8-bit data Bit7

17	SD_SS	SD card SPI bus chip select signal, low level enable
18	SD_DI	SD card SPI bus MOSI signal
19	SD_DO	SD card SPI bus MISO signal
20	SD_SCK	SD card SPI bus clock signal

Figure 31 Table for the pin connections taken from data sheets

Appendix 3: Speaker and TIP29A Datasheet

CSS-66668N Speaker: https://www.cuidevices.com/product/resource/digikeypdf/css-66668n.pdf

MODEL: CSS-66668N | **DESCRIPTION:** SPEAKER

FEATURES
- 90 dB SPL
- 200 Hz resonant frequency
- Nd-Fe-B magnet

RoHS

SPECIFICATIONS

parameter	conditions/description	min	typ	max	units
input power	maximum power: IEC-60268-5, filter 60s on/120s off, 10 cycles at room temp		3.0	4.0	W
impedance	at 0.8 kHz	6.8	8	9.2	Ω
resonant frequency (Fo)	at 1.0 V	160	200	240	Hz
frequency response	output SPL -10 dB	Fo		15,000	Hz
sound pressure level	at 1.0 W, 0.1 m ave, at 0.6, 0.8, 1.0, 1.2 kHz at 3.0 W, 0.5 m ave, at 0.6, 0.8, 1.0, 1.2 kHz	98 88	101 91	104 94	dB dB
distortion	at 1.0 kHz, 1.0 W			5	%
buzz, rattle, etc.	must be normal at sine wave between Fo ~ 20,000 Hz		4.9		V
dimensions	66 x 66 x 29				mm
magnet	Nd-Fe-B				
material	metal				
cone material	paper & foam				
terminal	solder terminals				
weight			101		g
operating temperature		-20		55	°C
storage temperature		-30		70	°C
RoHS	yes				

Notes: 1. All specifications measured at 15~35°C, humidity at 45~85%, under 86~106 kPa pressure, unless otherwise noted.

Figure 32 Data sheet of speaker module

TIP29C: https://www.st.com/resource/en/datasheet/tip29a.pdf

TIP29, A, B, C (NPN), TIP30, A, B, C (PNP)

Complementary Silicon Plastic Power Transistors

Designed for use in general purpose amplifier and switching applications. Compact TO−220 package.

Features

- These Devices are Pb−Free and are RoHS Compliant*

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Collector − Emitter Voltage TIP29G, TIP30G TIP29AG, TIP30AG TIP29BG, TIP30BG TIP29CG, TIP30CG	V_{CEO}	 40 60 80 100	Vdc
Collector − Base Voltage TIP29G, TIP30G TIP29AG, TIP30AG TIP29BG, TIP30BG TIP29CG, TIP30CG	V_{CB}	 40 60 80 100	Vdc
Emitter − Base Voltage	V_{EB}	5.0	Vdc
Collector Current − Continuous	I_C	1.0	Adc
Collector Current − Peak	I_{CM}	3.0	Adc
Base Current	I_B	0.4	Adc
Total Power Dissipation @ T_C = 25°C Derate above 25°C	P_D	 30 0.24	W W/°C
Total Power Dissipation @ T_A = 25°C Derate above 25°C	P_D	 2.0 0.016	W W/°C
Unclamped Inductive Load Energy (Note 1)	E	32	mJ
Operating and Storage Junction Temperature Range	T_J, T_{stg}	−65 to +150	°C

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

1. This rating based on testing with L_C = 20 mH, R_{BE} = 100 Ω, V_{CC} = 10 V, I_C = 1.8 A, P.R.F = 10 Hz

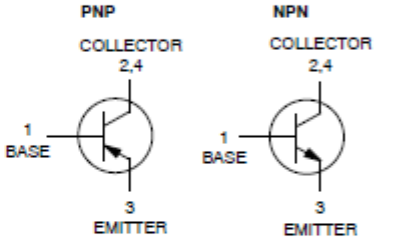
THERMAL CHARACTERISTICS

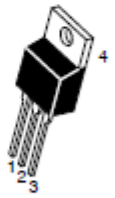
Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction−to−Ambient	$R_{\theta JA}$	62.5	°C/W
Thermal Resistance, Junction−to−Case	$R_{\theta JC}$	4.167	°C/W

ON Semiconductor®

www.onsemi.com

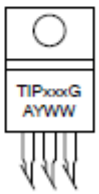
1 AMPERE POWER TRANSISTORS COMPLEMENTARY SILICON 40, 60, 80, 100 VOLTS, 80 WATTS





TO−220
CASE 221A
STYLE 1

MARKING DIAGRAM



TIPxxx = Device Code:
29, 29A, 29B, 29C
30, 30A, 30B, 30C
A = Assembly Location
Y = Year
WW = Work Week
G = Pb−Free Package

Figure 33 Datasheet for BJT transistor