EE 41430

SmartLint©

Final Report

Group 3: SmartLint
William Bailey, Aaron Diaz, Ryan Frost, Rene Frank Gahitira, Salvador Lort, Grayson Zinn
April 28th, 2022

Table of Contents

1. Introduction

The Problem

Every year, over 15,000 clothes dryer fires are reported each year — causing an estimated total of 5 deaths, 100 injuries, and $35 million in property loss. An estimated 34% of these fires can be attributed to excess lint build-up in the dryer's ventilation system — making it the single largest cause of dryer accidents and related property damage. Older dryers are still prevalent in many homes and apartment buildings and have no mechanism for detecting and/or communicating if and when the ventilation pipe is clogged with lint. Cleaning the dryer lint trap regularly is necessary to decrease the risk of fires, but lint can also build up in the venting pipe and/or system without the owner's knowledge. Moreover, the efficacy of the dryer's ability to dry clothes effectively in a single cycle and to dry larger loads can be impacted by a lint blockage. This problem is further complicated by households and multi-family living arrangements in which many different people use the dryer and may not know when or where to check for lint build-up. The problem was brought up by another student, Michael Pitz, when our Senior Design class was conducting a brainstorm of ideas for projects.

The main issue from this data was determined to be an inability to see into the dryer lint pipe to determine if there was a blockage. The build-up of dryer lint in the removable trap could be an issue, but this is easy to check and does not directly lead to the ignition of fires. Thus, the main issue is finding a way to alert the dryer owner when cleaning is needed in the exhaust pipe. Doing this would require a device that could infiltrate the pipe and determine its effectiveness and risk of fire while still keeping the pipe connected as a closed system. As well, an issue arises as to how best determine the effectiveness of the pipe and the risk of fire. The obvious choices are determining if air is still flowing at the rate that is expected and seeing if the temperature of the pipe is high enough to cause the ignition of a fire, and if the pressure build-up is indicative of a threat. With this information, there would need to be an alert system for the user to interact with that can easily inform them if any of these measurable aspects are indicative of a lint build-up or blockage.

Overview of Solution

To solve this problem, we created a device that will detect and alert the dryer's owner (whether it be a homeowner, landlord, or renting tenant) when it believes the lint in the dryer ventilation pipe is reaching a critical level. First, we designed a method to indirectly measure the amount of lint (without seeing or collecting it) and then communicate to the user if there is a critical amount of lint that needs to be removed. The methods to measure the lint include the use of two sensors: an FS3000 air velocity sensor and a BME280 sensor (capable of sensing temperature, pressure, and humidity). These values are then checked against an acceptable tolerance, and a determination of risk level is made based on how far within (or beyond) tolerance the combination of these values are. Without the opportunity to produce our own experimental data, we were unable to determine if there was an effective correlation between humidity levels and lint ignition, so we chose to only measure and use values for pressure, temperature, and airflow. Each of these values was assigned two thresholds: an initial and a critical threshold. For airflow, the dryer is operating optimally if the airflow measured is above the initial threshold and is inhibited if the airflow is below one of the two thresholds. For pressure, the dryer is operating optimally if the pressure measured is below each of the two thresholds and is inhibited if it is above either. The same can be said of temperature.

If all of the measured values are considered to be within tolerance, the dryer pipe is operating in the green mode and it is considered business as usual. During green mode, the website and LED will display this color and let the user know there is no issue, and the dryer is running. If one or more of the values that are measured pass an initial threshold, the dryer ventilation pipe will enter yellow mode. During yellow mode, the LED is yellow and the website will let the user know that there could potentially be a hazard and the dryer is not quite operating at its best potential. No alarm will be sounded. This mode does not mean the dryer should be immediately turned off, but does mean the user should disconnect and inspect the pipe manually at their earliest convenience and monitor the measurements. Finally, if the values are considered immediately worrisome, the dryer pipe is in the red mode. During this state the piezo alarm will sound and LED show red to alert the user that immediate hazards will occur if the dryer continues operating and should be turned off and checked immediately. The general requirements for this mode are that at least one of the measurements (or more) are beyond the critical tolerance and the dryer should be turned off and checked for a lint blockage as soon as possible.

The mechanisms to alert the user consist of multiple communication mediums: audibly via an alarm or buzzer, visually with an LED located on the device, and a website that is updated with the dryer's status and sensor data that is accessible via WiFi. Combined, these mechanisms give users multiple ways to safely check the dryer tubing and avoid a fire.

Reflection

Looking back on the project, the design is nearly identical to expectations. It consists of a sensing system to collect values that are then analyzed to alert the user of a problem. The design itself did not change a ton over the course of the project time, as the main idea and overall design stayed the same. The main uncertainties came in what type of sensors would sense what, and we answered these by finding two popular, reliable sensors to measure pressure, temperature, and airflow. The system could be said to exceed expectations about accuracy, as it was an issue determining how to weigh what data was received. Rather than simply measuring one aspect as was originally expected, three determining factors were measured so the problem became what values to say are fine and what factors to value the most when determining risk of a fire. However, the use of multiple factors makes the system more accurate.

The biggest challenge by far of the project came near the end when the permanent board came in and was finally being implemented with the code tested on the breadboard. There was an issue with the board having different pins designed for I2C on our board than the default pins specified by the ESP32 libraries that powered our programs. Several hours were spent trying to remap the pins, which on the surface seems like a simple issue. The issue was finally resolved when it was realized that the pins labeling on the board was different from how the pins were represented in the actual code itself. After using the “Wire” object to remap the pins using a couple of lines, our existing code was able to interface again and the two sensors connected without any issues. A second issue arose when we had a memory allocation problem arise on the ESP32 after that. We were able to resolve this by reverting to an earlier version of the code

One possible change from the planning was the question of whether to measure lint in the vent itself or in the lint trap. After some research, it was evident it made more sense to check the vent, as the lint trap is easily accessed and thus most fires start due to a block in the vent itself that restricts airflow out of the house or unit.

2. Detailed System Requirements

Success was judged off of the overall system's ability to reliably detect excessive amounts of lint in the pipe attached to the back of the dryer and its ability to successfully alert the user. While it was considered infeasible to simulate and test the environment of an actual dryer ventilation pipe (and the thresholds at which a fire would start from the ignition of lint), it was still considered successful due to its ability to accurately detect changes in the test environment. The strategy was to detect when the level of lint in the dryer pipe or filter causes the dryer to be at risk of a fire as a result of the lint igniting. This was accomplished by measuring the airflow through the dryer pipe, the temperature on the surface of the pipe near the entrance to the dryer, and the pressure in the pipe. This was done using a temperature sensor, a pressure sensor, and an airflow sensor, which lives inside the vent pipe to sense and send data back to the microcontroller. The exact levels which are considered safe levels are based on design specifications from dryer manufacturers. However, due to exhibition of this project being in an environment without conditions similar to a pipe, these levels were adjusted to allow complete demonstration of the features in a classroom setting.

The board itself needed to take in the temperature and airflow readings and then decide if these are within a safe range. If the logic sees a problem with either, it will then begin the alert protocol which will entail sounding the alarm and sending out notifications. It was intended to determine and develop thresholds that are conservative and alert the user when the lint trap or the pipe need to be cleaned out, as well as more dire thresholds that recommend the user refrain from running the dryer at all, for instance. This means the board also needs WiFi capabilities to send out an appropriate message to the user. The only physical user interface on the device is a power plugin and a reset button for when the user wants to turn off the alarm. This is a reset to return the board to its original state and ensure that if there's a bug it can be fixed. The board and sensors are all powered via a wall wart that converts the power to a USB mini or USB micro that can be easily taken to power a microcontroller. Unplugging this USB will power off the board.

As mentioned, the device itself is powered via a wall adapter. This then interfaces with a protective casing, which houses the board, microcontroller, WiFi module, and other wires that protrude to connect to the sensors. It has a simple, convenient hardware user interface on the outside that consists solely of a switch and an audible alarm. Out of the casing, wires extend down to connect the temperature and airflow sensors. These sensors are on a breakout board that connects back to the main board. These boards are attached via epoxy to the dyer sleeve. The airflow sensor insertion is sealed off to prevent airflow from leaving the tubing, by simply using epoxy and the natural barrier of the external casing. There is no safety hazard from the wall adapter, thus the system is not grounded. The voltage from the adapter is much too low to cause any safety issues.

There are no weight requirements, as weight of the device is not an issue as it is immobile and mounted on a relatively robust structure. The sensors are small and light enough as well as out of the way on the side of the pipe to not interfere with the lint’s ability to flow through the vent pipe. The sensors are already small, thus making our casing for the sensor small as well. The case is sized to fit the minimum height, width, and length requirements of the board and any protrusions from its attachments. Besides an area for the board, there is about 8 mm of plastic encasing the board. The alarm is mounted underneath holes cut into the top of the SmartLint casing to make it more audible to the outside world. Since our proposed design interacts with the dryer directly, all our materials and sensors are able to withstand 250℉ without experiencing any malfunctions.

3. Detailed Project Description

3.1 System Theory of Operation

The system operates on the principle that increased access to data will allow a user or homeowner to reduce the risk of a fire that could be damaging to life and property. The designed device enables the collection and analysis of data, then making a determination if that data indicates a safe or unsafe condition within the pipe. Starting with the FS3000 sensor and BME280 sensor, these take data that includes temperature, pressure, and airflow speed. Next, using an I2C interface, this data is sent back to the ESP32 on the main board. This data is then processed using the programmed logic, mostly consisting of a series of if-statements that continuously check the dryer's state and the sensors' measurements against what our research determined to be acceptable operating conditions. This process is continuously repeated to keep checking the dryer while it is operating. If there is no concern, the system is in the green mode. If there is a possible concern but not enough to be urgent, it would be in the yellow mode. Finally, if the conditions present an immediate safety concern, this would constitute the red mode and an alarm would sound, hopefully prompting the user to stop their use of the dryer and inspect their ventilation pipe. Based on these modes, the system outputs to the website so the owner knows the conditions their dryer is operating at. The system also changes the color of the LED accordingly based on what mode the system is operating in. Additionally, if the system is in the red mode the piezo will be turned on using a digital 'high' signal for a certain amount of time (roughly a second). This will serve as a way to alert the owner without having to check the website. The system can also recognize if the dryer is on or off, and will turn the LED off completely if it is off. The website will also display a message if the dryer is completely off. One roadblock with this system is that there has to be a way to determine when the dryer is turning on/off or running in general. Otherwise, the airflow will be read as below the critical threshold and the readings will make the ESP32 think the dryer is in the red mode, sounding the alarm, when the dryer is actually just off or starting up / cooling down. To fix this, the logic gives some leeway when the airflow is sensed as speeding up or slowing down in a fashion that would resemble a dryer turning on. During this time, the logic will assume the dryer is off using a delay and thus not falsely alarm the owner. Also, for the example of airflow it must meet the minimum requirement of ~3.5 miles per hour for the dryer to register as operating. Otherwise, it will be assumed that the dryer is not running and is simply off as even with lint blockage the airflow would still register over this.

