ECE 342 Developer Guide

Faaiq Waqar[1], **Paige Barylsky**[1], **Jeffrey Unrein**[1]

[1]School of Electrical Engineering and Computer Science
Oregon State University

waqarf@oregonstate.edu, barylskp@oregonstate.edu, unreinj@oregonstate.edu

1. System Overview

The goal of the project is to design and implement a contact-less temperature sensor, made to be easy to operate for the end user. The resulting project is a smart-temperature sensor capable of accurate measurement from a range of up to 30 cm without any contact to operate. SmartTemp uses a set of intuitive interfaces, including visual notifications on the front mounted LCD panel, indicative tones from the piezo speaker for different reading types, an IR remote with instructional button labels, and wireless communication to the mobile GUI, capable of exporting data in CSV format. The system makes use of a pair of an ultrasonic temperature and IR temperature sensor to understand the surface's distance from the sensor to compensate for a drop off in IR radiation captured by the sensor, giving accuracy from greater ranges. The device stores user and temperature information for a full 24 hours at a time, and this history can be visited directly on the device. At the heart of this project is an Arduino Nano 33 IoT, which drives the accuracy and speed hungry elements of this project, while also making the device IoT capable with its surface mount NINA-W102 radio. This document will outline the electrical specifications of the device, give detailed system operation instructions, and develop the wiring, programming and mechanical development of the device. Additionally, information on the student designed PCB used in this project is described and shown. The end of this document details information on the parts necessary complete this project.

2. Electrical Specifications

Below, the electrical specifications for all electrically active system components can be seen in table 1 and 2 below. Table 1 discusses the minimum and maximum voltage specifications of each component. Table 2 discusses the nominal and maximum currents of the components, as well as the operating temperature ranges.

3. User Guide

The use of the device is broken into 2 primary groups, users and owners. For the user, device operation is focused on the local device features, including measurement, user ID entry and checking logs. For the owner, the focus is using the device to check on who is operating the device, and the temperatures recorded remotely using the GUI on a mobile device. Both are discussed in the following subsections.

3.1. For the User

1. To power on the device, plug in a USB type C cable to the connector on the right side of the device. This USB source should not exceed 21 volts.

Table 1. Electrical Specification Table [Voltages]

Component	Component Name	Min Voltage	Max Voltage
1	Arduino Nano 33 IoT	4.5 V	21 V
2	TSOP38238 IR Reciever Diode	2.5 V	5.5 V
3	MLX90614 IR Temperature Sensor	3.6 V	5 V
4	LCD1602 Module	3.0 V	7.0 V
5	HC - SR04 Ultrasonic Sensor	4.5 V	5.5 V
6	BKM06 Speaker	- 15 V	15 V

Table 2. Electrical Specification Table [Currents and Operating Temps]

Component	Nominal Current	Max Current	Operating Temp
1	7 mA	196 mA	-40°C - 85°C
2	0.35 mA	0.45 mA	-25°C - 85°C
3	1 mA	1.5 mA	-40°C - 125°C
4	0.2 mA	0.4 mA	0°C - 60°C
5	15 mA	20 mA	-15°C - 70°C
6	5 mA	21 mA	-20°C - 70°C

2. Wait for the device to power on. This should take anywhere from 10-20 seconds, while the device attempts to connect to the local network
3. Once the initial prompt on the LCD shows up, it should read ”Enter User ID:”. Use the number pad [1-9] to enter a user ID. This ID will be tied to the operations conducted on the device.
4. To record a temperature, hit the ”°F?” button on the controller. doing so will begin the temperature recording process. The sensor will prompt the user to approach closer or move further on the LCD, until the user is in range. When this happens, a recorded temperature will appear on screen. If the recorded temperature is above 100.4°F, the screen will briefly display ”Fever!” and play a high C note on the side mounted speaker.
5. To view previous records, using the ”Enter Records” button on the remote. This will bring up a record screen with the record number being viewed out of the total records, the ID for the record, and the temperature associated.
6. To navigate other records, using the up and down arrows will take the user to different entries.
7. To exit the previous records, using the ”Enter Records” button once again will take the user out of the user history.
8. To change user ID/ switch off to another user, use the ”New User” button, and the original ”Enter User ID:” prompt will display again.

3.2. For the Owner

1. To download the application, use the QR code seen in Figure 1. This application works for Android and Apple devices alike.
2. Once in the application, the owner should hit the play button at the top right

Figure 1. QR Code to Download Mobile GUI

of the GUI. This will connect the GUI to the Arduino, and begin communication.

3. Information will be communicated wirelessly and show up on the current user, current temperature, and temperature graph modules. This requires no additional steps from the owner.
4. Using the tabs at the bottom of the temperature graph module will allow the owner to view a perspective of the temperatures recorded Live, and up to the range of 24 hours.
5. To export the recorded data, using the "..." button at the bottom of the temperature graph module will take them to a new screen, This screen has 2 options, "Export to CSV" will send the data in CSV to the email of the owner registered with the application.
6. To clear the recorded data, using the "..." button at the bottom of the temperature graph module will take them to a new screen, This screen has 2 options, "Clear Data" will clear the temperatures recorded to the GUI.

4. Design Artifacts

The following section contains the collection of artifacts used to properly connect, construct and operate the device. It is also intended to provide a high level understanding of the design philosophy of the device.

4.1. Block Diagram

Figure 2 shows a high level block diagram of the temperature sensor project. The block diagram makes use of high level descriptors to define collections of devices for each of the blocks. Sensing blocks, such as the temperature sensing and distance sensing blocks, are the amalgamation of the components used to operate the sensors, including passive elements such as resistors and capacitors. The power supply at a high level is the power supplied to the system via an external USB connection that drives voltage to the processor unit, that is then distributed via the system along PCB bus lines to the different system components.

Arrows in the diagram are used to notate the connection of different interconnects of devices. These are defined more explicitly in the next subsection, but it is worth mentioning what they mean. The format of the interface definitions goes *Source_Destination_Type*. In the block diagram, there are 7 types of interfaces visible. DCPwr interfaces are sources of DC power used to operate different components. Dsig notates digital signals, which are used for communication interfaces using either the digital processor lines, or I2C connectivity for communication. Asig are analog signals that are either transmitted as infrared light, or via the processor analog lines. The Env type describes any environmental variables that are measured by sensors. Mech refers to anything that is triggered by mechanical stress, such as the push of a button. UsrOut are types of interfaces used with the intention of communicating with the end user. Finally, Data types are packets of information communicated across network mediums, such as WiFi or an IoT network.

