Household Device Ecosystem

ECE 445 Design Document
Team 33 - Ian Goodwin, Sam Atac, John Armgardt
TA: Ali Kourani
Spring 2021

1 Introduction

1.1 Objective

In the near future, as 5G networks roll out, IoT devices are predicted to become an ever more important facet of modern life. It is predicted that by 2025, there will be 75 billion IoT devices in use across the globe, an increase from just 7 billion in 2018[1]. As this rapid transition unfolds, consumers will begin to expect to monitor and control more of their household devices remotely. While there is no doubt that many top end future household devices will come with IoT functionality baked in, the vast majority of people will not own one of these devices for many years to come. Microwaves can last up to 10 years, Washing machines up to 16 years, and ovens up to 23 years[2]. The lifespan of these often expensive items will last far into the IoT revolution, slowing adoption of the technology as a whole. One other issue faced by the IoT revolution is device compatibility. As IoT devices have exploded in popularity, the number of options for smart device ecosystems have increased drastically. Most of these ecosystems are “walled gardens”, only allowing devices from their own brands to be added to their network[3]. This limits potential options for the consumer, and only allows connection to the device types offered by that particular brand. The cumulative effect of these issues will slow IoT technology from reaching its full adoption and value. Having greater home connectivity can lead to improvements in energy savings, environmental impact, security, and comfort[4]. These are benefits that we cannot afford to ignore.

Our solution to these issues is to develop a series of WiFi-connected devices containing sensors focused on monitoring a wide range of current appliances. These devices would allow for monitoring/control of existing household devices without the need to buy dedicated smart devices or replace existing household appliances. Our devices would also be open source and as modular as possible, in order to allow the user to accommodate addition of IoT devices not designed by us. This open source philosophy would help to make the best use of any customers current IoT device setup.

Goals:

- Create three different nodes for household appliances that allow the user to monitor them wirelessly via internet
- Develop intuitive web app that allows for easy integration of newer and additional sensors
- Develop sensors such that they continue working if WiFi fails
- Create sensors at cheap, cost effective price so users can forgo buying entirely new smart appliances, ideally less than $50

Features:

- Sensors that monitor the status of washing/drying machine, microwave, and room temperature/humidity
- Button to power on/off sensor
- LED to indicate working status
- Speaker to alert user of status change when internet fails
- Web app for convenient, clean layout of appliance states
- Small and lightweight
- Easy setup and minimal upkeep

1.2 Background

The current IoT device landscape is very fractured, with certain brands tending to focus on specific sectors. The security sector is led by brands like Ring and Nest [5]. The smart electronic sector is the most mature, with major tech brands such as Google and Apple leading the way [6]. The appliance sector is currently very split, with offerings from some major brands like Samsung [7]. These devices, however, are generally stand alone, and no major ecosystem exists to connect them. Finally, there exist standalone sensors from companies like SmartDry that connect to an application. These devices, however, require their own app per sensor and are rather expensive still.

To provide a better solution than current offerings, our ecosystem needs to focus on compatibility as much as possible. Our sensors should work on a wide range of appliances, from all major brands. From a software perspective, keeping the code open source and utilizing common IEEE standards will increase device compatibility [8]. Open source will also allow savvy users to contribute their own code to further enable compatibility.

1.3 High-Level Requirements

- Sensors Modules must be able to connect to a web server via Wi-Fi module, allowing data to be transmitted to/from the sensor at high speed (less than 1 second latency).
- Sensor Modules must be able to alert the user in a different way within a range of 50 ft should the internet connection fail.
- Sensor Modules must correctly function on a majority (90+%) of the hardware in its class. (Ex: accelerometer works on 90+% of consumer washing machines)

