Household Device Ecosystem

By

John Armgardt

Sam Atac

Ian Goodwin

Final Report For ECE 445, Senior Design, Spring 2021

TA: Ali Kourani

05 May 2021

Group 33

Abstract

Our project, the household device ecosystem, consists primarily of sensor “nodes”. These nodes use built-in sensors to receive data from the household appliance they are attached to, and relay it to a web app for users to view. The user can also receive notifications from the nodes when the appliance changes state or reaches a certain value. During the course of development, we met all of our project goals, and left room for future expansion of the product with modular design.

Contents

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. Having greater home connectivity can lead to improvements in energy savings, environmental impact, security, and comfort[3]. These are benefits that we cannot afford to ignore.

Our solution to these issues was to develop a series of WiFi-connected devices containing sensors focused on monitoring a wide range of current appliances. These devices allow for monitoring/control of existing household devices without the need to buy dedicated smart devices or replace existing household appliances. Our devices are also open source and as modular as possible, in order to allow for future expansion of the ecosystem. The appliances we chose specifically were the washer/dryer, the microwave, and the room's humidity.

1.2 Goals and Features

1.2.1 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

1.2.2 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

2. Design

2.1 Design Overview

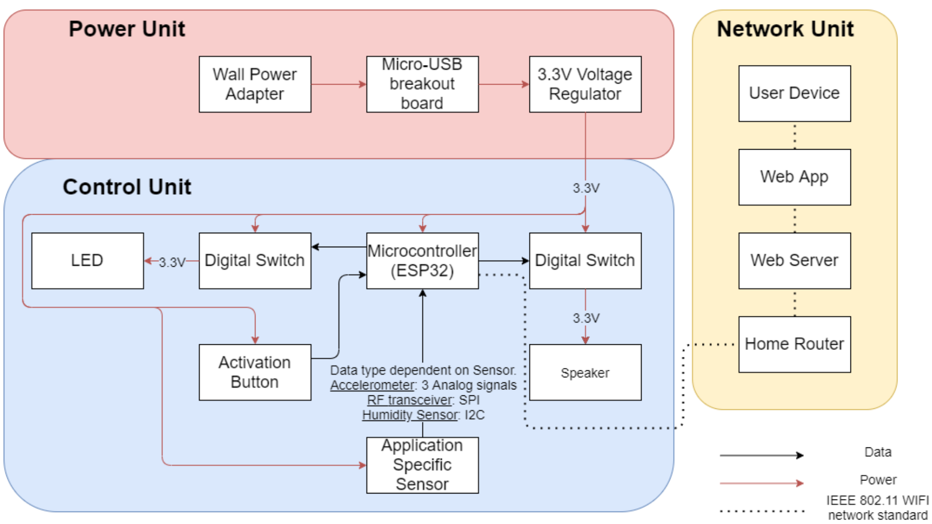

Fig. 1: Block Diagram

Our design consists of a series of Nodes. Each Node has three components: the Control Unit, the Power Unit, and the Network Unit (which is shared between all the modules). These devices all interface with each other and the user through their shared Network. 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. 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 radio transceiver, and for the room's humidity, 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.