Each block and interface shown in Figure 2 has properties that specify how its inputs and outputs should operate. These properties specified for all system blocks and interfaces are defined in table format sequentially in Figures 4,5,6,7 and 8.

4.2. Schematic

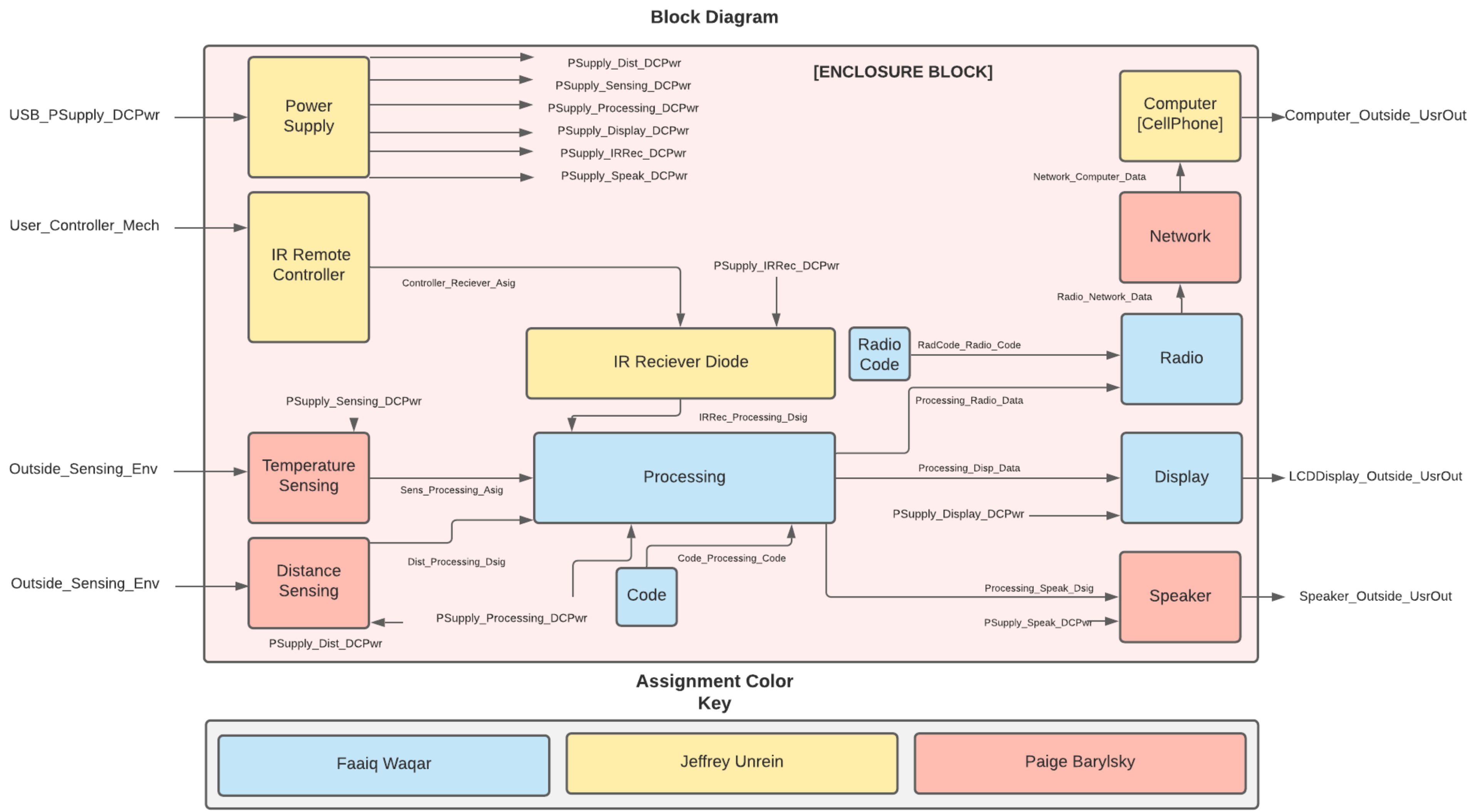
The interconnect of the components used in this system can be seen in Figure 8. The components shown in this schematic are the ultrasonic sensor, the ir temperature sensor, the piezo speaker the LCD, and of course the Arduino. As seen in this interconnect, the majority of components are used on the digital set of pins from the Arduino, with the exception of the temperature sensor, which rely on I2C communication driven by SCL and SDA on the analog A4 and A5 pins. The interconnect seen in this schematic contains 2 surface mount resistors, R2 and R3, and a single current limiting resistor R1.

4.3. Mechanical Drawings

Here, Figures 9, 10, 11, 12, 13, 14, and 15 show mechanical drawings with dimensions for all primary system components. These have been generated to contain measurements of different radii, widths, heights, and lengths. Figure 9 shows the footprint of the Arduino Nano 33 IoT, with dimensions labeled in inches. Mounted directly to this, seen in Figure 10, is the NINA-W102 radio module, with measurements shown in mm. Its worth noting that the dimensions of this part dictate the height dimension of the Arduino.

Figure 11 shows the drawing of the HC-SR04 ultrasonic sensor used to measure distance of the user. Figure 12 is the mechanical drawing for the Melexis IR temperature. Figure 13 is the mechanical drawing of the piezo speaker. Figure 14 is a mechanical drawing of the LCD1602 module. The drawings for these connected components can be seen measured in mm.