2 Design

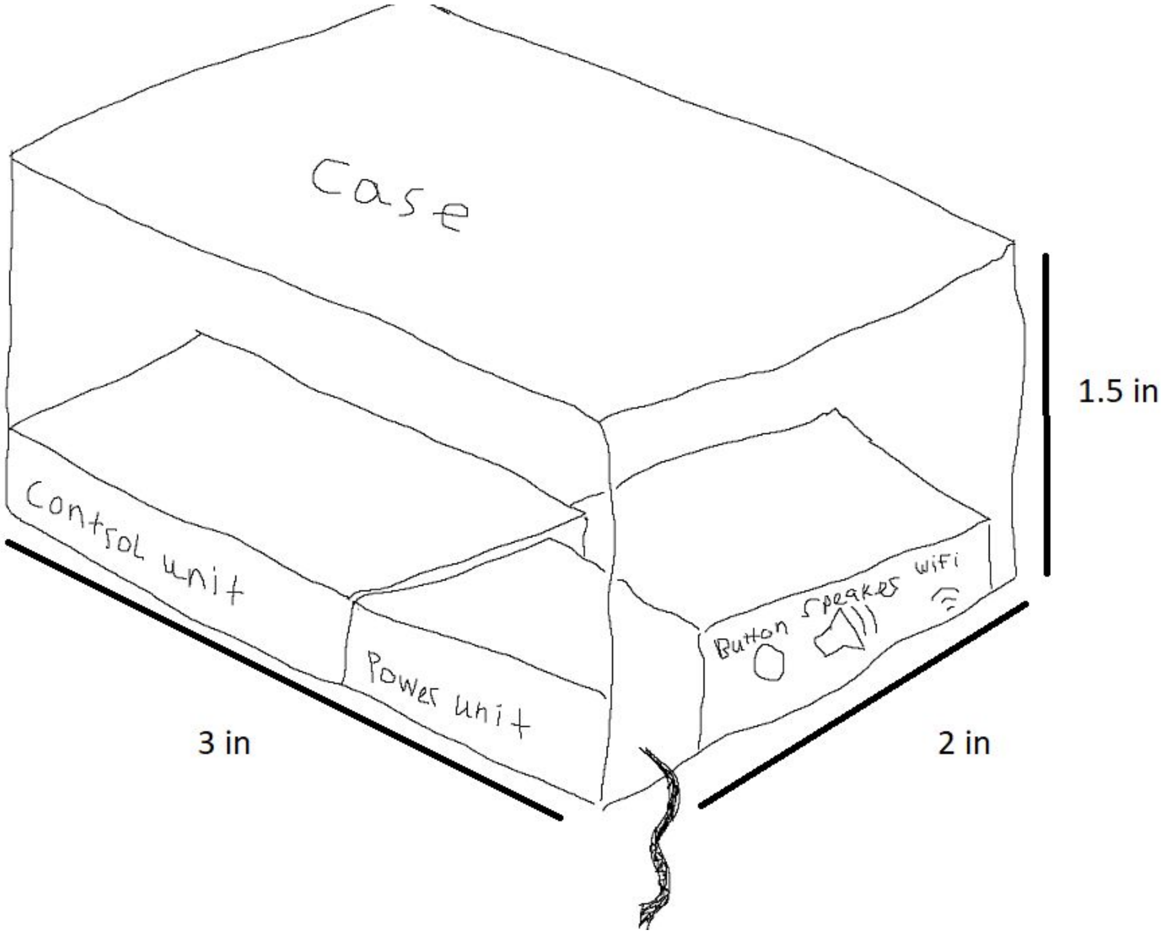

Figure 1: Sketch of general sensor

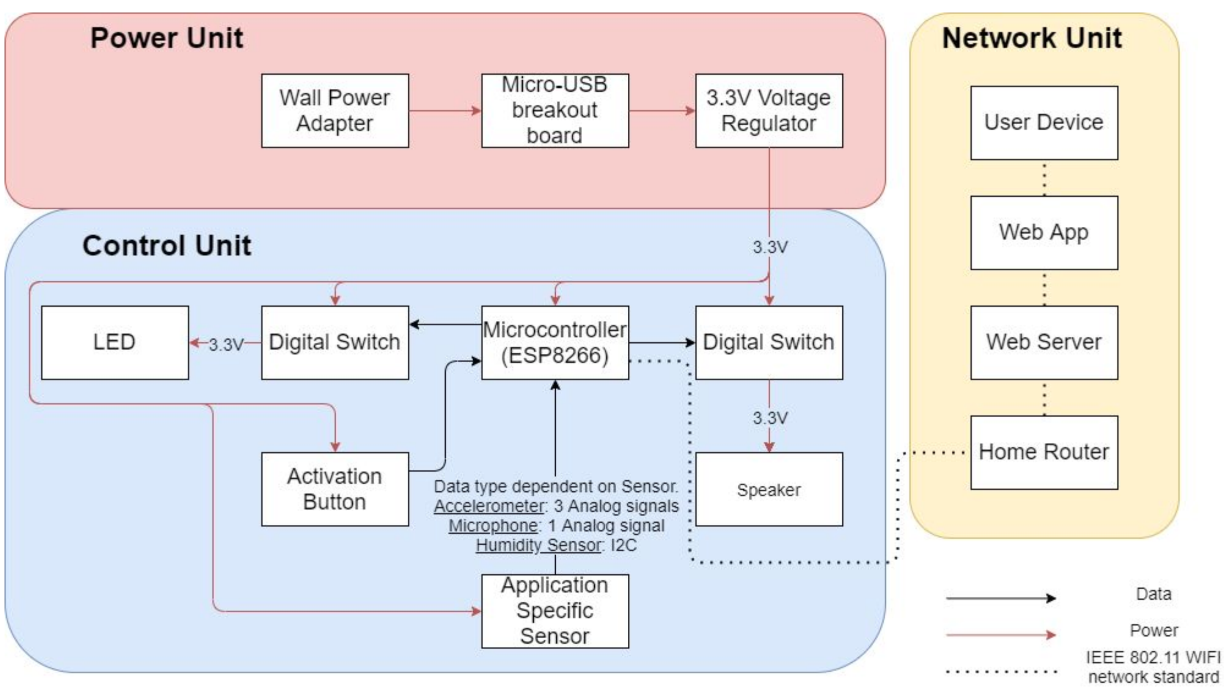

Figure 2: Block diagram

Our design consists of a series of Nodes. Each Node has three components: the Control Unit, the Power Unit, and the Network (Which is shared between all the modules). These devices all interface with each other and the user through their shared Network. The Control Unit contains the Microcontroller, Activation Button, Status LED, Speaker, Digital Switches, and the Appliance Specific Sensor. The Appliance Specific sensor varies depending on the appliance that the Node is measuring for the washer/dryer it is an accelerometer, for the microwave it is a microphone and for the humidifier it is a humidity sensor. The Control Unit Communicates with the Network Unit via the ESP32 Microcontroller over the IEEE 802.11 WIFI network standard. However should the network connection, fail the speaker will sound off in place of the web app notification. The Power Unit contains a Power Adapter, a Micro-USB breakout and a Voltage Regulator. It provides 3.3V and GND to the Control Unit. Both the Power Unit and the Control Unit are housed in the same case.

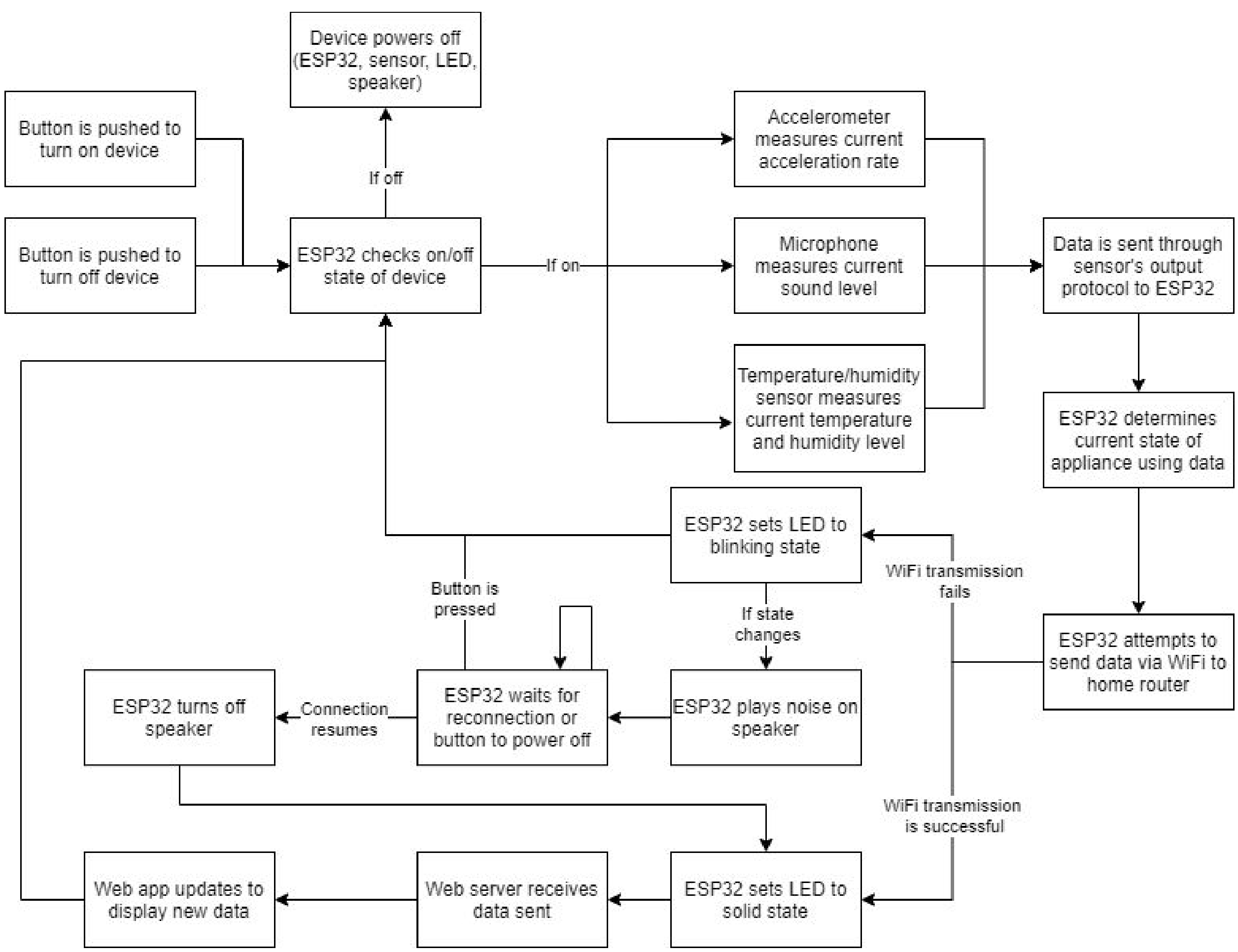

Figure 3: Flowchart of device algorithm

2.1 Power Unit

2.1.1 Wall Power Adapter

The Power adapter plugs into your standard US home 120V wall outlet and provides 5V out of a micro-USB cable.

Requirement	Verification
1. Provides 5V from your standard US home 120V circuit	1. a. Connect Power Adapter to wall socket. b. Use a voltmeter to ensure that the output voltage is 5V.

2.1.2 Power Breakout

The Power Breakout will be a Micro-USB breakout that connects to the Wall Power Adapter and outputs 5V which will then be regulated down to 3.3V by the Voltage Regulator.

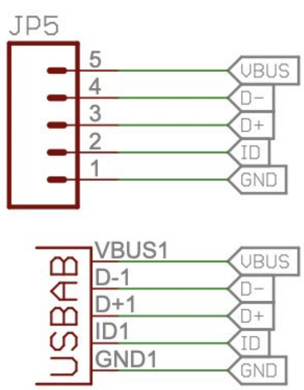

Figure 4: Power breakout board schematic[9]

Requirement	Verification
1. Outputs >500 mA at 5V	1. a. Plug Wall Power Adapter

	into the wall socket and the Power Breakout Board b. Use a voltmeter to measure the Voltage and the Current. c. Ensure that the voltage is 5V and the current is > 500 mA.