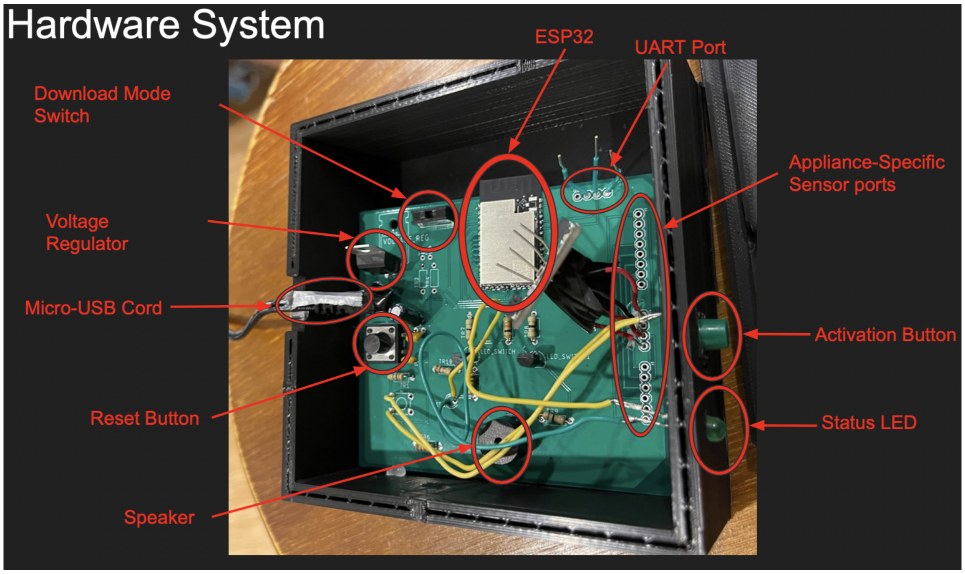

Figure 2: Labeled photo of final product

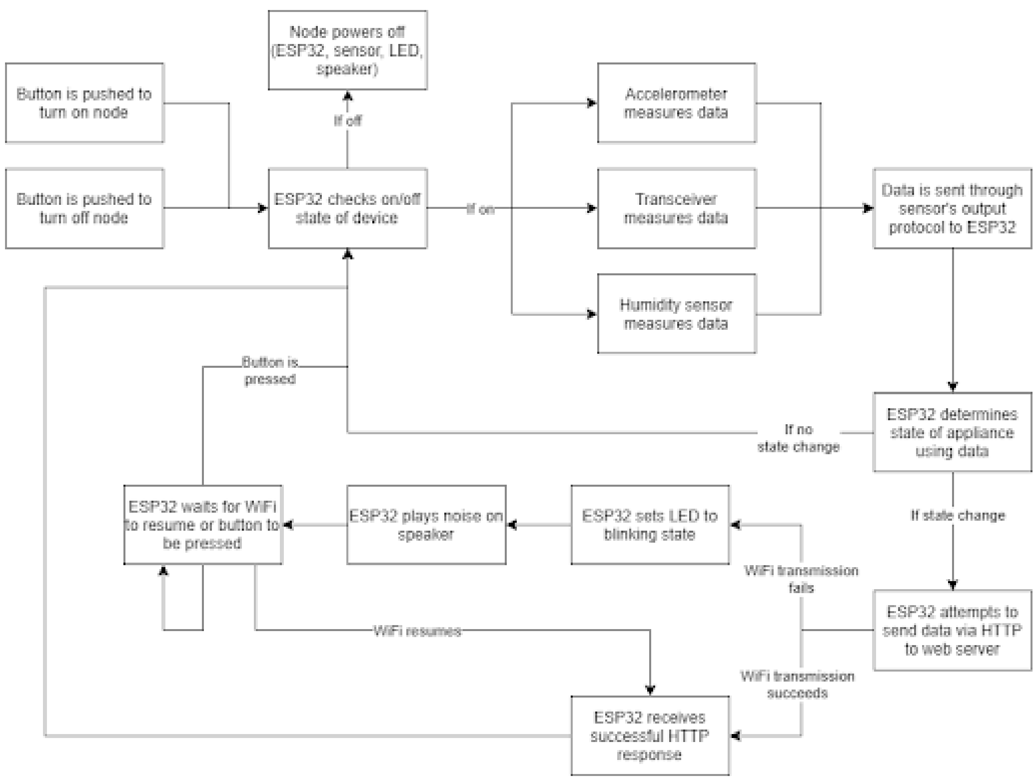

Figure 3: Flowchart of Algorithm on ESP32

2.1.1 PCB

Our PCB was designed with modularity in mind. Primarily, it contains all the necessary components and peripherals. These include the microcontroller, the voltage regulator, the digital switches, the LED, the speaker, the activation button, and any necessary resistors, capacitors, and switches to have the components operate. Additionally, each PCB contains holes and lines for sensors with 4, 6, and 8 pins. These pins are used for our appliance-specific sensors. To keep our PCB modular, these pins could be theoretically swapped with any sensor that has the same number of pins.