Figure 15 is the built enclosure for the device. This construction of this was made of plywood planks, and held together using wood glue. The mechanical


Block Diagram
[ENCLOSURE BLOCK]
USB_PSupply_DCPwr
Power Supply
PSupply_Dist_DCPwr
PSupply_Sensing_DCPwr
PSupply_Processing_DCPwr
PSupply_Display_DCPwr
PSupply_IRRec_DCPwr
PSupply_Speak_DCPwr
Computer [CellPhone]
Computer_Outside_UsrOut
Network_Computer_Data
User_Controller_Mech
IR Remote Controller
Controller_Reciever_Asig
PSupply_IRRec_DCPwr
Network
Radio_Network_Data
IR Reciever Diode
Radio Code
RadCode_Radio_Code
Radio
PSupply_Sensing_DCPwr
IRRec_Processing_Dsig
Processing_Radio_Data
Outside_Sensing_Env
Temperature Sensing
Sens_Processing_Asig
Processing
Processing_Disp_Data
Display
LCDDisplay_Outside_UsrOut
PSupply_Display_DCPwr
Dist_Processing_Dsig
Code_Processing_Code
Outside_Sensing_Env
Distance Sensing
Code
Processing_Speak_Dsig
Speaker
Speaker_Outside_UsrOut
PSupply_Processing_DCPwr
PSupply_Speak_DCPwr
PSupply_Dist_DCPwr
Assignment Color Key
Faaiq Waqar
Jeffrey Unrein
Paige Barylsky

Block	Inputs	Input Specifications	Outputs	Output Specifications
Distance Sensor	Outside_Sensing_Env PSupply_Dist_DC Pwr	Sends DC signal to trigger, input is the reflected echo read out on the echo pin [Digital] 5 Volts Power Stage Supply	Dist_Processing_Dsig	Digital output contains information about time for reflection Operates on 2 digital pins for echo and trig on D2 & D3 [Processing]
Enclosure	N/A	Holes Created appropriately to prepare sensors for input [IR diode, temp sensor, distance sensor] 2 Mounting holes for inputting hooks and/or nails for wall mounting	N/A	Holes Created appropriately to prepare devices for output [Display]
Temp. Sensing	Outside_Sensing_Env PSupply_Sensing_Pwr	3V VDD/3.6V Max operating voltage 7.2mW Maximum Power Works in tandem with clock and data lines from the processing unit [I2C] , a microcontroller (Arduino Nano)	Sens_Processing_Asig	∓0.5°C Reading of temperature at room temp Capable of reading accurately at 0°C - 85°C
Speaker	Processing_Speak_DSig	Specifications: 80mV, 34 Ohms	Speaker_Outside_UsrOut	Frequency Range: 31Hz-4k Hz
Network	Radio_Network_Data	10-30 foot range during use of temperature device (Radio to	Network_Computer_Data	Signal computer GUI containing information about user and temp

Figure 3. Interface Definition Table [Part 1]

		Computer distance). Capable of using local wifi connection to wirelessly transmit data. Communicates user and temperature data (digital)		recording
Power Supply	USB_PSupply_DCPwr	Using USB connection driving the Processor USB-C to Micro-USB conversion line	PSupply_Dist_DCPwr PSupply_Sensing_DCPwr PSupply_Processing_DCPwr PSupply_Display_DCPwr PSupply_IRRec_DCPwr PSupply_Speak_DCPwr	Maximum output voltage of 5 Volts Direct link to the processing unit (Microcontroller, Arduino Nano) Indirect link to distance sensor, temp sensor, speaker, display, ir receiver diode.
IR Remote Controller	User_Controller_Mech	250 ∓ 100 gf (gram force) response onto plastic button pad	Controller_Reciever_Asig	Outputs analog infrared signal with a specific code to IR diode.

Figure 4. Interface Definition Table [Part 2]

IR Receiver Diode	Controller_Reciever_Asig	Analog signal in form of IR light from the IR Remote Controller 5V from the power supply, connection to common ground	IRRec_Processing_Dsig	Digital signal containing information from the remote controller to the processing unit
Computer [CellPhone]	Network_Computer_Data	Signal containing information about user and temp recording	Computer_Outside_UsrOut	Display of specific user and temp recording for that user on GUI Ability to export the data of temperature information collected over time in format of CSV via. email.
Processing	Sens_Processing_Asig IRRec_Processing_Dsig Dist_Processing_Dsig PSupply_Processing_DCPwr Code_Processing_Code	5 Volt operating voltage , 32 KB of flash memory, 30Kb of which are usable (2KB are used by OS bootloader) Receives signals from sensors connected to digital pins Receives I2C communication from analog pins	Processing_Speak_Dsig Processing_Disp_Data Processing_Radio_Data	Analog high C note output (~1kHz) to the speaker during interrupt operation for temperature threshold Outputs data signals to be displayed and outputted via radio.
Code	N/A	N/A	Code_Processing_Code	Code completed in C language for Arduino Devices Uses polling methods to do recording and measuring

Figure 5. Interface Definition Table [Part 3]

				sequence for temp reading Uses polling methods to switch between users on different buttons Controls storage of user data for elongated period of time, with the ability to view previous entries Controls speaker output when the reading is compared to temp and measures above 100.4 F, part of interrupt method for measurement Drives code for LCD Display of data collected
Radio Code	N/A	N/A	RadCode_Radio_Code	Controls radio signal for connection with network Allows connection to gateway and to pass information collected through network to the computer Writes specific information to digital pins to be read by GUI.
Radio	RadCode_Radio_Code Processing_Radio	Surface Mounted to the Processing Unit. (Included in Package)	Radio_Network_Data	10 foot range during use of temperature device (Radio to

Figure 6. Interface Definition Table [Part 4]

	_Data	Can send and receive data.		Computer distance). Capable of using local wifi connection to wirelessly transmit data. Communicates user and temperature data (digital)
Display	PSupply_Display_DCPwr Processing_Amp_Asig	Supply voltage: 5V(via Pin) (4.7V-5.3V operating voltage from the power supply at DC Requires 6 Digital Pins, 2 Power Pins from Processor block (Arduino Nano)	LCDDisplay_Outside_UsrOut	Outputs max of 2 lines,16 characters each Blue LED backlight with white character color, adjustable contrast Display of both user and temperature information refreshed on button commands

Figure 7. Interface Definition Table [Part 5]