2.1.3 Voltage Regulator

The Voltage Regulator, a LD1117V33 chip, will regulate the 5 Volts output by the Power Breakout down to 3.3V for use by the Microcontroller, LED, Activation Button, Digital Switches, Speaker and the Appliance Specific Sensors.

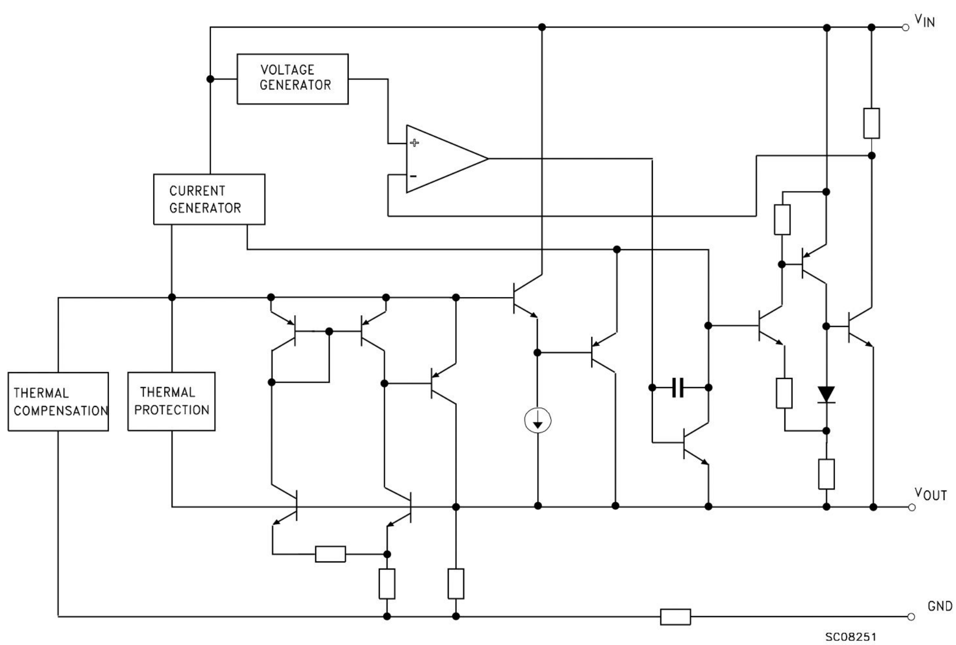

Figure 5: Voltage regulator circuit[10]

Requirement	**Verification**
1. Outputs > 200 mA at 3.3V	1. a. Plug the Wall Power Adapter into the Wall and the Micro-USB breakout board. b. Connect the 5V and GND lines to the Voltage

	Regulator. c. Use a voltmeter to measure the voltage and current. d. Ensure that the voltage is 3.3V and the current > 200 mA.

2.2 Control Unit

The Control Unit accepts data from the sensors, packages that data and sends it out to the home router over wifi. The Control Unit communicates with the Network via the IEEE 802.11 WIFI network.

2.2.1 Microcontroller

The Microcontroller, an ESP32, is responsible for communicating with the local devices. The Microcontroller will be accepting data from the activation button (via an analog signal) and the Appliance Specific Sensor, via an array of analog signals in the case of the Accelerometer, a singular analog signal from the Microphone and via I2C in the case of the Humidity Sensor. The Microcontroller will be transmitting data to the Web Server via the IEEE 802.11 WIFI network standard. The Microcontroller is also responsible for providing a signal to each of the Digital Switches in order to trigger either the Status LED or the Speaker.

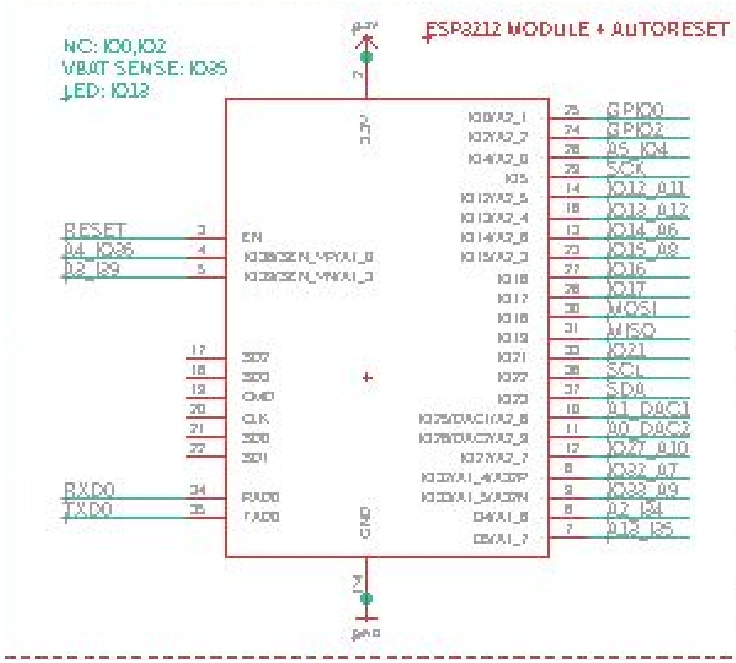

Figure 6: Microcontroller schematic[11]

Requirement	**Verification**
1. Able to communicate over the IEEE 802.11 WIFI network standard with the Web Server with sub 1 second latency. 2. Receive data over I2C from Humidity Sensor with a latency of < 50 ms	1. a. Setup Web Server to respond to requests with sample data b. Connect Microcontroller to PC c. Program Microcontroller to send a Web Request to the Web Server and start a timer d. Read out response on Microcontroller and stop timer e. Ensure that timer is less than 2 seconds(Due to the round-trip). 2. a. Connect Microcontroller to USB UART bridge b. Program Microcontroller to respond to UART message by replying with the same

	message c. Connect USB UART bridge to computer d. Use python and a serial library, such as pyserial, connect to the USB port e. Start a timer f. Send a message g. Wait to receive that same message h. Stop the timer i. Check the response to ensure that it is the same j. Check the time it took

2.2.2 Activation Button

The Activation Button is used to indicate to the Microcontroller that the Appliance has begun a use cycle. The Activation button is a very simple tactile push button that is monitored by one of the Microcontroller's GPIO pins and receives 3.3 volts from the Voltage Regulator. The Activation button is active high.

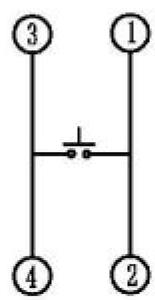

Figure 7: Activation button circuit[12]

Requirement	Verification
1. Able to be easily pressed, with no more than 1 Newton of Force necessary.	A. We will put an moderately sized apple on the button and see if it depresses

2.2.3 Status LED

The status LED indicates to the user whether or not the device is working as intended: the device is powered, information is being exchanged with the webserver, sensors are outputting reasonable data. It will be solid green if everything is working as expected and blinking if the wifi connection has failed or off if the board is unpowered. The Status LED is a two pin green common cathode and will be powered by a Digital Switch which is triggered by a GPIO pin on the Microcontroller.

Requirement	**Verification**
1. Able to clearly differentiate solid green, blinking green and unpowered.	1. a. Connect LED to a breadboard b. Connect the voltage-in pin to 3.3V and the GND pin to GND c. Ensure that the color is Green d. Connect the voltage-in pin to a microcontroller pin that modulates the voltage high and low every second e. Ensure that the LED is Blinking f. Connect all of the pins to ground g. Ensure that the LED is colorless

2.2.4 Speaker

The Speaker serves as a backup way to alert the user should either the home network or the web server go down. If that is the case and the home appliance has finished (i.e: microwave or washing machine) the Speaker will chirp. The Speaker will receive power through one of the Digital Switches which will output 3.3V and will be triggered by one of the GPIO pins. The Speaker will stop making noise if either the Microcontroller is able to communicate with the Web Server again or the user hits the Activation Button.