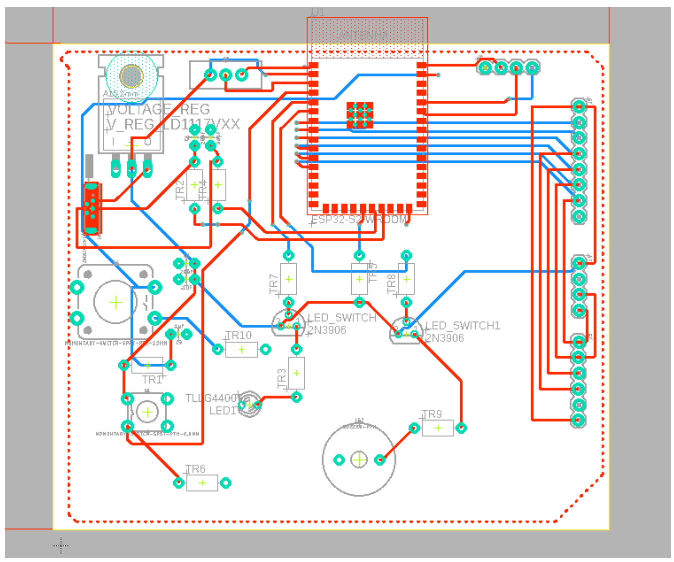

Figure 4: Eagle Drawing of PCB

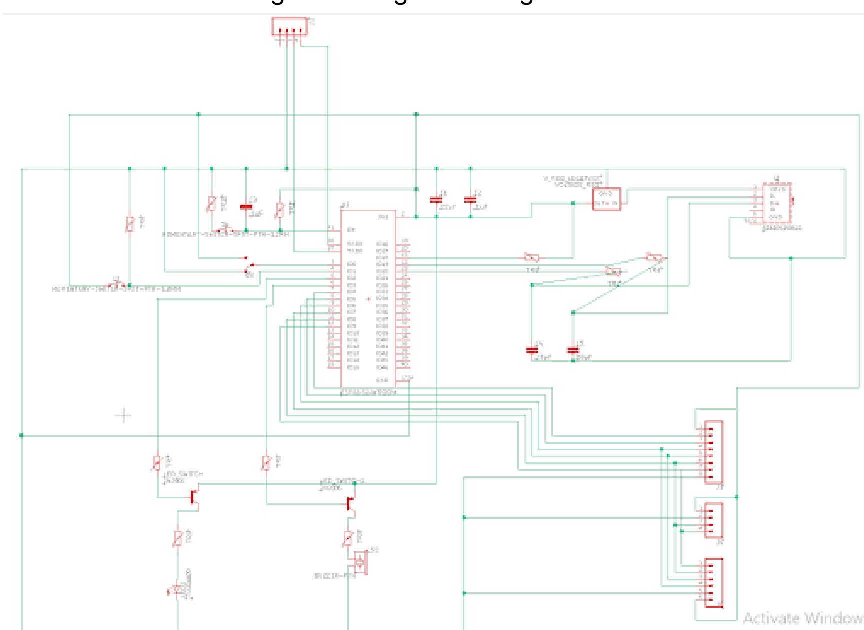

Figure 5: Eagle Schematic of PCB

2.2 Control Unit

The Control Unit accepts data from the sensors, performs calculations related to the appliance, packages that data, and sends it out to the web server via WiFi. The Control Unit communicates with the Network Unit 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 accepts 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, an SPI signal from the Transceiver, and I2C in the case of the Humidity Sensor. The Microcontroller transmits 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.

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.

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 is powered by a Digital Switch which is triggered by a GPIO pin on the Microcontroller.

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 receives power through one of the Digital Switches which output 3.3V and are 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.

2.2.5 Digital Switch

We use 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.

2.2.6 Appliance-Specific Sensors

In order to detect the state of a certain appliance, we use specific sensors to measure data related to that appliance. The sensors are 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 use 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 has three analog wires connecting to the Microcontroller, each with a voltage corresponding to either the X, Y and Z acceleration.

2.2.6.2 Microwave: Transceiver

In order to detect when our Microwave is finished we will be using a Radio Transceiver. Using the Transceiver we will detect the rf waves that escape the Microwave while it is running and use that to determine the status of the device. We use the nRF24L01.

2.2.6.3 Room's Humidity: Humidity Sensor

The Humidity Sensor is used to detect when a room comes to the desired humidity, it is intended to be used with either a humidifier or dehumidifier. We use the Qwiic Humidity AHT20. The Humidity Sensor communicates with the Microcontroller via an I2C signal.

2.3 Power Unit

2.3.1 Wall Power Adapter

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

2.3.2 Power Breakout

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

2.3.3 Voltage Regulator

The Voltage Regulator, a LD1117V33 chip, regulates 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.

2.4 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.

2.4.1 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.