Looking towards the non-electrical hardware components, the casing encompasses the whole system into a nice-looking block that is easy to attach to a dryer pipe. For this project, an intermediary section of pipe is also part of this system as it is easiest for the customer to simply attach this pipe into their existing exhaust system. The case has two main parts, one that rests against the pipe and holds the board, and a top part that covers the board and presents a logo to the user, as well as has holes to allow the noise of the piezo to permeate through. Overall, this part of the system is simply to hold all the electrical hardware in the proper places as well as protecting it.

3.2 System Block Diagram

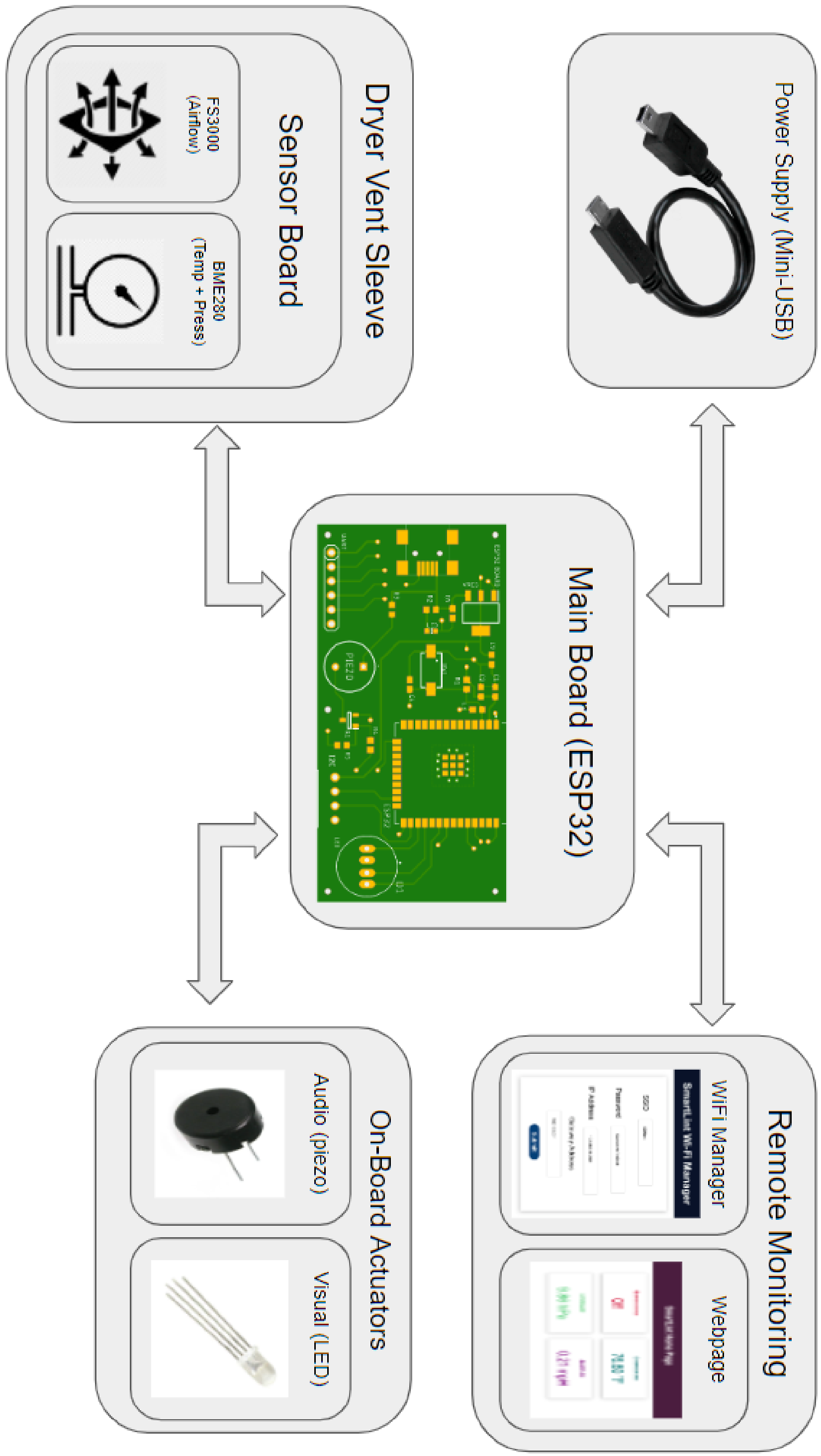


Figure 1. System Block Diagram

Looking above at Figure 1, the system block diagram, everything stems from the ESP32 microcontroller. The ESP32 contains all the logic and is able to run the system entirely, processing all inputs and sending the corresponding outputs. Soldered directly onto the board are the two on board actuators, the LED and piezo, for visual and auditory stimulation of the user. Their operation and usage has been explained in previous sections.

Next, looking at what takes the actual inputs from the dryer sleeve, we have the Sensor board. Attached to the sensor board are the two sensors that are actually inserted into the sleeve, the FS3000 and BME280. These communicate with the main board through an I2C interface and supply the relevant data on the dryer conditions to the main board for processing.

The main board, and thus the whole system is powered through a mini-USB power supply, which can be powered through any stepped-down wall adapter or USB port. Finally, we have the remote monitoring system. This is wirelessly connected to the main board, and creates the ability for the ESP32 to send data over WiFi to the device's website — outputting the measurements received and the state the dryer is operating at.

3.3 Detailed Operation of Main Board

The main board is a two-layer ESP32-centered board with all the components on the top layer and a ground plane on the bottom layer. Since our system operates in a high-temperature environment, we are using a high-temp ESP32-WROOM-32E with 4MB memory since we do not need a significant amount of memory storage. In addition, the microcontroller's center has 12 vias under its center that connect it to the ground plane on the bottom layer of our board to ensure adequate heat transfer when operating. The microcontroller operates at 3.3V, which is supplied by the voltage regulator that inputs 5V from the USB-Mini port and outputs 3.3V. As part of our power supply, there is an 0603 red LED that serves as a physical indication of our board's ability to receive power.

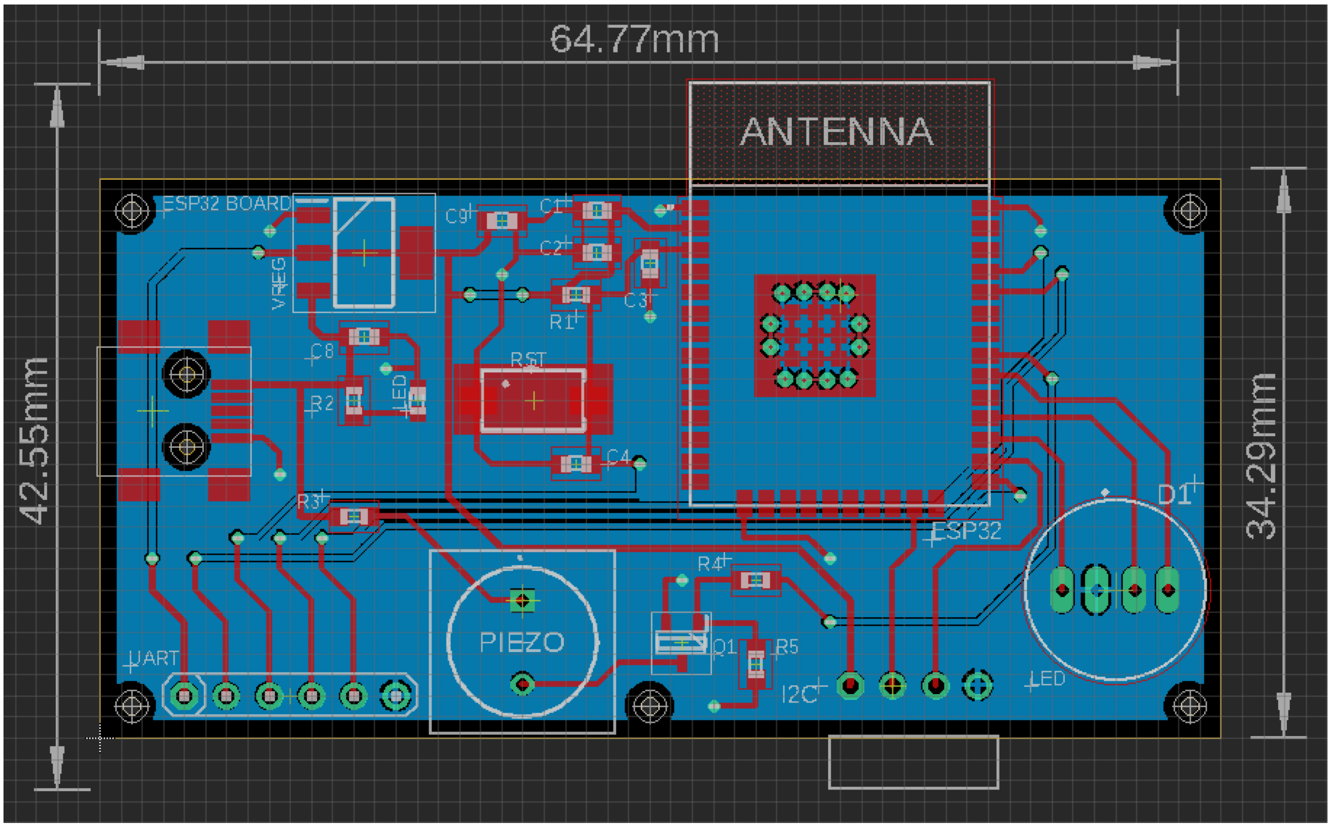


Figure 2(a). Main board PCB layout

When powered up, the main board has a reset button that connects the EN pin to ground when pressed, effectively resetting the ESP32. This board houses our physical means of communication with the user, namely the piezo and the four-pin LED that is capable of emitting both red, green, and blue light. The piezo's circuit is a simple power-switch circuit with the micro serving as the enabler that turns on the piezo when needed. Upon resetting or starting up the device, the setup() function will sound the piezo once to indicate that the device is about to enter operating conditions.