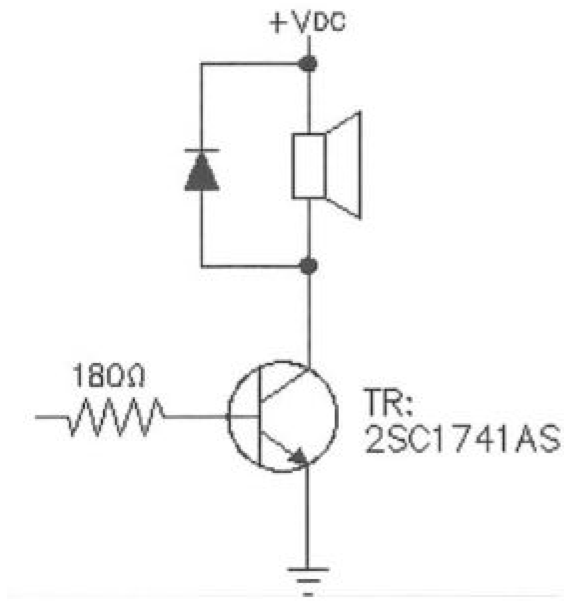

Figure 8: Speaker circuit[13]

Requirement	Verification
1. Able to hear from another room, we estimate at >= 90 dB. 2. Able to receive internet failure signals and inform the user.	1. a. Activate Speaker in kitchen b. Walk into bedroom ~40 ft c. Check if the speaker is still audible 2. a. Connect Microcontroller to Wifi Module b. Connect Wifi Module to Wifi c. Monitor Wifi Connection d. Turn off house Wifi e. If Microcontroller receives bad network message, trigger Speaker

2.2.5 Digital Switch

We will be using two pnp bipolar junction transistors as digital switches to power the Speaker and the Status LED. The collector will be supplied by the 3.3V Voltage Regulator. The emitter will be connected to the voltage-in pin on the Speaker and the Status LED. The base will be connected to one of the Microcontrollers GPIO pins.

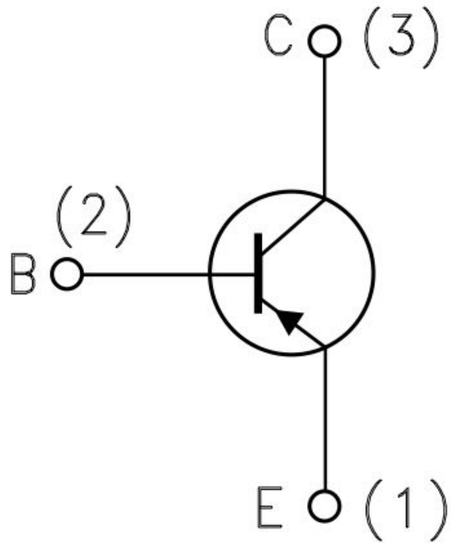

Figure 9: Digital switch circuit[14]

Requirement	Verification
1. Able to allow voltage to flow from the collector to the emitter when a 12mA, 3.3V signal is applied to the base.	1. a. Hook collector up to a 3.3V source b. Connect GND to base c. Use voltmeter to ensure that the emitter is outputting low voltage d. Connect a 3.3V and 12mA signal to the base. e. Use voltmeter to ensure that the emitter is outputting high voltage

2.2.6 Appliance-Specific Sensors

In order to detect the state of a certain appliance, we will use specific sensors to measure data related to that appliance. The sensors will be powered using the voltage regulator, and the data will be outputted into the Microcontroller.

2.2.6.1 Washing/Drying Machine: Accelerometer

In order to detect when our Washing Machine or Dryer has finished we will be using a simple 3-axis accelerometer, specifically the ADXL335. The idea being that the machine goes through cycles with varying levels of vibration and that when the cycle is over the vibration should stop almost completely. The accelerometer will have three analog wires connecting to the Microcontroller, each with a voltage corresponding to either the X, Y and Z acceleration. The Voltage will be a range between -325 mV-325mV at a frequency of 100hz. We will require that our accelerometer has a measurement resolution of .01 m/s^2, and can record up to 1g at a minimum. These values are derived from the tolerance analysis.

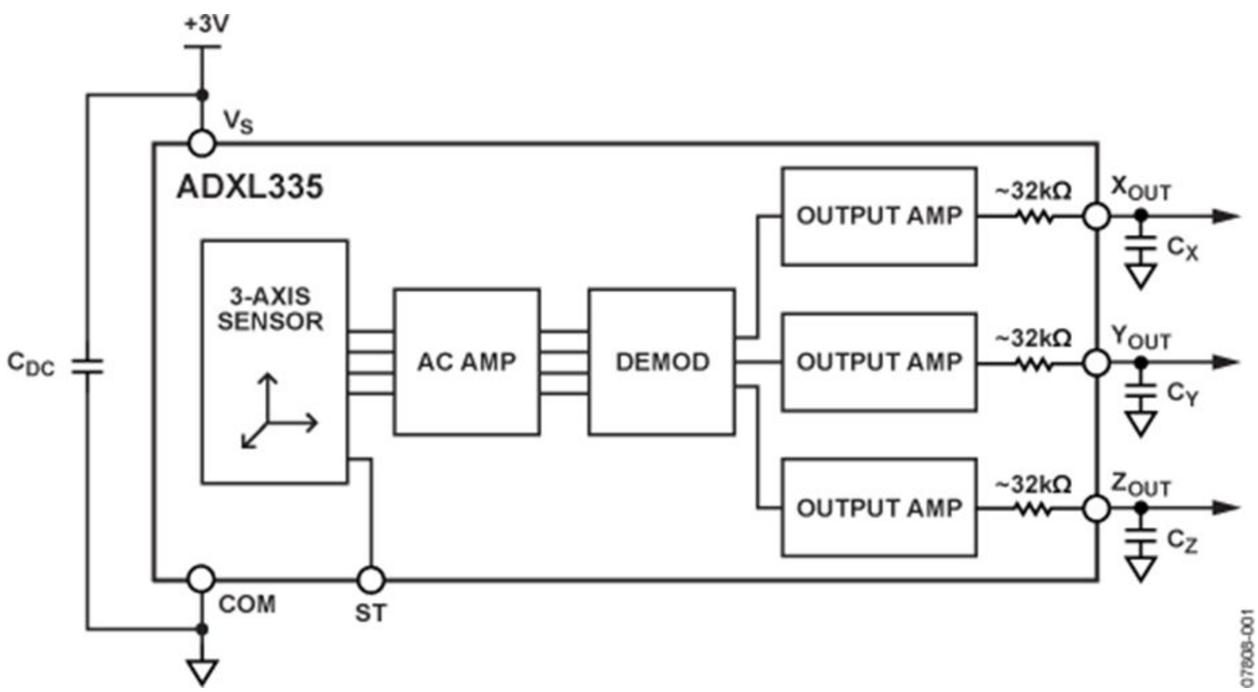

Figure 10: Accelerometer schematic[15]

Requirement	**Verification**
1. Accelerometer must have a measurement resolution of .01 m/s^2 at a maximum 2. Accelerometer should have a total measurable range of at-least -1 g to 1 g	1. a. Connect accelerometer to breadboard with power b. Drop breadboard setup and measure data on accelerometer c. Ensure data is within .01 m/s^2 with Earth's gravitational acceleration 2. a. Connect accelerometer to breadboard with power b. Drop breadboard setup and measure data on

	accelerometer c. Ensure data reaches a value greater than or equal to 1 g

2.2.6.2 Microwave: Microphone

In order to detect when our Microwave is finished we will be using a Microphone. Using the Microphone we will detect the hum of the Microwave and the beeps that it makes upon finishing. The Microphone that we will be using is the ADMP401 which is an Omnidirectional Microphone with Analog Output. ADMP401 will output a single analog signal which will vary between .5V and 3.3V at 15kHz. We will require that our microphone has a measurement resolution of 5 db, and can record up to 60 db at a minimum. These values are derived from the tolerance analysis.