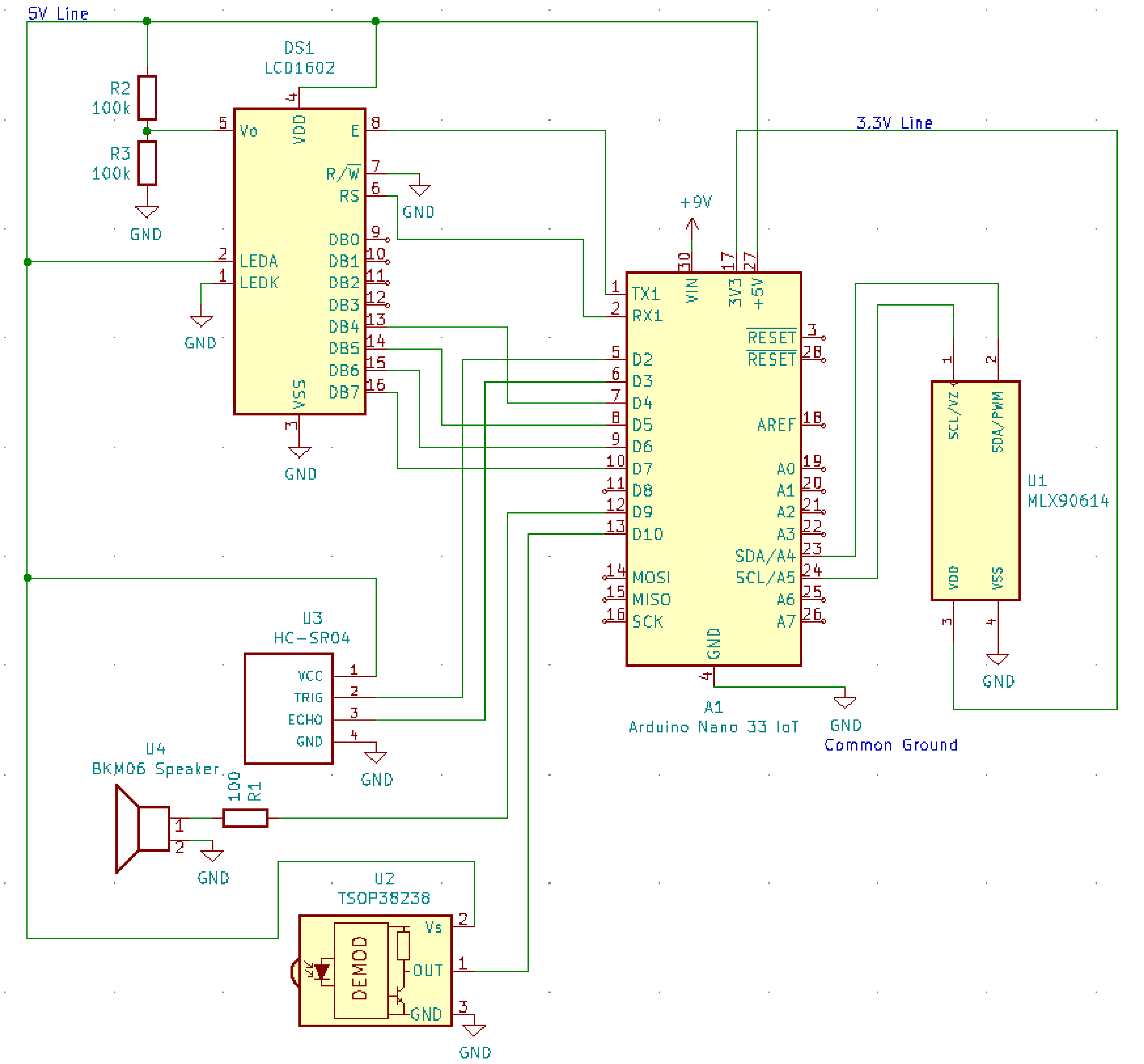


Figure 8. Final Wiring Schematic of System

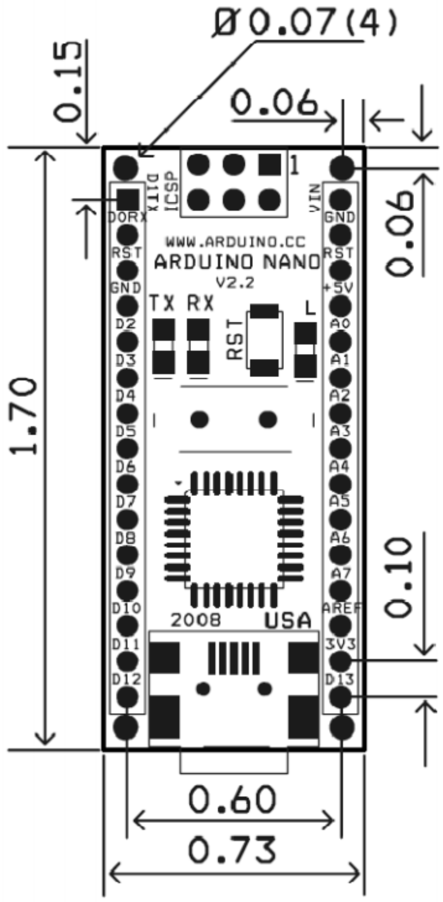


Figure 9. Mechanical Drawing of Arduino Nano

drawing is the resulting enclosure after construction. The measurements are all in mm.

4.4. Programming Discussion

5. PCB Information

A significant portion of the interconnect was developed for use on the main PCB. The primary idea behind the PCB used was to effectively connect the supply of power to the system, and easily develop the interconnect for processing connections. Because of this, jumpers are used to easily organize the system connections on the perimeter of the board. This organization can be seen in the PCB layout schematic in Figure 16. This organization was then translated to PCB traces, the layers of this design which can be seen in Figure 17. The physical interpretation of this model is seen in Figures 18 and 19.

Its worth noting the space for the PowerBoost on this. In the initial design of the system, a battery was planned to be used for the system, with the PowerBoost used to step up the voltage to be used for the Arduino. Due to some changes in design in the end, this is no longer used.