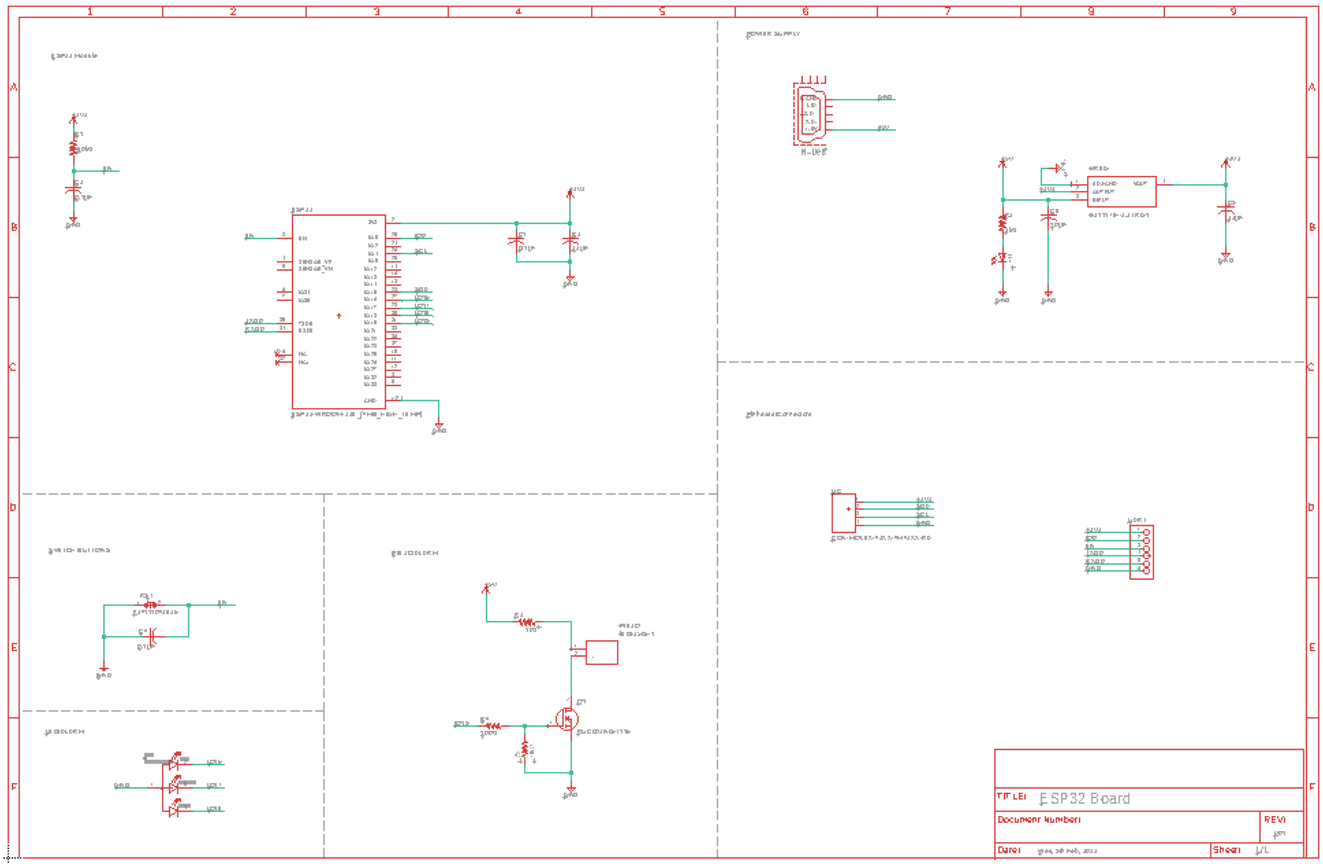


Figure 2(b). Main board schematics

Since the ESP32 has both UART and I2C, we use the UART interface to communicate with a USB-UART bridge designed by Professor Schafer to establish communication between a computer and the main board. The I2C interface is used to communicate back and forth with the sensors' board. Both of the connectors to these interfaces are placed on the same side of the board for easy access.

3.4 Detailed Operation of Sensing Subsystem

The sensor's board is also a two-layer board with two sensors on the top layer and a ground plane and connector on the bottom layer. For effective lint build-up monitoring we use the FS3000 as an airflow and pressure sensor and the BME280 as a temperature sensor. Both sensors collect the needed data independently and send the data to the main board via a 4-pin I2C connector located on the bottom layer of the board. The relevant signals here are power (3.3V), data (SDA), clock (SCL), and ground (GND).

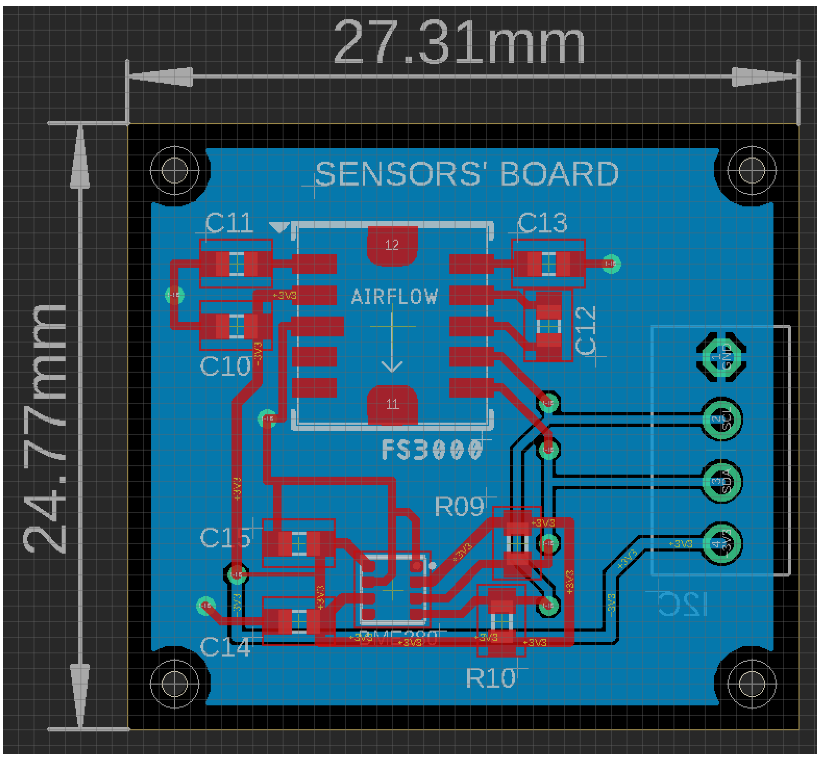


Figure 3(a). Sensors' board PCB layout

Both sensors' independent circuits are designed according to their datasheets, which determines the values of the decoupling capacitors used in both circuits to minimize noise and ensure proper functionality. Since both sensors are connected to the same I2C interface, the sensors' board only needs two pull-up resistors instead of four: one on the

SDA line and another on the SCL line. The resistors ensure that our data and clock lines are pulled up when not being used and pulled down whenever there is communication between the ESP32 and our sensors.

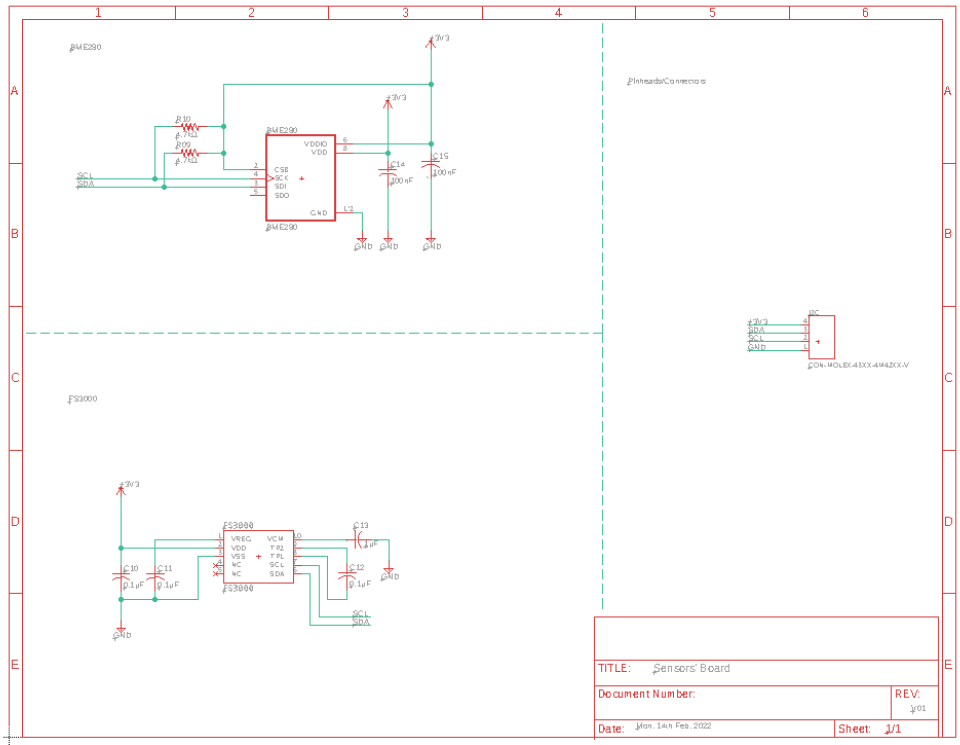


Figure 3(b). Sensors' board schematic

This breakout board is attached to the pipe through a heat-resistant epoxy that protects the wiring and the plastic connector, then a hole is drilled to run the wiring back to the main board. This should be enough to hold the breakout board in place when the dryer is running.