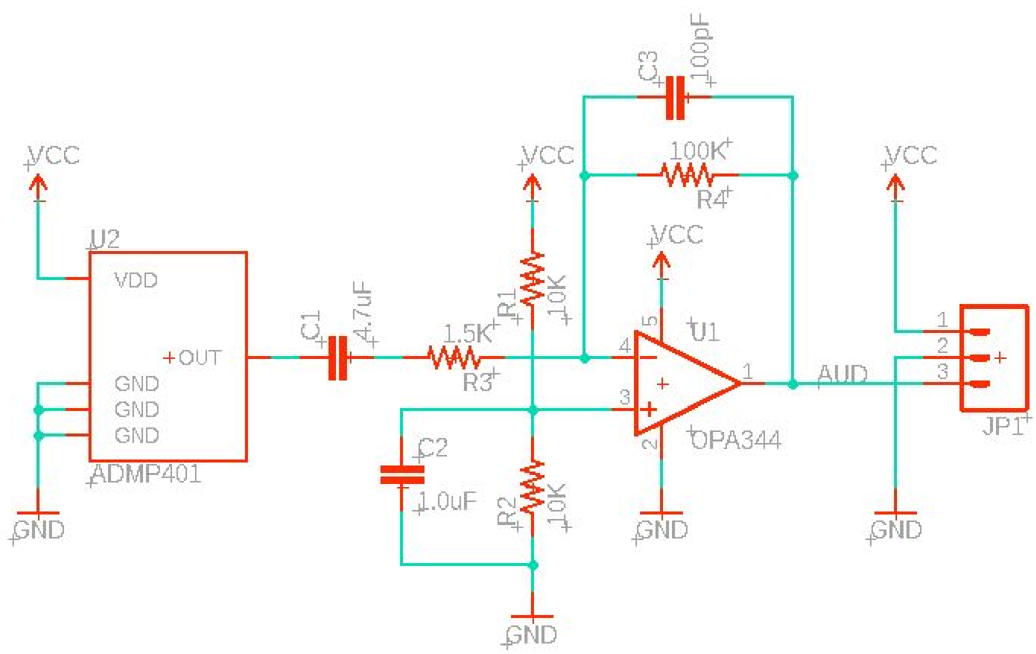

Figure 11: Microphone schematic[16]

Requirement	Verification
1. Microphone can measure the difference between 5 dB sounds. 2. Microphone can record up to 60 dB at a minimum.	1. Play two sounds at 5 dB apart and compare recorded voltages 2. Play sound getting increasingly louder until 60 dB and compare

	data recorded

2.2.6.3 Room: Humidity Sensor

The Humidity Sensor will be used to detect when a room comes to the desired humidity, it is intended to be used with either a humidifier or dehumidifier. We will be using the Qwiic Humidity AHT20. The Humidity Sensor will communicate with the Microcontroller via an I2C signal. We will require that our humidity sensor has a working range of 0-80% minimum, a response time of 15 seconds maximum, and an accuracy error of ±5%. These values are derived from the tolerance analysis.

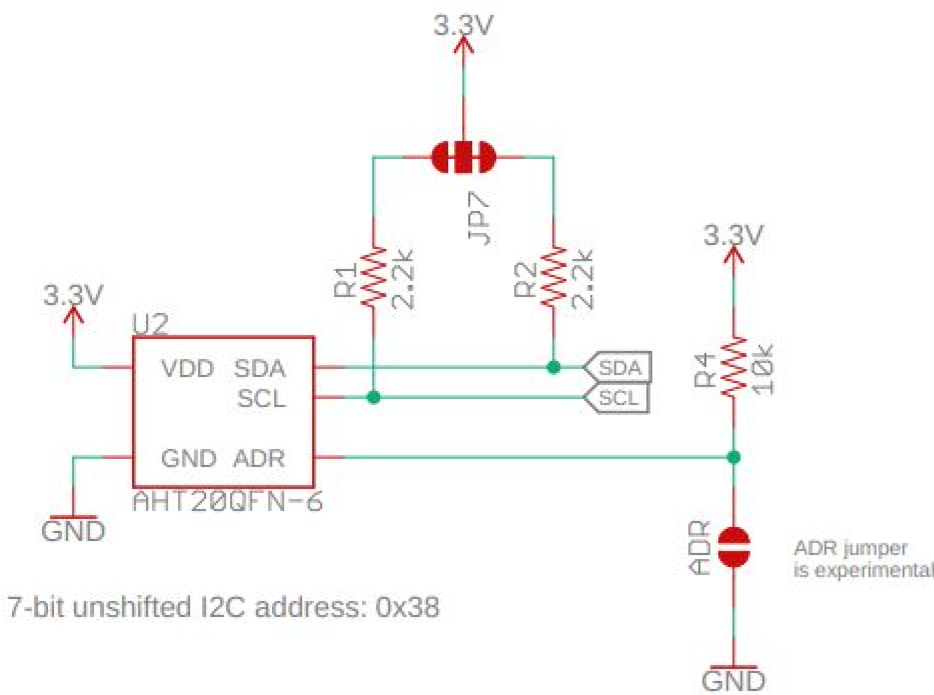

Figure 12: Humidity sensor schematic[17]

Requirement	**Verification**
1. Temperature sensor a. Working range of 0-40°C minimum b. Response time of 15 seconds maximum	1. Temperature sensor a. Place sensor in 0°C and 40°C environments and verify correct voltages b. Place sensor in 20°C environment, wait for sensor

c. Accuracy error of ±2°C. 2. Humidity sensor a. Working range of 0-80% minimum b. Response time of 15 seconds maximum c. Accuracy error of ±5%.	to output 20°C, change to 25°C environment and record time elapsed c. Place sensor in 20°C environment and measure error in recorded temperature 2. Humidity sensor a. Place sensor in 0% and 80% humidity environments and verify correct voltages b. Place sensor in 50% environment, wait for sensor to output 50%, change to 60% environment and record time elapsed c. Place sensor in 50% environment and measure error in recorded temperature

2.3 Network Unit

The network unit encompasses the hardware we will need to collect the sensor data, process it, and send it to the user. The core of the network unit is the remote server, which we will have one of. This server will receive data from all of the users and their respective nodes, and serve the web app.

2.3.1 Home Router

The home router will be utilized in our design to pass data from the nodes to our web server. It will be assumed that the user already possesses one, as these are common household devices.

2.3.2 Web Server

The web server design consists of an EC2 server from AWS. This server will run an instance of Node.js to handle REST requests and internal server operations. We will also utilize an instance of MongoDB that the Node instance can interface with to store user and sensor data.