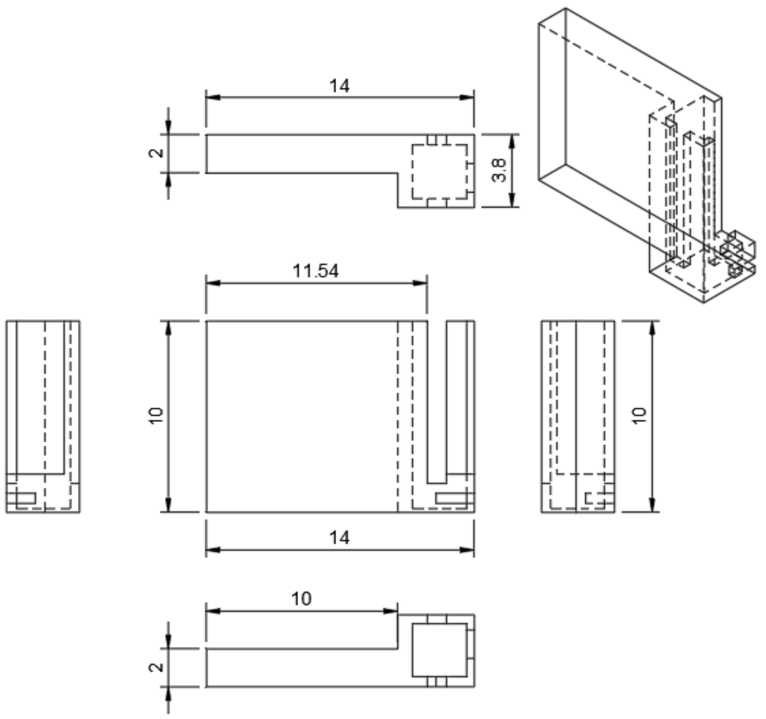


Figure 10. Mechanical Drawing of NINA Radio

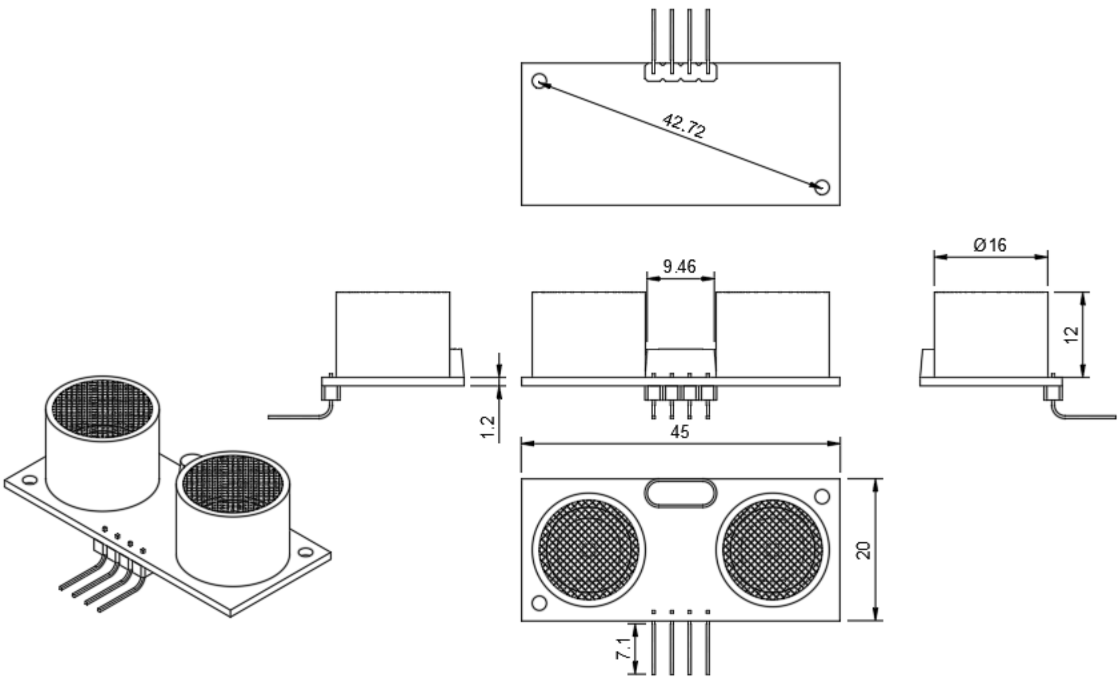


Figure 11. Mechanical Drawing of Ultrasonic Sensor