In order to program the BME280 sensor, we followed this tutorial from RandomNerdTutorials. In order to program the FS3000 sensor, we followed this tutorial from Sparkfun.

3.5 Detailed Operation of WiFi Subsystem

In order to connect our board to the user's home WiFi network, there are a number of steps that must be undertaken by the user. First, the ESP32 acts as an Access Point (AP), in which it emits a WiFi signal that allows a connection to be established with another device in its proximity. In this case, the device that connects to the ESP32 will be a user's smartphone, laptop, or other internet connected device. The user will navigate to the 'SMARTLINT-WIFI-MANAGER' network, which will appear in the list of available networks. Then, the user will be prompted to enter the SSID and password to their home network. Crucially, this will be "remembered" by the ESP32 by using SPIFFS (or SPI Flash File Storage, the on-chip memory). This way, in the case that the board loses power for any reason (such as in a power outage), the user's home network will remain stored in the ESP32 so that it can reset and reconnect to the WiFi network without the user needing to re-program the board, because our product is meant to sit at the back of the dryer vent for a long time (for many years, potentially). We used this tutorial from the website RandomNerdTutorials to set up the WiFi manager. Here is a system diagram of how the WiFi manager process works:

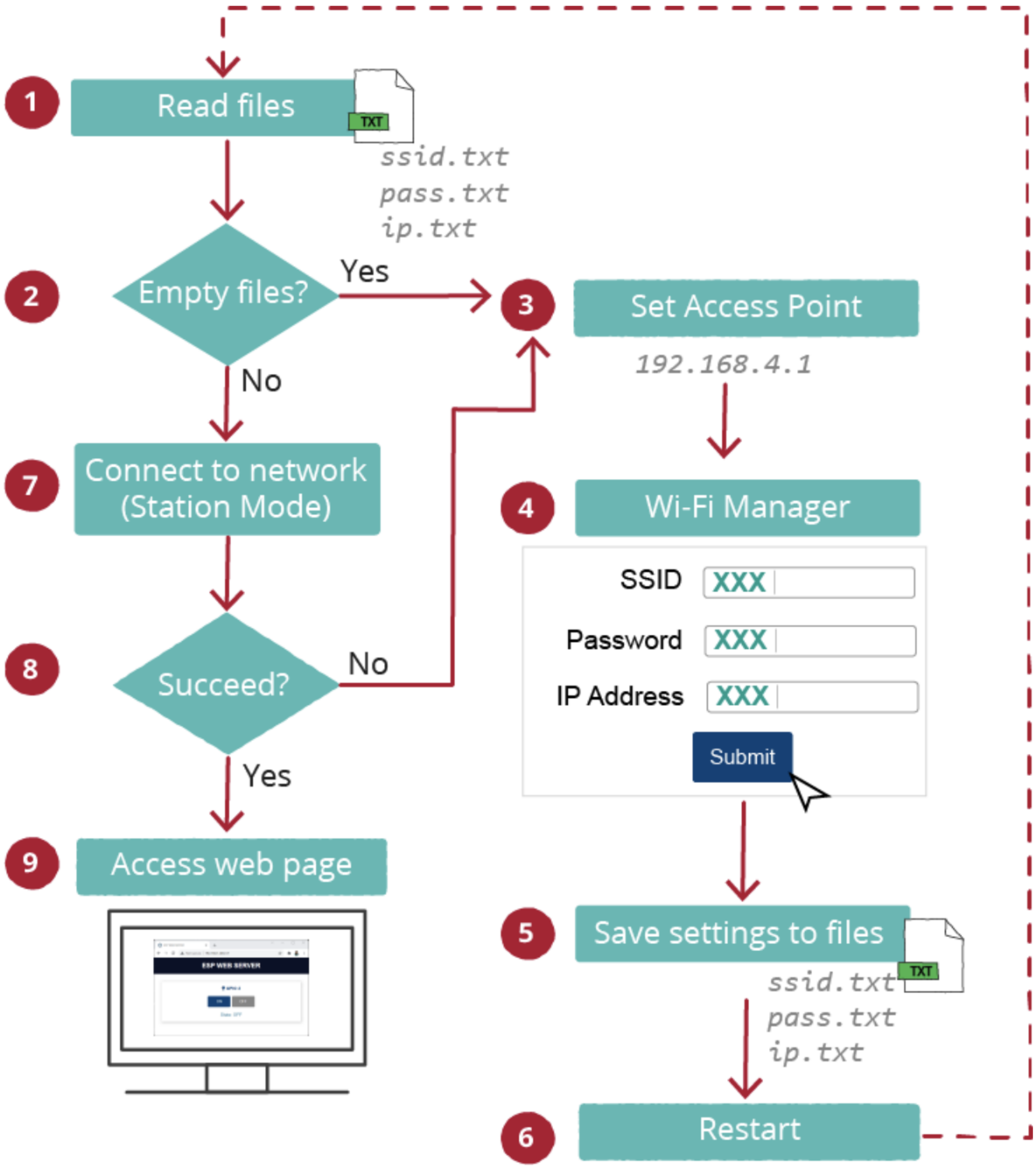


Figure 4. WiFi manager process

After the device restarts, SmartLint is up and running and the user can navigate to an IP address (192.168.0.200) on a device that is also connected to their home network in order to view the user interface and the measurements from the sensors and the status of the dryer.

3.6 Detailed Operation of Casing

The device's casing is a stationary, 3D printed outer shell that helps hold the hold system together on top of the pipe. It protects the circuit board while also allowing the user to interface with the reset button, see the LED, and hear the alarm. It also allows the sensor board to protrude and insert into the pipe where it can make readings, as well as allow the power cord to plug into the main board. The choice of material was based on what was available in the EIH to allow our design to be easily 3D printed. Thus, the final product was simply black plastic. The general shape is designed based on what can easily fit around the pipe while still looking nice and holding everything together. The case is two separate parts, with a bottom part to make contact with the pipe and hold the board, and a top part to protect the board and create a nice aesthetic for the product. The most difficult part for the casing was trying to figure out where to cut holes in the top to view and hear the LED and alarm respectively since the board was constantly being changed. There was little to reference for spacing so that was quite the challenge. Testing was simply taking the final 3D printed casing and dry fitting it against the pipe and rest of the system to ensure it all fit together properly.

Shown below are Figure 8, the top part of the casing, and Figure 9, the bottom part of the casing. The bottom part is designed to fit around the cylindrical pipe while still allowing for holes for sensor insertion in the bottom, a hole in the side for a power plug in, and attachment holes for the top casing. The top has holes for the reset button and LED, as well as three small holes to allow the alarm sound to exit. As well, a SmartLint logo can be seen at the bottom.

Figure 5, Top of casing

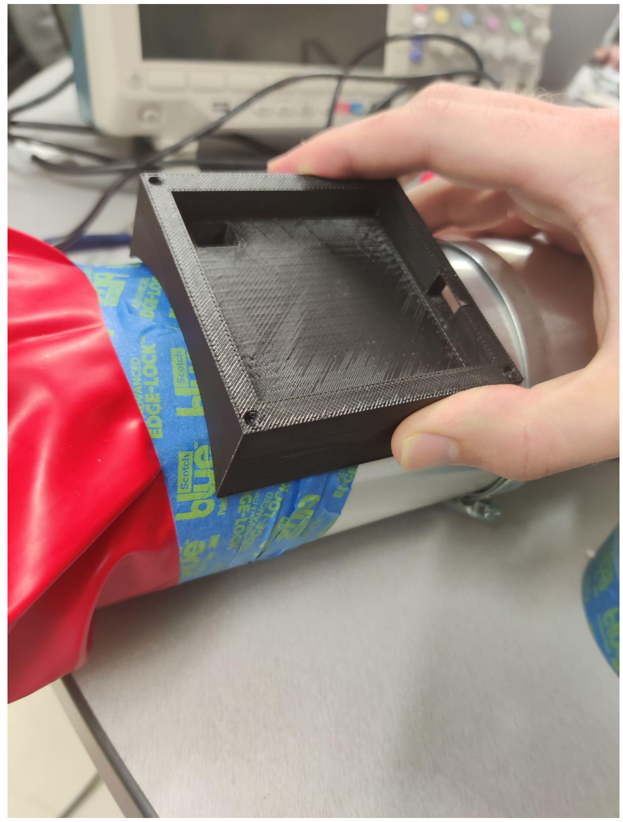


Figure 6, Bottom of casing

3.7 Interfaces

The user interfaces include both the physical interfaces of the system and the website. As far as physical inputs, there is the reset switch and the ability to unplug the system. These are both used to reset the system in case of an error, as described in the user manual. For physical output, the board is equipped with a piezo buzzer and LED. The LED will light up green, yellow, or red based on the readings taken from the pipe. Green means the pipe is operating well, yellow indicates there could be a problem soon, red indicates there is a serious risk of fire and likely blockage, this will also sound the alarm. The casing is designed to allow easy access to the reset button so the user can troubleshoot, as well as allowing the LED and piezo to alert.

Looking at the website part of the interface, this website gives what values are being read for pressure, temperature, and airflow speed. Also, the device will indicate what the risk of dryer operation is (based on the readings from the three sensors) and what the corresponding risk of lint ignition is. It also indicates if the dryer is running or not based on the level of airflow detected by the FS3000 air velocity sensor; if the reading surpasses ~3.5 mph, the dryer is thought to be running and readings will occur roughly every second. If the reading is below that threshold, the device perceives the dryer to be off and it will only check for readings every 10 seconds. This allows for the dryer to get up and running in those 10 seconds so that the sensor doesn't mistake the low levels of airflow for a lint blockage when there is none present.