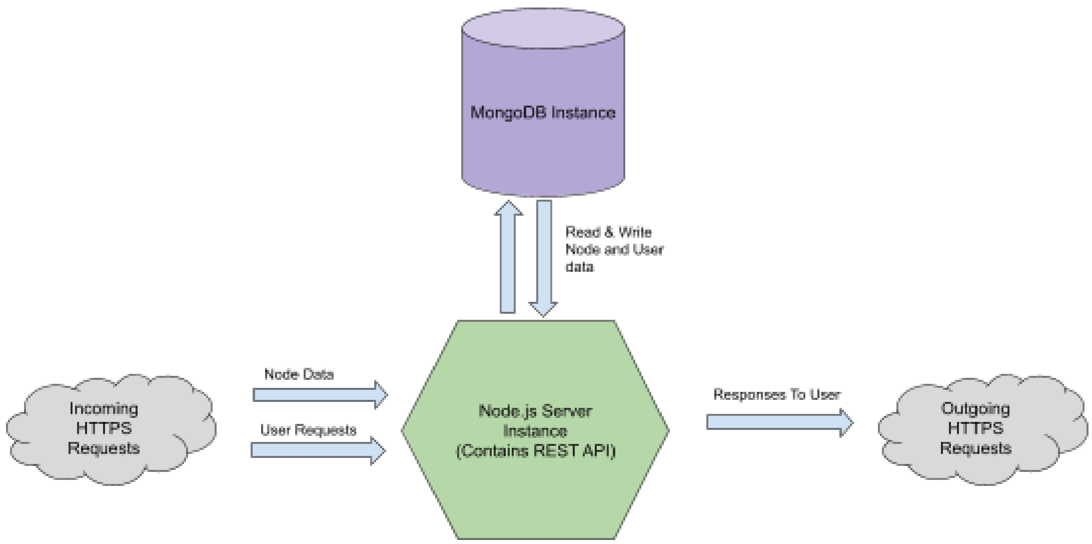

Figure 13: Web server flowchart

Requirement	**Verification**
1. Server receives HTTPS Request from Node with less than 200ms latency 2. Server Receives HTTPS Request from User with less than 200ms latency 3. Server Reads from and Writes data to MongoDB Instance 4. Server performs under load, can handle up to 1000 API Requests a minute	1. Have Node device send out HTTP request, with timestamp. Log the request on server with timestamp on receival. Verify the difference is under 200ms 2. Have User device send out HTTP request, with timestamp. Log the request on server with timestamp on receival. Verify the difference is under 200ms 3. Set up test GET and POST API calls. POST a piece of data remotely, then retrieve it via the GET call. 4. Run python script to request resources from the server 1000 times within the minute. Verify that all of the requests were handled via terminal output

2.3.3 Web App

The Web App will provide the primary user interaction for our project. This app will be dynamically scalable for both mobile and desktop view, and lightweight to support machines with a wide range of compute power. It will allow users to log on, and receive push notifications as they desire from their nodes.

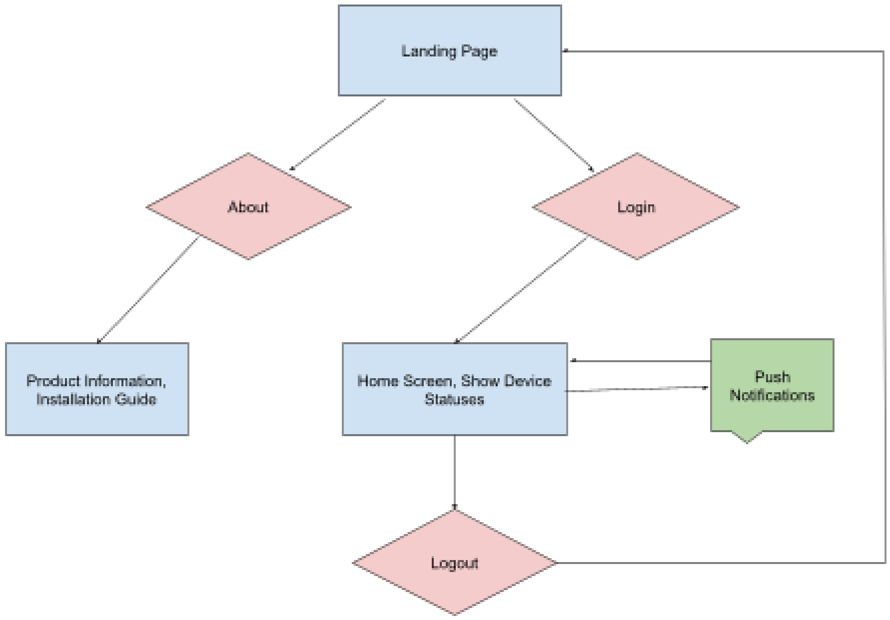

Figure 14: Web app flowchart

Requirement	**Verification**
1. User is able to log in and out of the application 2. User can receive push notifications 3. User can access the product information and installation guide 4. User is able to see data from their nodes	1. Access the website, and enter account credentials. Verify that the user is taken to the home screen, and the correct node data is displayed. 2. On the home screen, select the option for notifications. Trigger one of the nodes manually, and verify that the user received a push notification. 3. Access the website, and select the about screen. Verify that the user can see the product information and installation guide 4. From the home screen, verify that the user is able to see all of the active nodes in their account. Check this with the nodes stored in the server for accuracy.

2.4 Case

The case, shown in Fig 2., will protect the electronic components from outside interaction and tampering. It will also keep the components stable, so that movement the device undergoes will not disturb the internal wiring.

Requirement	**Verification**
1. Internal components are not moved when case is moved 2. Case is resistant to dust and water up to IP54	1. a. Open case up, note placement of components, and close case b. Shake case thoroughly c. Ensure components have not moved after shaking and device still works

2.5 Power Calculations

Supply

3.3V Voltage Regulator: 2A at 3.3V

Consumption

ESP32 : 250 mA
Speaker : 35 mA
LED : 12 mA
Button : Negligible
Digital Switches : Negligible

Appliance Specific Sensor

- Microphone : 200 μA
- Accelerometer : 350 μA
- Humidity Sensor: 23 μA

Appliance Specific Sensor Max : 350 μA

Total Current Requirement : 297.35 mA

2.6 Tolerance Analysis

The most important tolerance we must maintain is the sensitivity and functionality of the appliance-specific sensors. To ensure our devices are to work properly, we must ensure the sensors themselves are sensitive enough to measure the difference between the states. Furthermore, we must ensure that they are provided the right amount of power to function properly.

First, we will determine the tolerance required for the washing/drying machine's accelerometer. To measure this, we place a OnePlus 7 Pro on the top end of the machine and measure the z-direction acceleration using the internal accelerometer. We then normalize this data by subtracting 1 from each point, and taking the absolute value, since there is 1g acting on the accelerometer by default due to gravity.

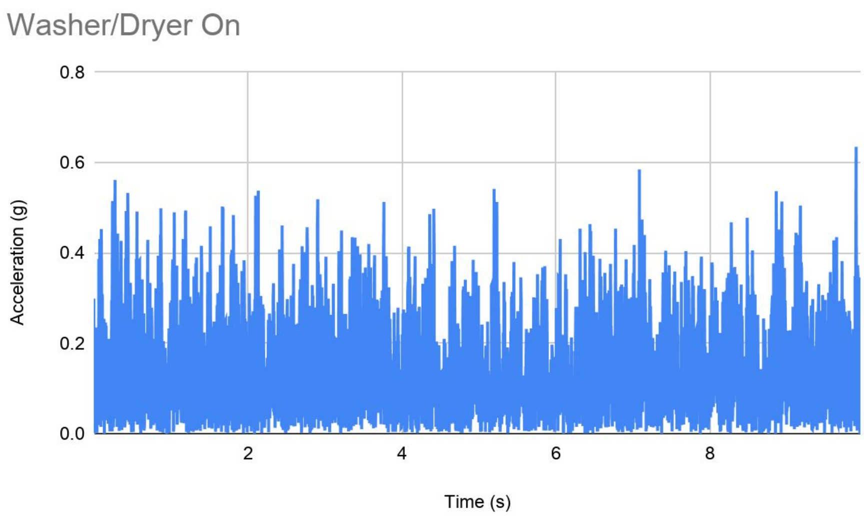

Figure 15: Acceleration data when sensor's machine is on

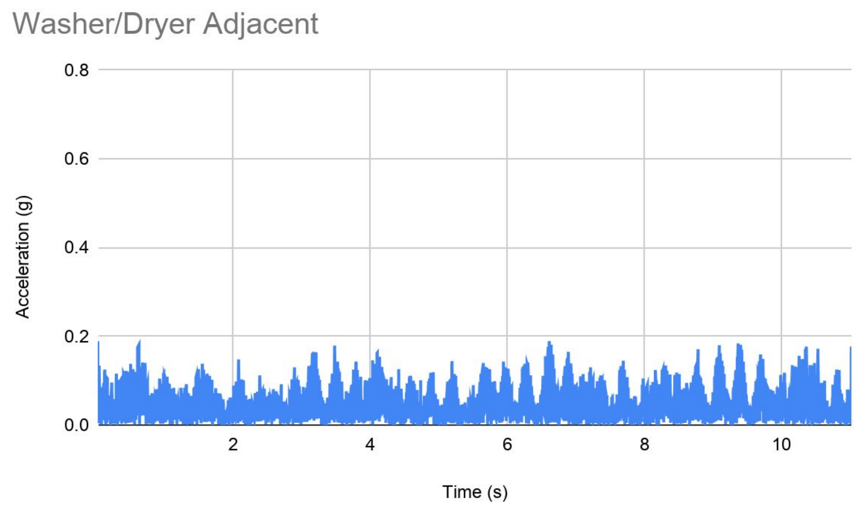

Figure 16: Acceleration data when sensor's machine is off and adjacent machine is on