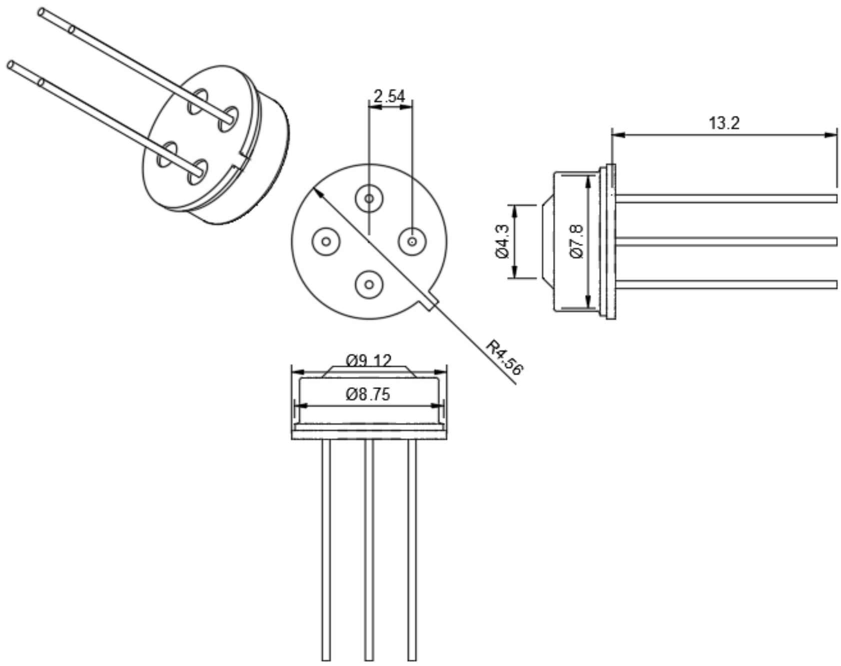


Figure 12. Mechanical Drawing of IR Temperature Sensor

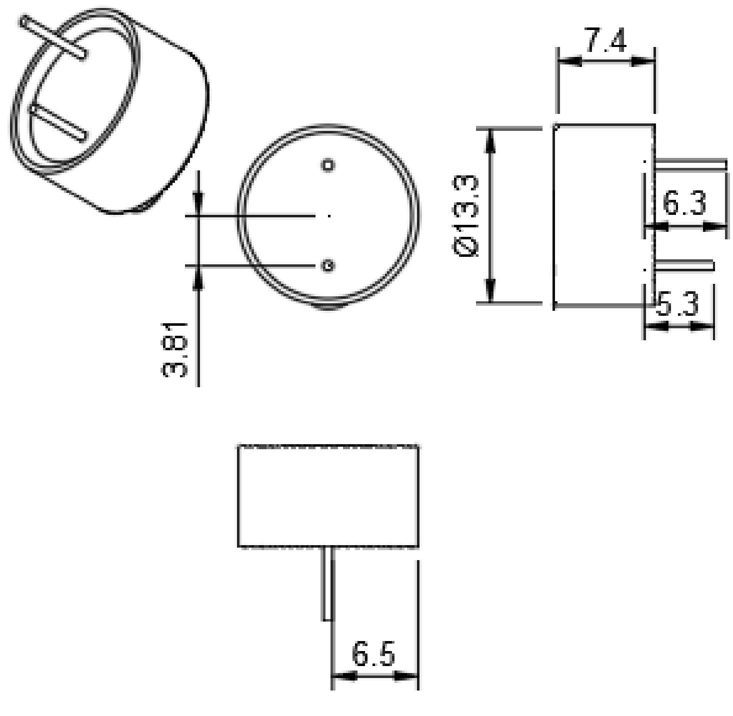
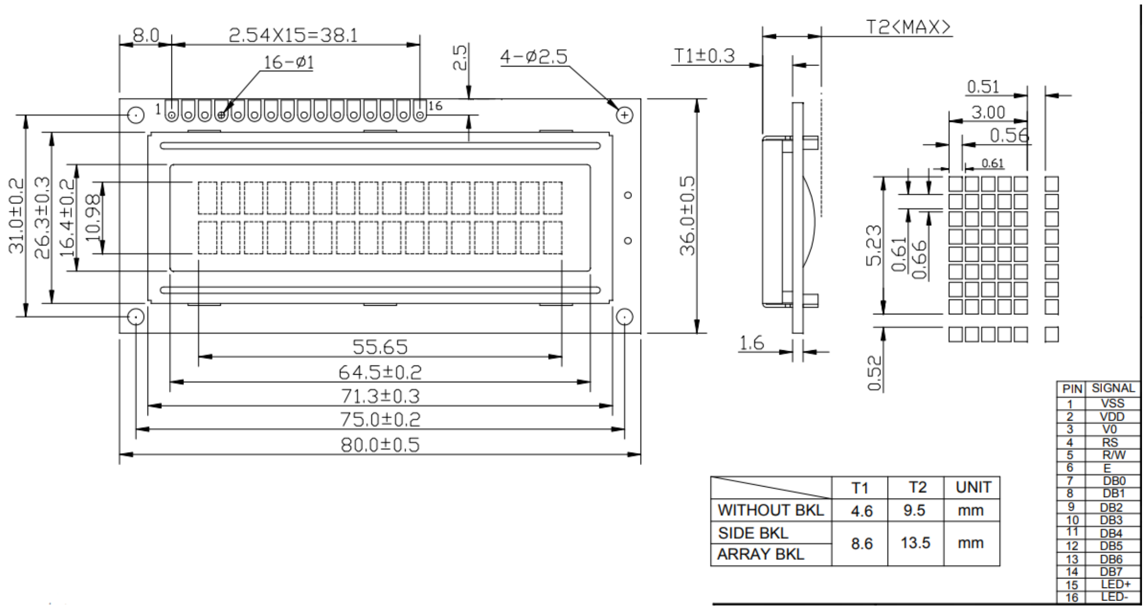