4. System Integration Testing

4.1 Description of Integration Testing

The main way the integrated system was tested was using a hair dryer because it provides heat and airflow. The hair dryer was able to simulate the warm air at a velocity similar to an actual dryer. This proved to be an effective way to test the system without hooking it up to a dryer pipe as it was able to get up to very high temperatures and provide direct airflow over the sensors. However, this was probably most lacking in the pressure department as it was not a closed system like a real dryer vent. We had a hard time successfully recreating the operating pressure conditions of a dryer under a lint blockage.

To solve the lack of pressure, the back of the pipe segment was taped off and a balloon was used at the front of the segment to create an airtight seal around the dryer. Using this method, a more accurate, closed system was created and pressure was properly increased, meaning all 3 conditions were simulated. With this setup all system requirements were able to be tested and verified.

4.2. How Testing Achieves System Requirements

The hair dryer test was able to effectively show that the system could take readings of temperature and air flow and effectively make a judgment as to how safely the system is operating, then output that as an LED color as well as transmit to the main website. The system was able to detect the airflow and heat from the hair dryer, then transition into green mode as it recognized the dryer would be turning on. Next, as the heat began to get too high from the threshold it transitioned into yellow mode, indicating a potential issue. When the heat was turned on extremely high, well outside of the acceptable range, it finally transitioned to red mode and the alarm went off. Finally, when the hair dryer was transitioned off, the system was able to recognize this and turn itself off, returning to its original state before the hair dryer was turned on.

During this time, the website was able to output accurate values for temperature, airflow, and pressure. The website also transitioned from indicating the dryer was off to each color state the dryer was operating in. This showed a successful communication between the hardware alert system and the digital alert system. Thus, the system requirements were met in a makeshift test using a hair dryer. If it were possible, our next step would be to test the system requirements on an actual dryer to finalize the product. With this testing we would also be able to replicate the installation process the user would have to do when installing the SmartLint.

5. User Manual and Installation

<u>How to Install SmartLint</u>:

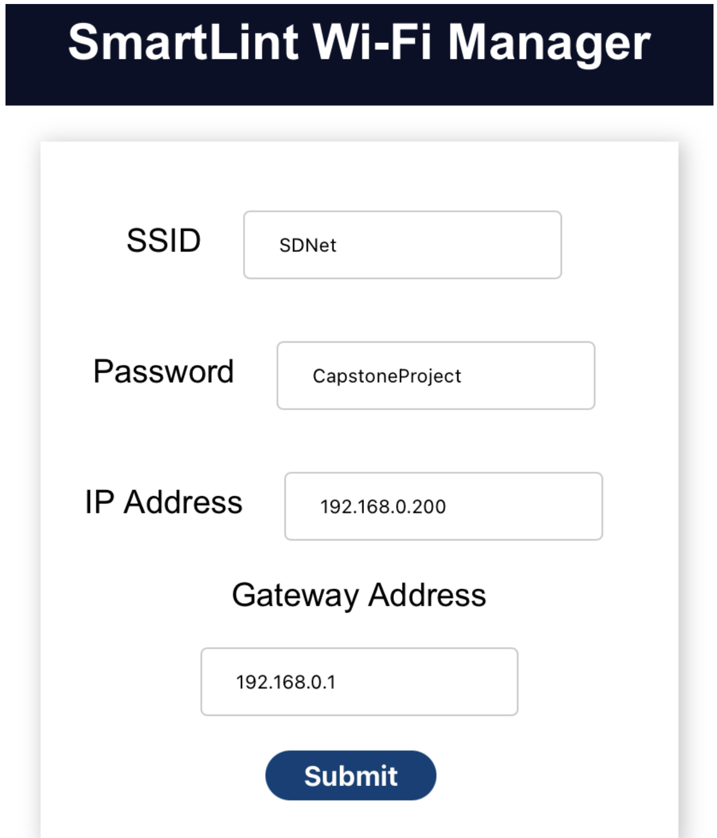
The SmartLint system comes with its own tubing, enabling easy installation between the back of the dryer and the start of the ventilation hose.

1. Locate a disconnect point between the dryer sleeve in the dryer.
2. Pull off the dryer sleeve and use the SmartLint system to connect back to the dryer.
3. Attach rings included with SmartLint system to ensure tight connection.
4. Plug in the SmartLint system into a wall outlet.

<u>How to Connect SmartLint to Local Network</u>:

SmartLint is designed to run on your home WiFi network. The device's readings and status can be accessed through an IP address specified in setup. Once the device has power:

1. Navigate to the available WiFi networks on your desired device.
2. Select the network named "SMARTLINT-MANAGER".
3. Navigate to the device's IP address using a browser: 192.168.4.1
4. The webpage will look like this:



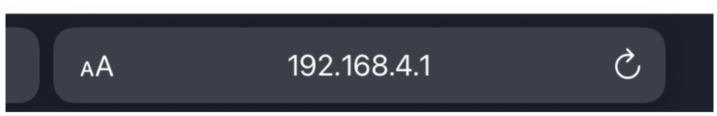


Figure 7. WiFi manager interface

Enter the SSID of your home WiFi network into the field labeled "SSID"

5. Enter the password of your home WiFi network into the field labeled "Password"

6. Your SmartLint device should now be connected to your home network, access the device's measurements and status by visiting the IP address listed on the previous webpage in the field labeled "Gateway Address". The IP address provided should be 192.168.0.200.
7. The webpage will look like this:

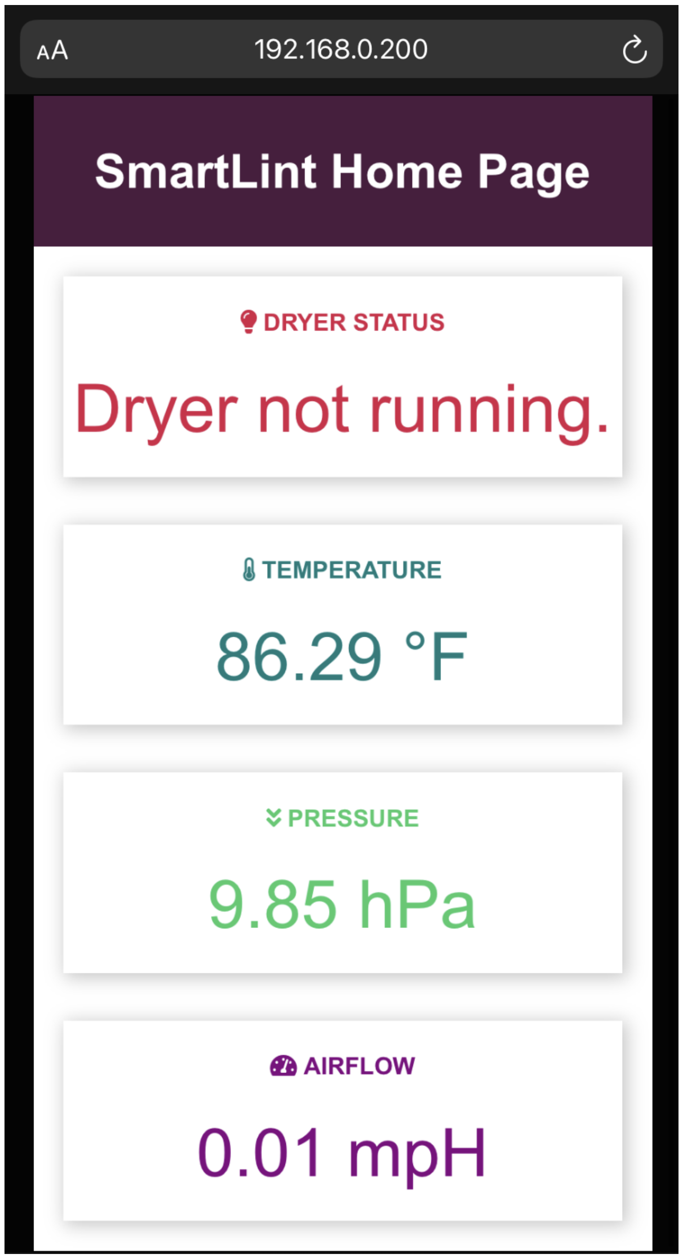


Figure 8. SmartLint user interface

How to Use SmartLint:

SmartLint has two main ways to alert you to your dryer's status:

1. Physical cues, including an LED colocated with the device to indicate the status (colors are green, yellow, and red) and a piezo alarm that will sound in the case of a concerning sensor reading.
2. The SmartLint device's website, which can be accessed at any time and will provide the user with readings for temperature, pressure, and airflow at the back of the dryer, if the dryer is on or off, and an alert message to the user if an issue is detected.

When the dryer is functioning properly and neither the temperature, pressure, or airflow in the dryer pipe exceed or fall short of a threshold that would indicate a lint blockage, the LED will be green and the alarm will not sound. If only one of these measurements surpasses one of these lower thresholds, the LED will be yellow but the alarm will not sound. In this case, it is recommended that the user access the device's web page and determine which of the measurements is indicating an issue. Remove any lint from the dryer trap and consider checking for a lint accumulation at the point where airflow is expelled from the residency. If one or more of these measurements exceeds a dire threshold, defined as high temperature, high pressure level, or lack of airflow that is threatening to the dryer's operation, the LED will be red and the alarm will sound periodically until the dryer is turned off. It is recommended that you remove the ventilation pipe from the back of your dryer and check for and remove any lint blockages and remove any lint from the dryer trap. Assistance from an HVAC technician may be required.

The SmartLint website will automatically refresh the sensor readings if the dryer is running. When the dryer is initially turned on, wait 30 seconds before accessing SmartLint readings. During this time, the device will not light up or issue an alarm on the basis of any sensor readings. Sensor readings are reported in hPa (pressure), ºF (temperature), and MPH (airflow).