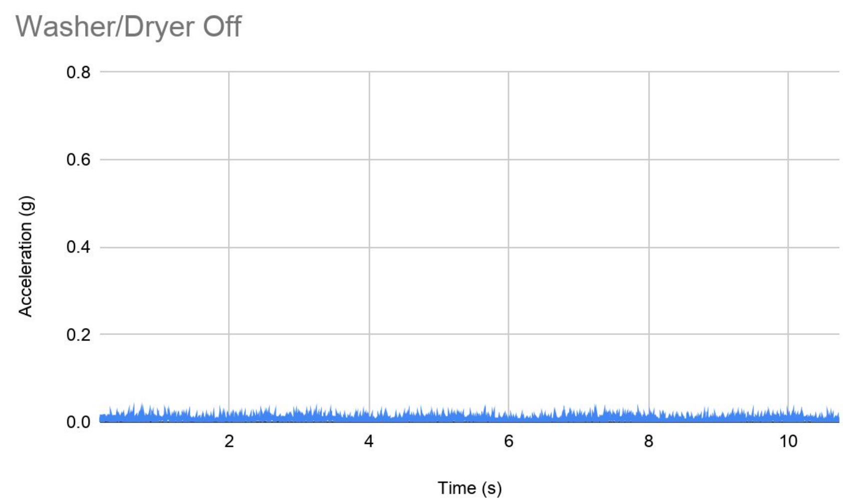

Figure 17: Acceleration data when both machines are off

We primarily care about the acceleration between the attached machine's on state, and the adjacent machine's on state. If we assume high enough tolerance between these two states, it will also be tolerant enough to determine the difference between the attached machine's on state and both machines' off state. Using the data recorded, we get the following results:

State	Normalized Average (g)
On	0.147
Adjacent	0.052

Off	0.007

Using these averages, we determine that the measurement resolution must be 10000 ug at a minimum, and able to record 1g at a minimum. For our accelerometer, we will choose the ADXL335, as it suffices these requirements. It can also function with our 3.3 V input, and has compatible output with our Arduino Nano.
This accelerometer allows for customized measurement resolution via capacitors. To determine the capacitance of our capacitors, we must evaluate the measurement resolution and the bandwidth of the accelerometer. The measurement resolution (in ug) is proportional to the square root of the bandwidth. Using the datasheet for the component, we get that a measure resolution of 10000 ug would allow for a maximum bandwidth of 100 MHz. Without any capacitors, we get a default bandwidth of 550 Hz. Therefore, we will opt out of using capacitors for our accelerometer.

Second, we will determine the tolerance required for the microwave's microphone. To measure this, we place a OnePlus 7 Pro on the back end of the device and measure the total noise using the internal microphone.

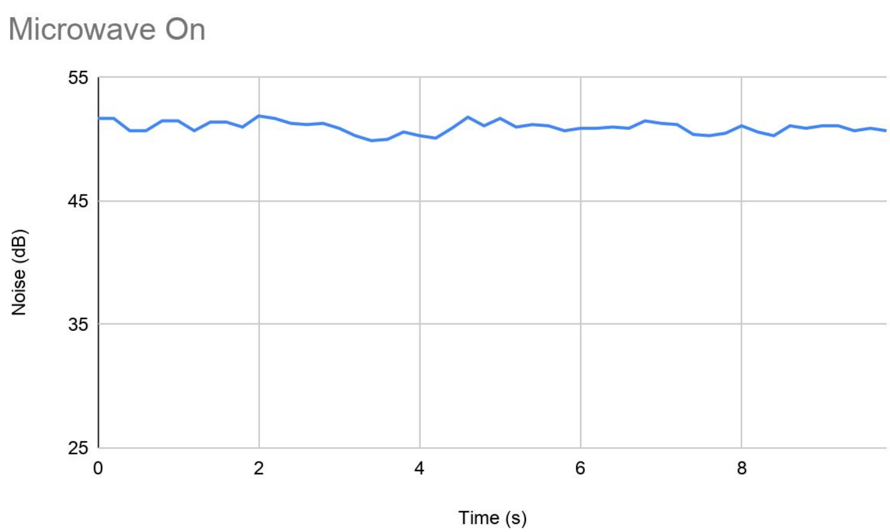

Figure 18: Microphone data when microwave is on

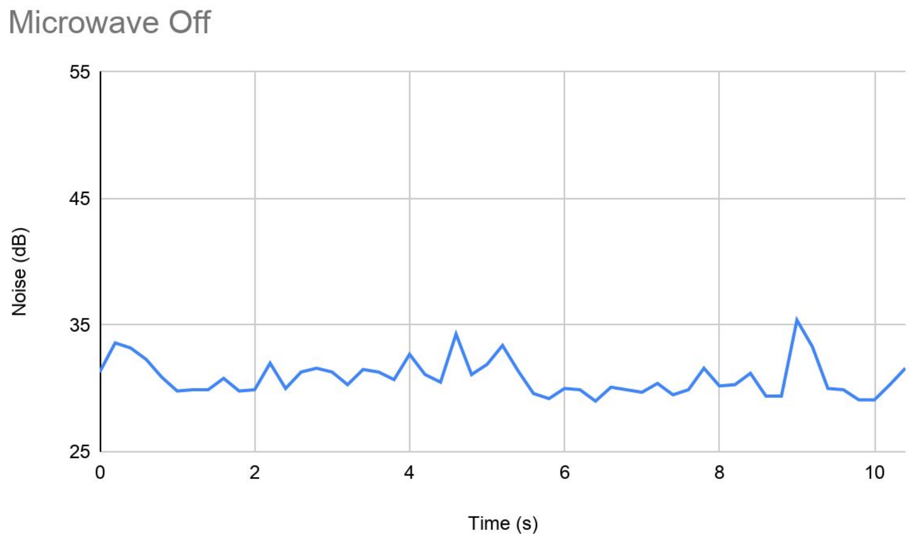

Figure 19: Microphone data when microwave is off

Using the data recorded, we get the following results:

State	Average (dB)
On	51.0
Off	30.9

Using these averages, we determine that the measurement resolution must be 5 dB at a minimum, and can record up to 60 dB at a minimum. For our microphone, we will choose the ADMP401, as it suffices these requirements. It can also function with our 3.3 V input, and has compatible output with our Arduino Nano.

Finally, we must determine the tolerance required for the room's temperature and humidity sensor. As there is no on/off state, but rather a spectrum of values, there are no hard requirements other than the ones we set based out of preference and compatibility. For quality purposes, we will choose a temperature sensor that has a working range of 0-40°C minimum, a response time of 15 seconds maximum, and an accuracy error of ±2°C. For the humidity sensor, we will choose one that has a working range of 0-80% minimum, a response time of 15 seconds maximum, and an accuracy error of ±5%. As such, we chose the AHT20 for both purposes, as it functions with our 3.3 V and has the I2C output interface, which the Arduino Nano is compatible with.