Figure 13. Mechanical Drawing of Speaker



	T1	T2	UNIT
WITHOUT BKL	4.6	9.5	mm
SIDE BKL ARRAY BKL	8.6	13.5	mm

PIN	SIGNAL
1	VSS
2	VDD
3	V0
4	RS
5	R/W
6	E
7	DB0
8	DB1
9	DB2
10	DB3
11	DB4
12	DB5
13	DB6
14	DB7
15	LED+
16	LED-

Figure 14. Mechanical Drawing of LCD1602 Module

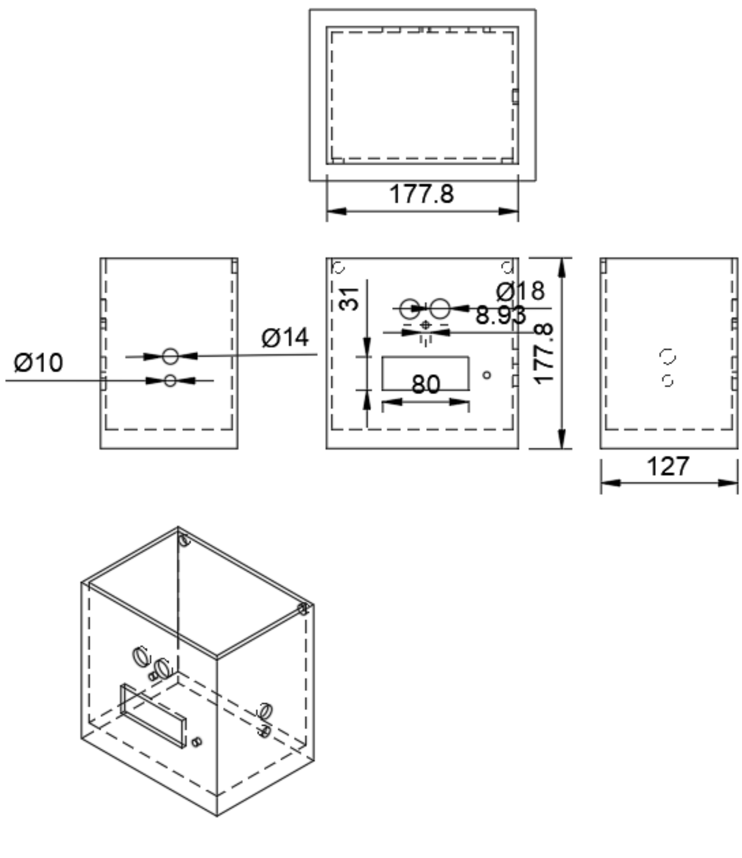


Figure 15. Mechanical Drawing of Enclosure

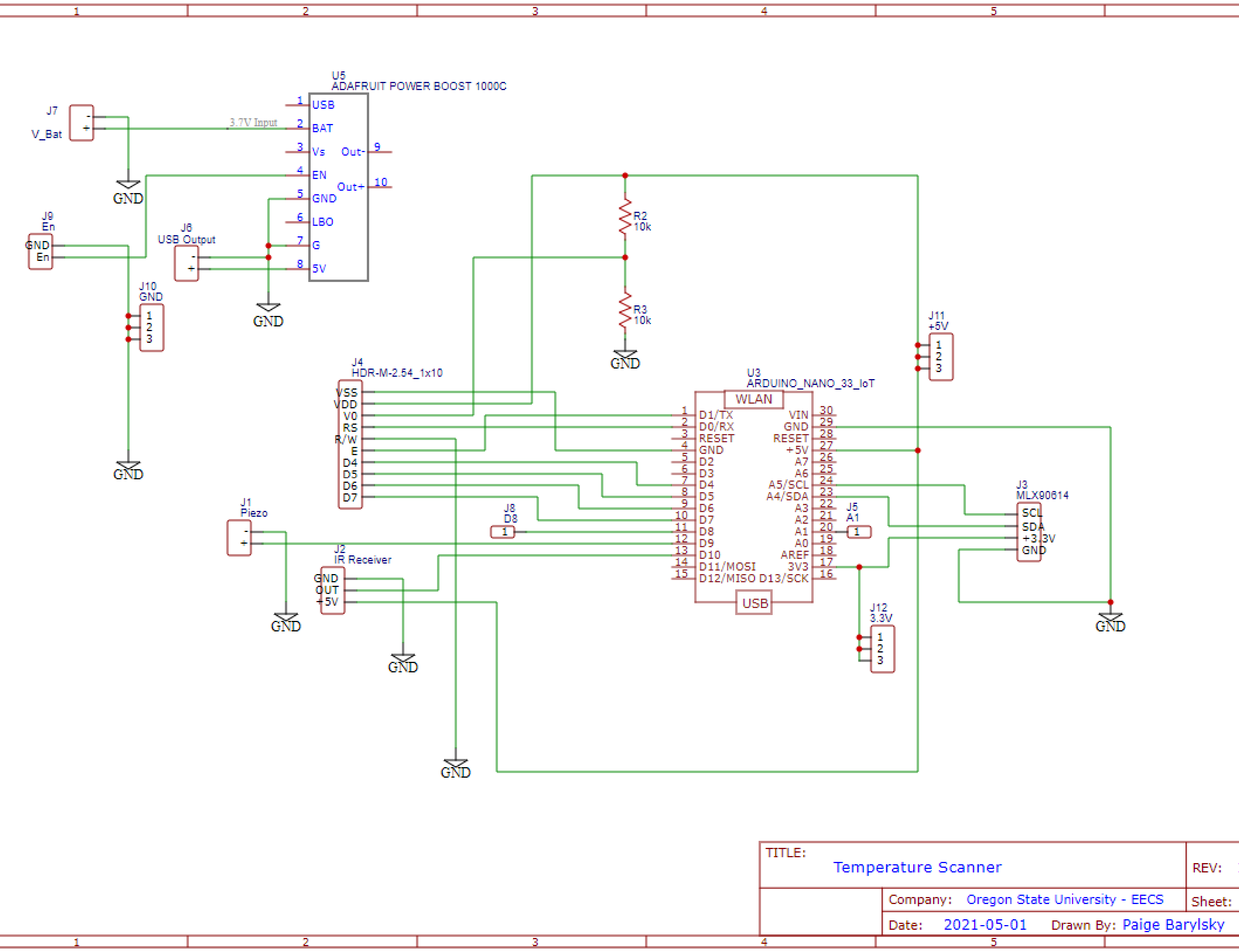


Figure 16. Interconnect Schematic of PCB

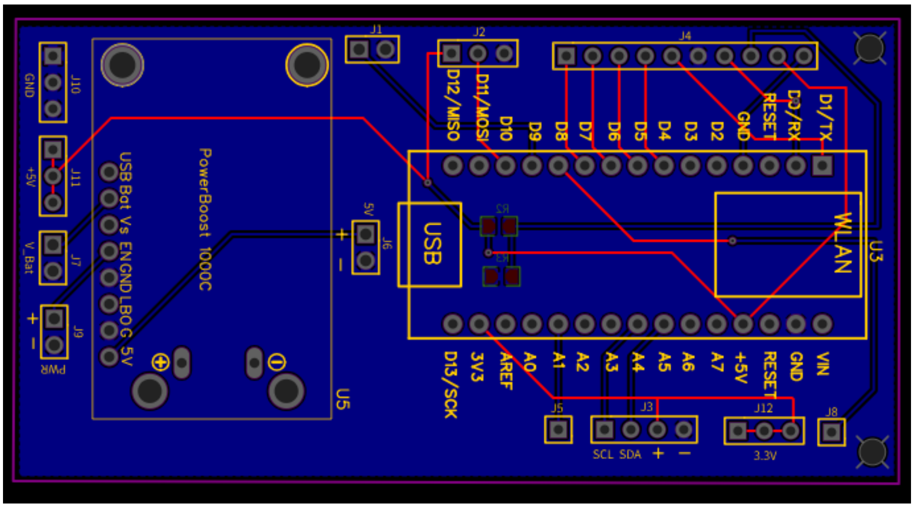


Figure 17. Layers and Traces of PCB

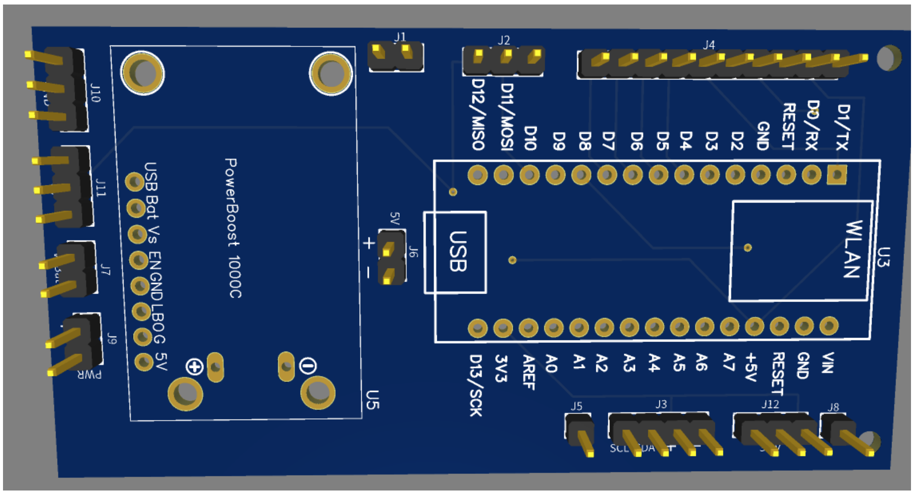


Figure 18. 3D Model of PCB [Front]

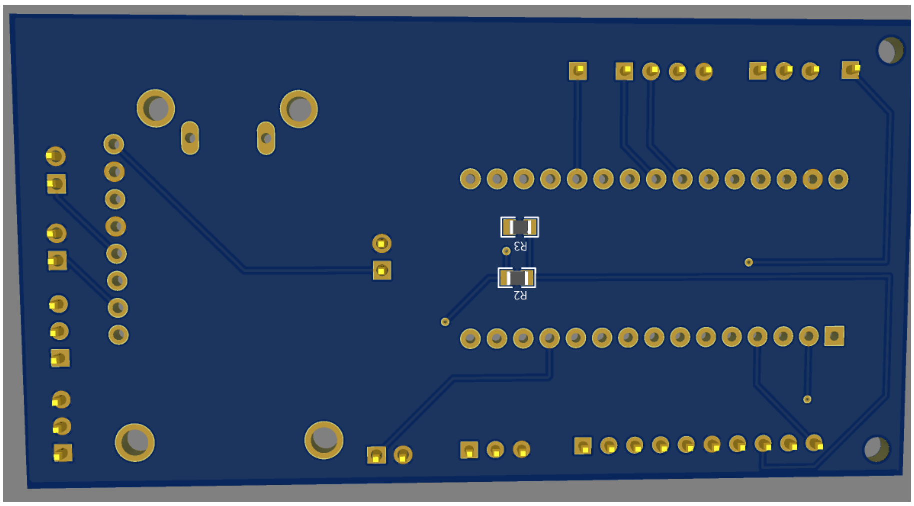


Figure 19. 3D Model of PCB [Back]

6. Part Information

In order to reconstruct the project in its exact form, one should consult and make use of the following materials found in the parts list. This parts list contains both materials used to operate the device, and those used to build the final enclosure. The list is split into 3 different tables in this document in order to fit all of the information about the different materials in an organized way. Materials follow the same format in each table, and can be indexed using the material column in each. Table 3 of the parts list contains information about the schematic designations, name of the components, quantity used, and the material and total cost of the components. Table 4 of the parts list contains manufacturing information, such as the number and manufacturer name, and the value of any components used electrically, such as resistance values. Table 5 of the parts list focuses on package size information in different dimensions.