The SmartLint device is only intended for monitoring and alerting the user of infractions in the pressure, temperature, and airflow in the dryer ventilation system. These readings are intended to be a reflection of the issues resulting from potential lint blockages. SmartLint does not automatically clean the dryer ventilation sleeve or the lint trap on the dryer and its readings are not indicative of the amount of lint present.

How to Troubleshoot Device:

SmartLint is designed to require zero user maintenance or intervention, and should automatically turn off and on with the dryer. However, if readings are not accessible when the dryer is running, the LED does not light, or the buzzer does not sound when a reading crosses a dire threshold. Follow these steps:

1. Unplug and replug-in the SmartLint device from the wall-wart
2. Press the reset button on the SmartLint device

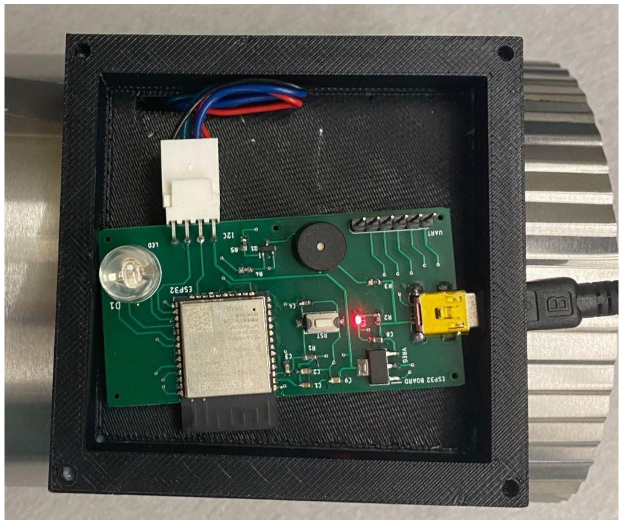


Figure 9. Top view of device on dryer pipe

3. Check to ensure that your home network is still available and accessible by the SmartLint device

If the SmartLint device continues to fail, you would need to call our support team and they can assist you.

6. To-Market Design Changes

Installation Without Pipe Built-in: Our current setup has the SmartLint system installed already on a piece of dryer sleeve. This is currently the best way to do this, as installation requires gluing within the dryer vent and would be too complicated otherwise. However, it would be best if the system were adjusted to be installed on an existing vent, through options such as the circuits protruding down and not needing to be manually glued in. This would allow for installation into areas where there isn't enough space for more tubing, and save bulky packaging of dryer tubing.

Sensor Protection: Due to the protrusion of the sensors into the pipe, it is very possible that lint could build up on and around the sensor. This would likely cause problems in the readings, especially with airflow. One solution would be some sort of cage/protector around the airflow sensor that would not cause lint to stick to it over time. This is a long-term issue with the design that should be fixed.

Notification Alert System: As of now, the user has to check the website constantly to see if the device is detecting any issues or if it is in the proper range. Outside of the physical alert system, there is no notification system for the user's phone. A much more convenient way would be to send a push notification to the user's phone, thus instantinly alerting the user of an issue. With more time this could be implemented before the final manufacturing.

Overall Testing of Thresholds: During this project the resources were not available to actually test this in a clogged dryer that results in a fire. This would be invaluable data to see how the sensor acts and what it reads briefly before an actual fire. Thus, thresholds had to be determined from manufacturer safety requirements. For a real product, it would be valuable to actually test this in real-world conditions.

Automatic WiFi Connection: If we were to distribute this commercially, we'd like to improve the user-friendliness of the process for connecting to a home WiFi network. Right now, the user needs to navigate to two separate IP addresses in order to, first, store the home network's network ID and password on the ESP32 in flash memory and, second, to navigate to the page that displays the results. We would rather the connection to the ESP's WiFi signal automatically pull up a page that prompts the user for their home network information, and from there prompt the user to visit the sensor's page without the manual input of any IP address.

7. Conclusions

This device has been months in the making and has been a true test of everything we have learned over the past four years. Multiple classes were needed to make this project: from Introduction to Engineering to Senior Design I. The original idea evolved from a possible detector just for the lint trap to a detector that was actually inserted into the dryer pipe, a much harder device to implement and one that would not be as useful at detecting fires. The system came along relatively well until integrating in the actual board rather than the bread board just used to test the software. However, finally the output pin issues were solved and the board worked seamlessly with the software.

Another regret in retrospect was the use of Atom instead of Visual Studio. Overall, Atom was not as conveniently integrated and easy to use as Visual Studio, thus it would’ve been wiser to switch over earlier in the project.

Overall, there was adversity throughout the project that had to be overcome. However, through cooperation and perseverance, the entire team was able to create something that solved our initial problem, dryer fires, in a unique way. The SmartLint team is very proud of the final product.

8. Appendices

8.1 Complete Hardware Schematics

<u>Main board schematics</u>:

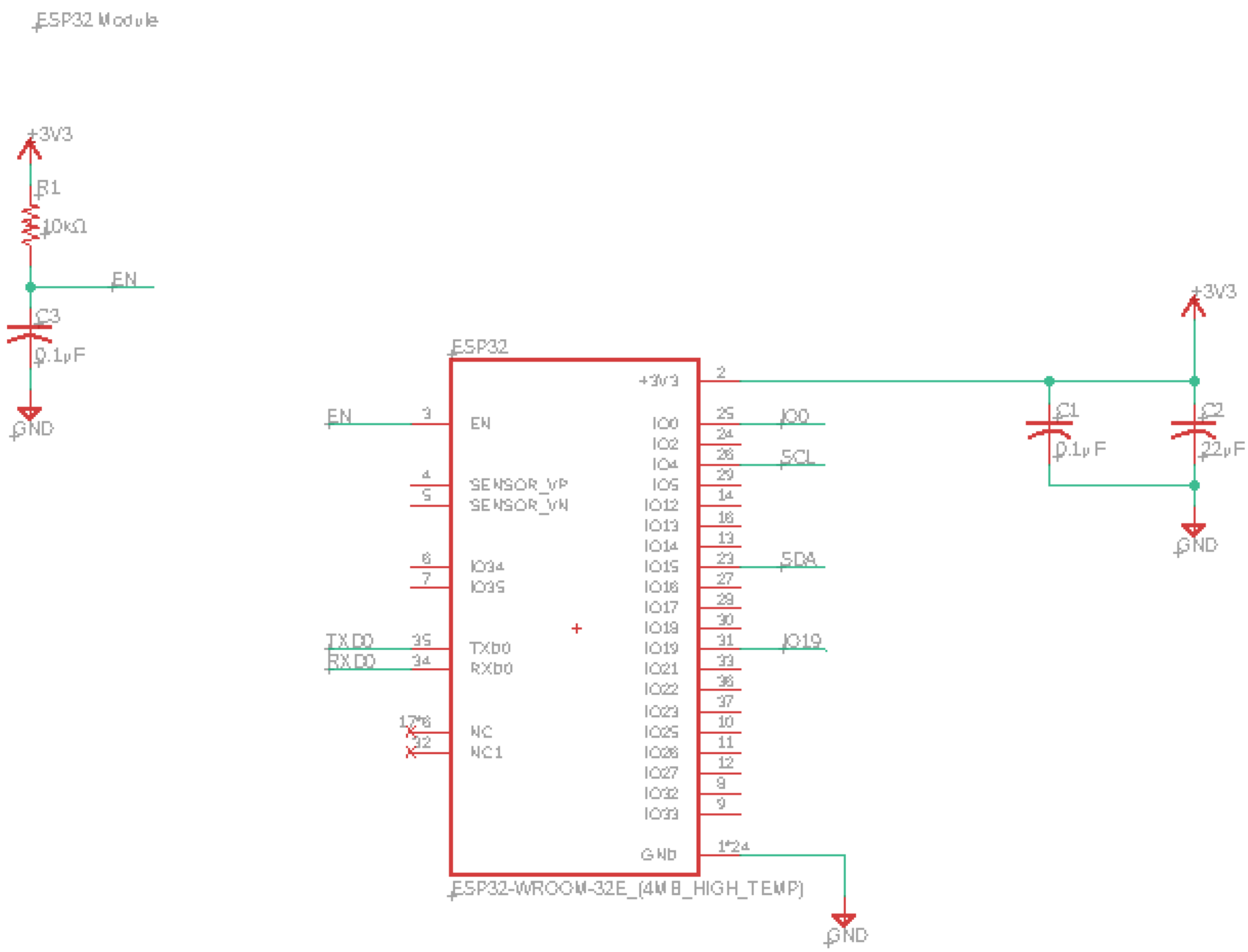


Figure 10. ESP-32 module

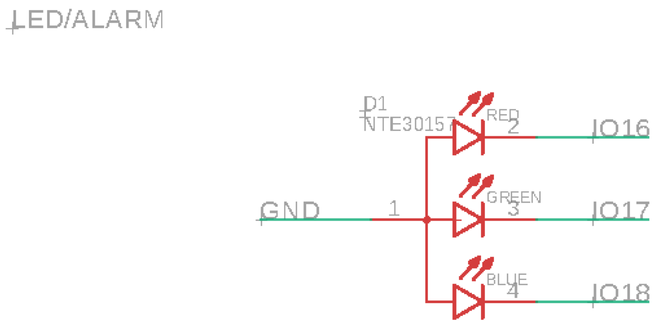


Figure 11. LED circuitry

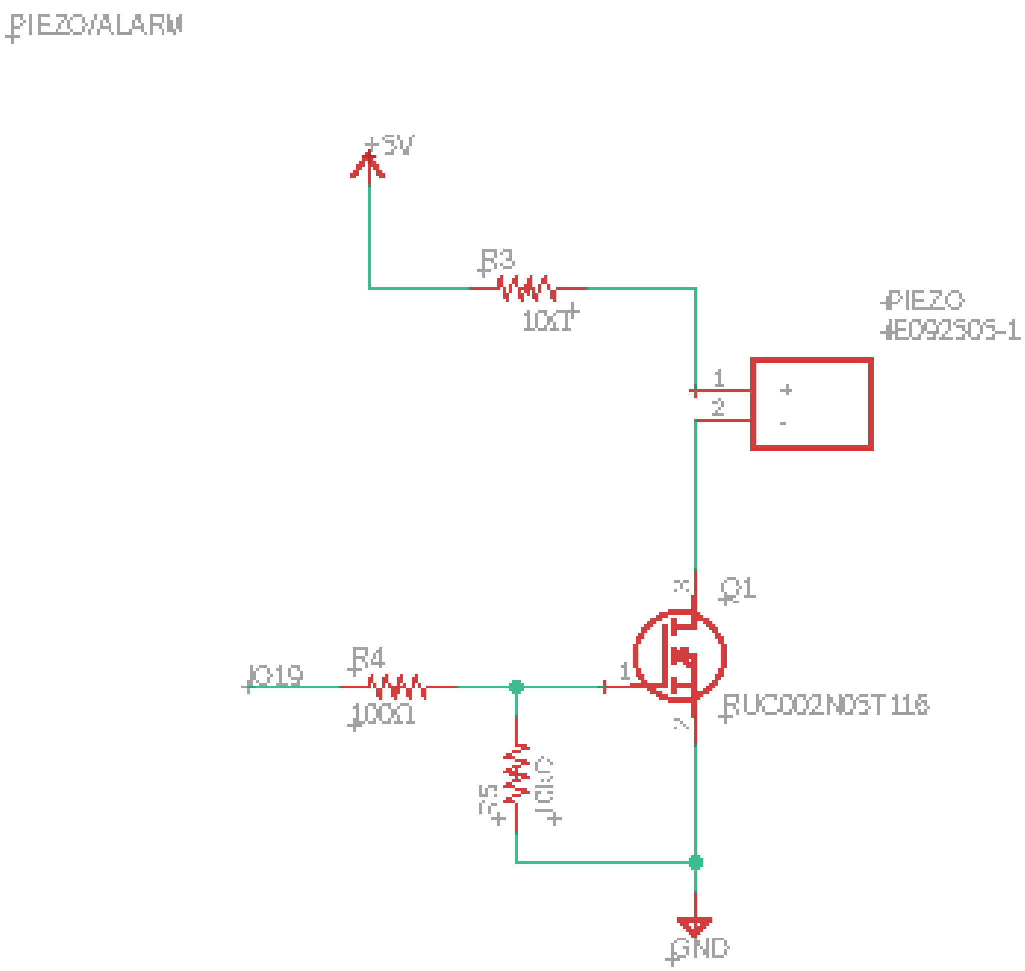


Figure 12. Piezo circuitry

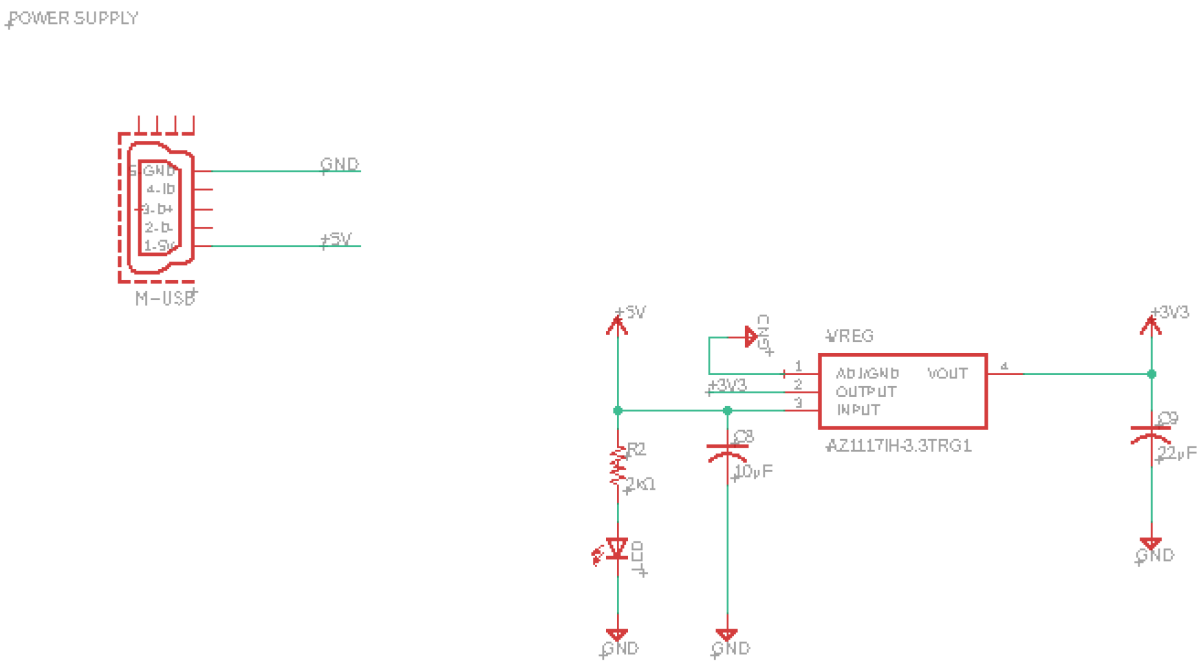


Figure 13. Power supply circuitry

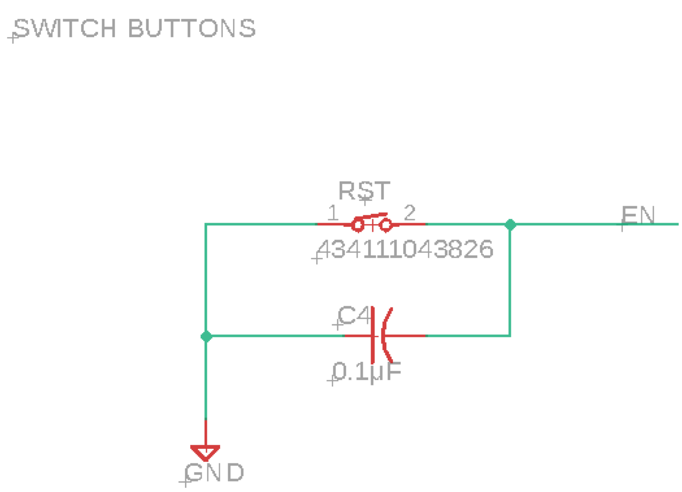


Figure 14. Reset circuitry

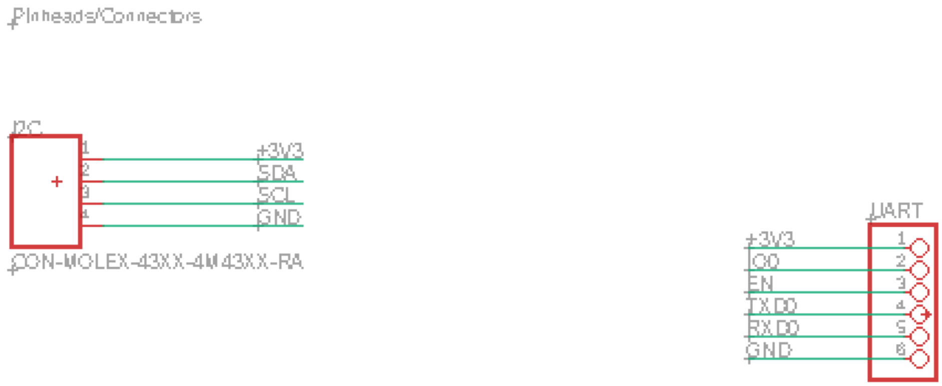


Figure 15. Main Board Pinheads and connectors

<u>Sensors board schematic</u>:

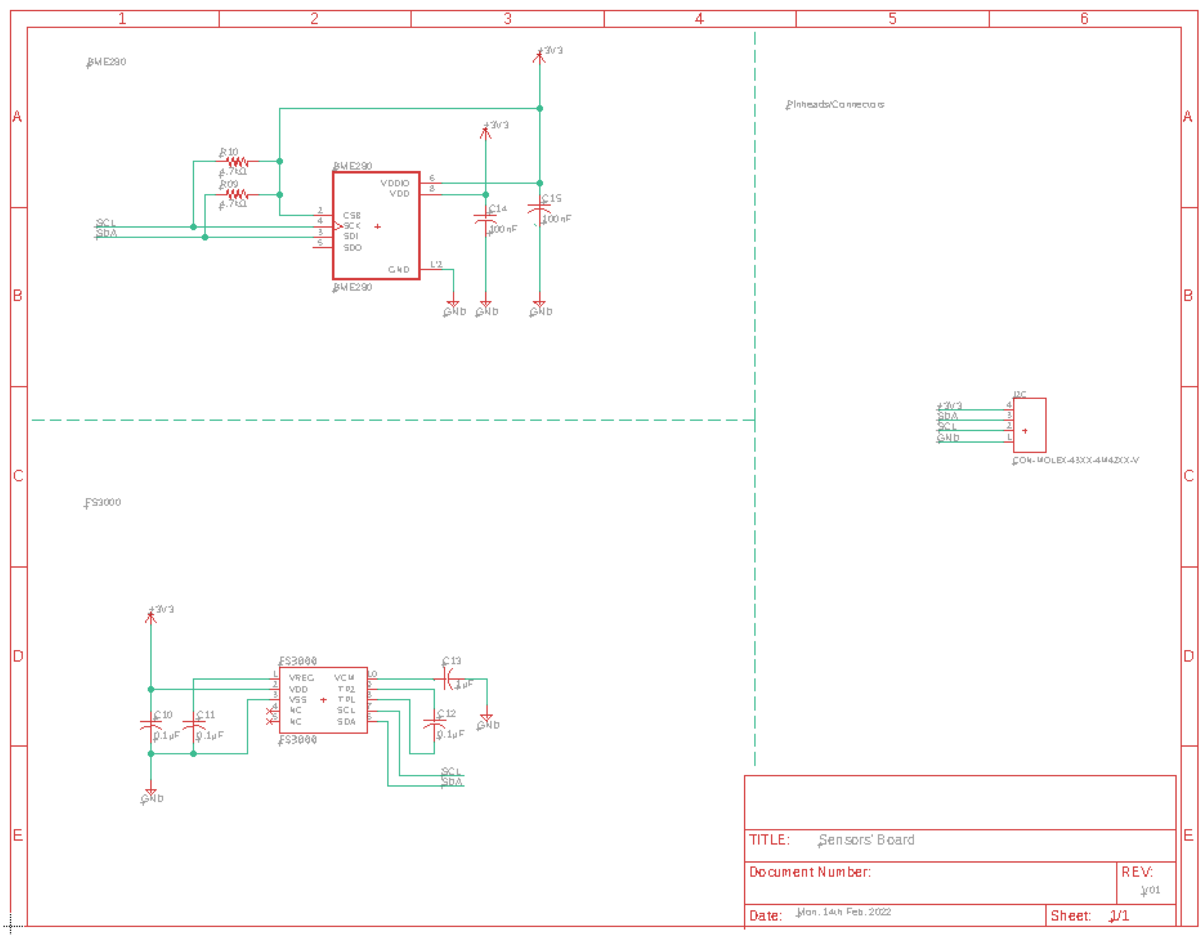


Figure 16. Sensors' board circuitry

Board Manufacturing Top & Bottom Views:

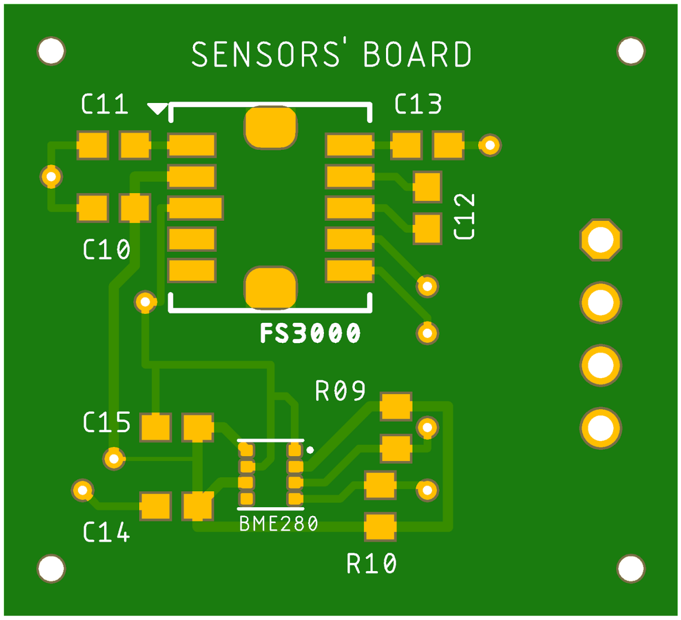


Figure 17(a). Sensors' board top view

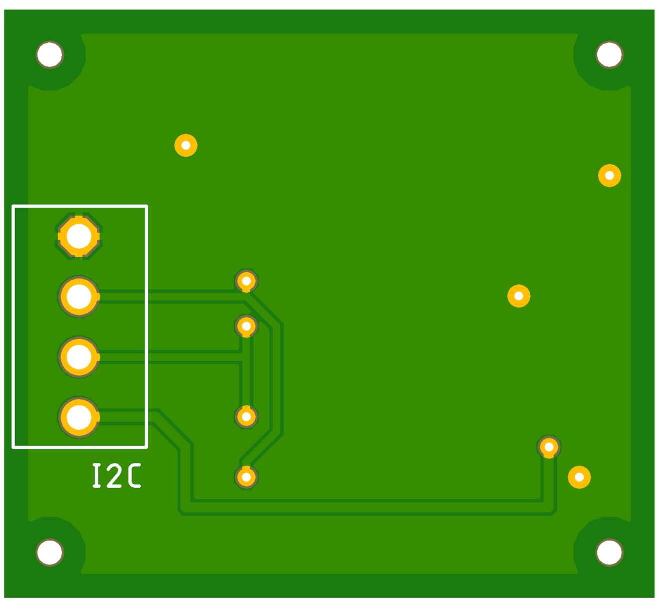


Figure 17(b). Sensors' board bottom view

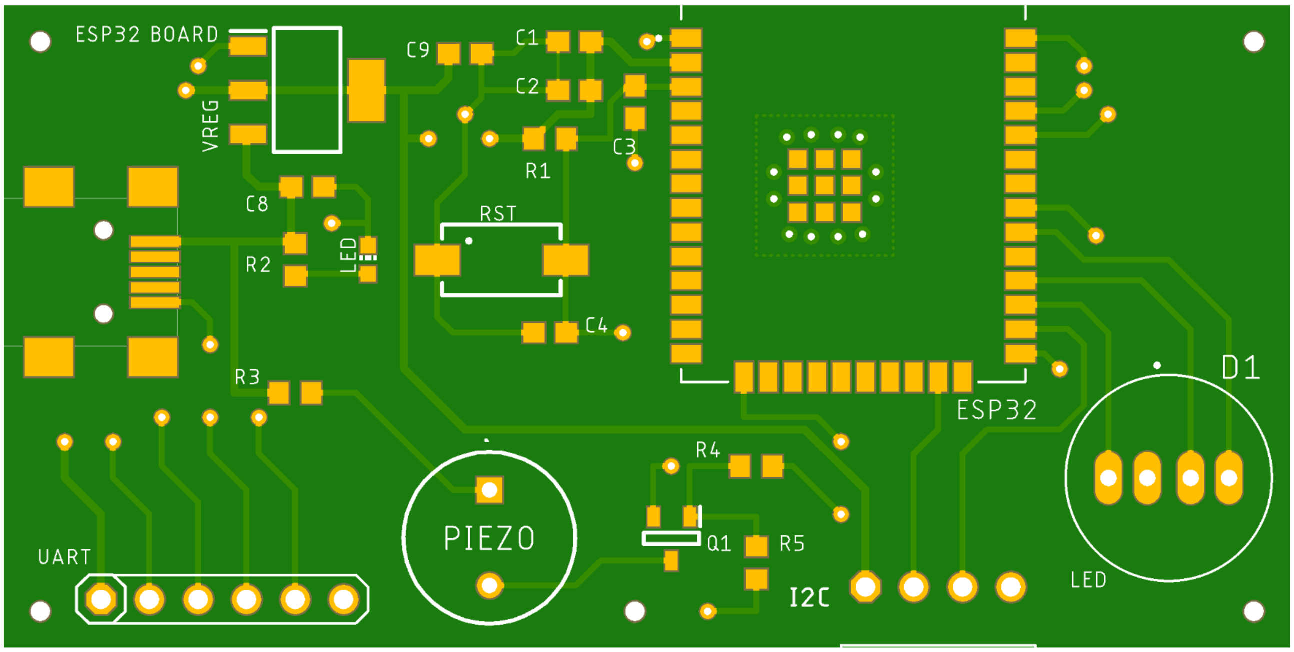


Figure 18(a). Main board top view

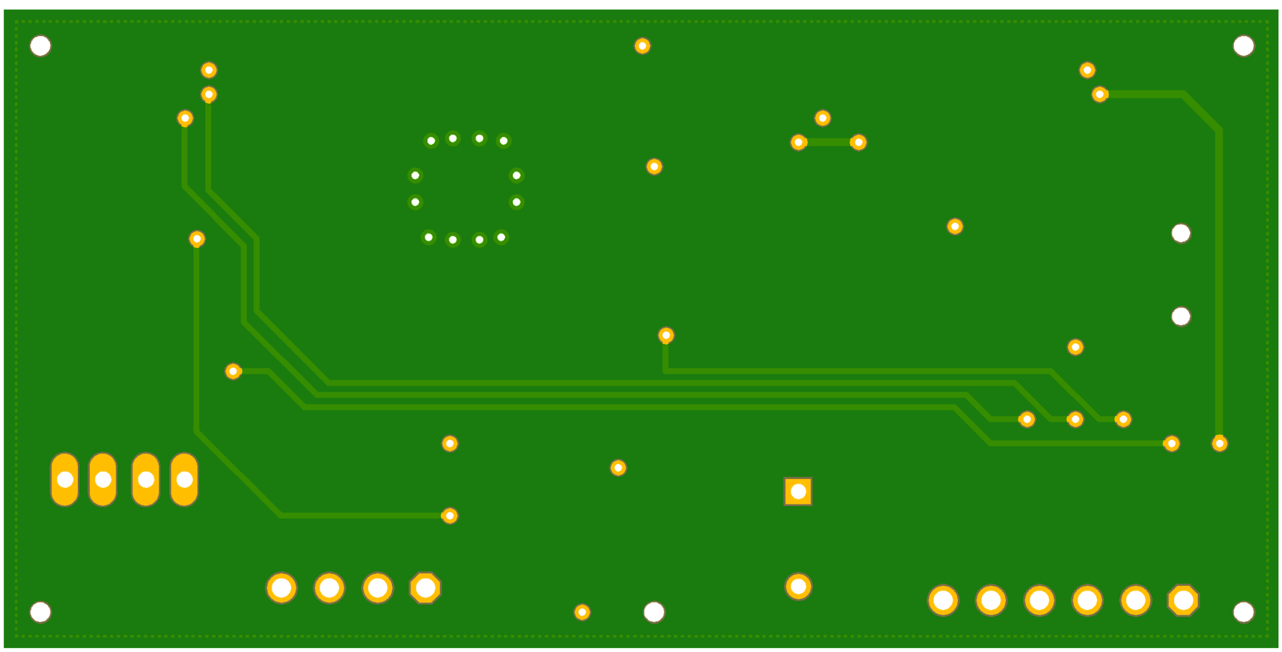


Figure 18(b). Main board bottom view

8.2 Complete Software Listings

The complete code for this project is accessible on GitHub: SmartLint Code

8.3 Relevant Parts

Air velocity sensor: FS3000

Temperature, pressure, and humidity sensor: BME280

WiFi-enabled microcontroller: ESP32-WROOM-32E-H4

Voltage regulator: AZ1117IH-3.3TRG1

Piezoelectric buzzer: IE092505-1

Multicolor LED: APT1608LSECK/J3-PRV

N-channel MOSFET: RUC002N05T116

Micro-USB connector: 10103594-0001LF

Push button: 434111043826