3 Cost and Schedule

3.1 Cost

We estimate our fixed development costs to be $50/hour, 10 hours/week, and working for 50% of the 16 week semester.

$$\frac{\$50}{hour} * 2.5 * \frac{10\ hours}{week} * (16\ weeks\ *\ 50\%)\ =\ \$10000$$

With three partners, this totals to $30000 in fixed development costs.

Part	Quantity	Cost (prototype)	Cost (bulk)	Purch ase
ESP32	3	$20.95	$10.99	Link
ADXL335BCPZ Accelerometer	1	$6.02	$3.23	Link
ADMP401 Microphone	1	$3.95	$3.56	Link
AHT20 Temperature and Humidity Sensor	1	$4.95	$4.95	Link
3.3V Voltage Regulator	3	$1.95	$1.95	Link
Micro-USB Breakout	3	$2.75	$1.50	Link
5V Micro-USB Power Adapter	3	$5.95	$5.95	Link
Speaker	3	$1.95	$0.39	Link
LED	3	$0.35	$0.17	Link
Activation Button (4-pack)	1	$1.60	$0.65	Link
PNP Transistor	3	$0.50	$0.17	Link
Total Cost		115.03	$62.04	

3.2 Schedule

Week	Sam	John	Ian
3/8/2021	Purchase listed components, test all components for full function	Research arduino programming and http client libraries	Set up AWS account, and initialize node server and MongoDB
3/15/2021	Assemble the control units and ensure function	Write code to drive sensors, buttons and speaker	Write API endpoints for both web app and nodes

3/22/2021	Assemble the power units and ensure function	Write basic script for WIFI event loop	Code basic framework for the web app
3/29/2021	Validate the appliance specific sensors	Test basic event driven sending of https requests.	Add data display and login features to the webapp
4/5/2021	Test the full nodes on their intended appliances	Program initial script to prompt user for WIFI credentials	Test the server under load, test user creation and saving user data
4/12/2021	Fix any hardware issues that may have arisen from testing	Test WIFI connection prompt, and reconnection after power down	Review UI, clean up formatting and design for ease of use
4/19/2021	Install the case around the nodes, and validate for protection	Test the full node software setup together	Test data input from multiple nodes on different accounts, final touches
4/26/2021	Work on final paper and presentation	Work on final paper and presentation	Work on final paper and presentation
5/3/2021	Present project	Present project	Present project

4 Ethics and Safety

Our project, covering such a wide range of technologies, has many potential safety concerns that must be mitigated. One concern we must consider before the product even exists is the design process this product will undertake. We will be present in the lab, among soldering kits, chemicals, and high voltage circuits, to name a few potential hazards. To understand the risks, each of our team members has completed lab safety training certification.

We also must consider the operational voltages of each component of our design. The design of the circuit should keep all components within safe voltage and current ranges in order to prevent component malfunction or potential fires caused by short circuits.

Our modules will attach to various surfaces, including the sides of washing machines and humidifiers or other humid environments. This means that the devices will have to be water resistant as well as resistant to dust and other particulates. Over 2,900 fires a year are started by dryer lint18], so mitigating the risk of this hazard if of great importance. Thus the casing for our modules will maintain IP54 rating in order to be sufficiently dust tight and resist splashes of water[19].

IoT devices come with numerous considerable security risks due to their constant connectivity. The risks include physical device tampering, remote attacks over network, RFID spoofing, botnet attacks, eavesdropping, and more.[20] These issues generally fall into 2 categories, physical attack and remote attacks. Handling physical attacks should be less of an issue, given these devices are meant for household use, and have limited capacity to be controlled directly other than being turned on and off. Handling network security, however, is a bigger task. The server will be well protected through Amazon's world class security at AWS, but the devices themselves will need to be secured thoroughly. To do this, all communication will be mediated by the server through HTTPS encryption standard with proper authorization headers.

The web app and accompanying user data is another potential security hazard. Users are often one of the weakest links in the chain of security, creating passwords that are vulnerable to dictionary attack or data breach[21]. To facilitate secure login and access while still providing ease of access, the Oauth 2.0 standard will be used, along with refreshing bearer tokens to keep the web app secure.

The IEEE code of ethics, #7, calls all engineers to "treat all persons fairly and with respect, and to not engage in discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression."[22] We hope that our app can be of assistance to all, especially those that may have a disability that creates difficulty in physical monitoring of their appliances.

5 Citations

[1] The Iot Rundown for 2020, Security Today, 1/13/2020. [Online] Available: https://securitytoday.com/Articles/2020/01/13/The-IoT-Rundown-for-2020.aspx?Page=2

[2] Appliance Life Expectation Chart, Mr Appliance, 2/15/2018. [Online] Available: https://www.mrappliance.com/expert-tips/appliance-life-guide/

[3] Walled Gardens Are Killing the Internet of Things, ConCannon, 12/21/2015. [Online] Available: https://www.concannonbc.com/walled-gardens-are-killing-the-internet-of-things/

[4] The Best Smart Home Security Systems For 2021, PCMag, 2/9/2021. [Online] Available: https://www.pcmag.com/picks/the-best-smart-home-security-systems

[5] The Best Smart Home Devices For 2021, PCMag, 12/21/2020. [Online] Available: https://www.pcmag.com/news/the-best-smart-home-devices

[6] The Best Smart Kitchen Devices For 2021, PCMag, 1/21/2021.[Online] Available: https://www.pcmag.com/picks/the-best-smart-kitchen-appliances

[7] ACM Code of Ethics and Professional Conduct, Association for Computing Machinery, 5/22/2018. [Online] Available: https://www.acm.org/code-of-ethics

[8] Internet of Things, IEEE Standards Association, 2/25/2020. [Online] Available: https://standards.ieee.org/initiatives/iot/index.html

[9] Power Breakout Board Schematic, Sparkfun, 8/7/2017. [Online] Available: https://www.sparkfun.com/products/12035

[10] Voltage Regulator Circuit, Sparkfun, 12/1/2005. [Online] Available: https://www.sparkfun.com/datasheets/Components/LD1117V33.pdf

[11] Microcontroller Schematic, Adafruit, 6/21/2017. [Online] Available: https://www.adafruit.com/product/3591

[12] Activation Button Circuit, Sparkfun, 8/2/2004. [Online] Available: https://cdn.sparkfun.com/assets/8/2/4/e/9/MulticolorButtonModel.pdf

[13] Speaker Circuit, Sparkfun, 8/11/2006. [Online] Available: http://cdn.sparkfun.com/datasheets/Components/General/cem-1203-42-.pdf

[14] Digital Switch Circuit, Sparkfun, 5/2/2003. [Online] Available: https://www.sparkfun.com/datasheets/Components/2N3906.pdf

[15] Accelerometer Schematic, Mouser, 10/30/2016. [Online] Available: https://www.mouser.com/datasheet/2/609/ADXL335-1503897.pdf

[16] Microphone Schematic, Sparkfun, 7/24/2010. [Online] Available:
https://www.sparkfun.com/datasheets/Components/General/ADMP401.pdf

[17] Humidity Sensor Schematic, Sparkfun, 12/5/2019. [Online] Avaliable:
https://cdn.sparkfun.com/assets/d/2/b/e/d/AHT20.pdf

[18] How to Prevent Dryer Fires, Consumer Report, 4/21/2020. [Online] Available:
https://www.consumerreports.org/clothes-dryer/how-to-prevent-dryer-fires/

[19] Ingress Protection And What It Means, DSM&T, 1,2,2020. [Online] Available:
http://www.dsmt.com/resources/ip-rating-chart/

[20] A Primer On IoT Security Risks, Security Intelligence, 2/8/2017. [Online] Available:
https://securityintelligence.com/a-primer-on-iot-security-risks/

[21] Encouraging Users to Improve Password Security and Memorability, International Journal of Information Security, 4/11/2019. [Online] Available:
https://link.springer.com/article/10.1007/s10207-019-00429-y

[22] IEEE Code of Ethics, IEEE Foundation, 6/4/2020. [Online] Available:
https://www.ieee.org/about/corporate/governance/p7-8.html