Table 3. List of Parts Table 1: Designation, Quantity and Cost

Material	Designation	Name	Quantity	Unit Cost
1	A1	Arduino Nano 33 IoT	1	$18.40
2	U1	Non-Contact IR Temp Sensor	1	$13.88
3	U2	IR Receiver Diode	1	$1.95
4	N/A	IR Transmitter Remote	1	$5.29
5	DS1	LCD1602 Module	1	$5.99
6	U3	HC-SR04 Ultrasonic Sensor	1	$3.95
7	N/A	Primary PCB	1	$20.00
8	U4	BKM-06 Piezo Speaker	1	$0.88
9	N/A	12mm Sande Plywood Plank	1	$17.29
10	N/A	Titebound Ultra Wood Glue	1	$8.29
11	R1	100Ω Resistor	1	$0.06
12	R2,R3	100kΩ 0805 1/8W Resistor	2	$1.34
13	N/A	Jumper Cables	30	$2.92
14	N/A	Micro-USB to USB-C converter	1	$3.50
15	N/A	Header Pins	40	$0.99
16	N/A	4x6 Prototyping Boards	3	$0.99
17	N/A	Flex Paste	1	$12.88
				Total: $118.60

Table 4. List of Parts Table 2: Manufacturing Information and Value

Material	Manufacturer Number	Manufacturer	Value
1	ABX00027	Arduino	N/A
2	MLX90614	Melexis	N/A
3	TSOP38238	Vishay	N/A
4	B01EE4VXS0	DAOKI	N/A
5	LCD1602	WaveShare	N/A
6	HC - SR04	Sparkfun	N/A
7	N/A	Osh Park	N/A
8	BKM06	Murata	N/A
9	N/A	Home Depot	N/A
10	N/A	Titebond	N/A
11	299-10K-RC	Xicon	100Ω
12	CRCW0805100KJNEAC	Vishay	100kΩ
13	GK1212827	*Z&T*	N/A
14	B07VBV1PY5	KUXIYAN	N/A
15	B07PKKY8BX	MCIGICM	N/A
16	DY1807	DEYUE	N/A
17	PFSWHTR16	Flex Seal	N/A

Table 5. List of Parts Table 3: Package Size and Datasheet Links

Material	Package Size (L x W x H*)
1	18 x 45 mm
2	19.15 x 11.7 mm
3	8.25 x 5 mm
4	104.9 x 51.1 x 23.1 mm
5	36 x 80 x 14 mm
6	45 x 20 x 15 mm
7	85.3 x 44.9 mm
8	38 x 38 x 30 mm
9	1/2 in x 4 ft x 1 ft
10	N/A
11	6 mm
12	2.0 x 1.25 x 0.5 mm
13	160 mm
14	147 x 88 x 6 mm
15	2.54 mm
16	101 x 152 mm
17	N/A