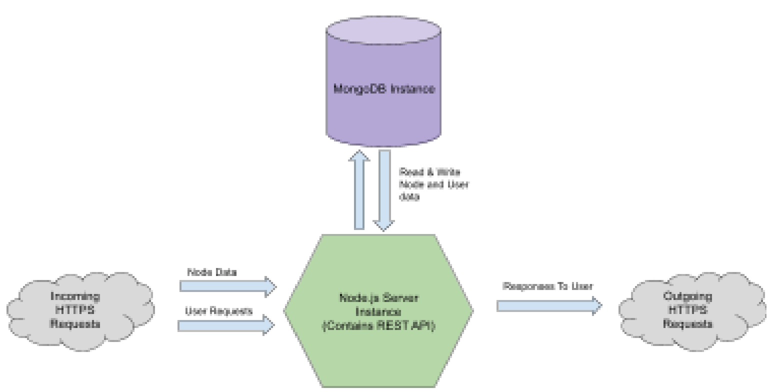

Figure 6: Web server flowchart

2.4.2 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.

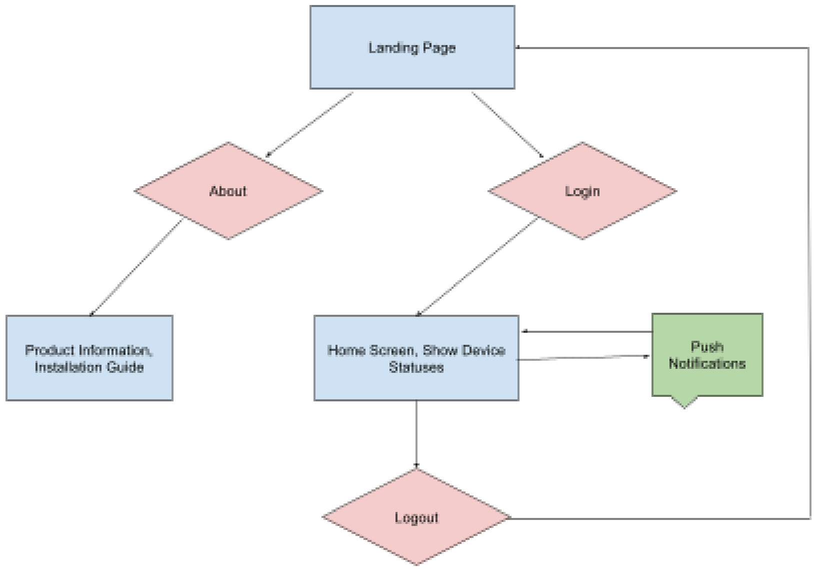

Figure 7: Web app flowchart

2.5 Case

The case is a 10 cm X 10 cm X 4 cm box with openings for the activation button, status LED and power cable. The case and the top are separate pieces. Within the case we glued the PCB to the bottom to prevent any movement within the box that could damage components.

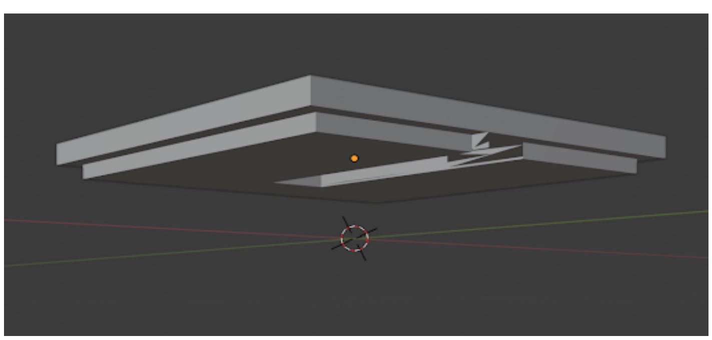

Figure 8: CAD Drawing of Lid of Case

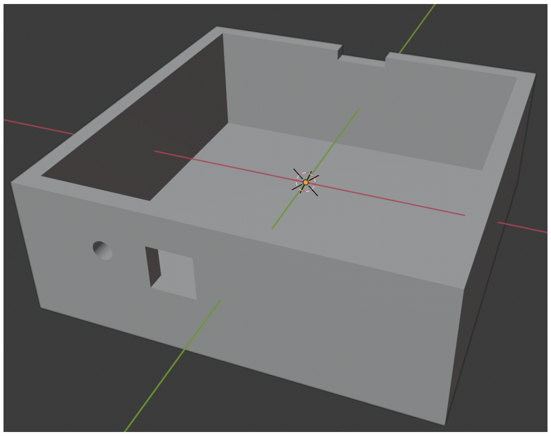

Figure 9: CAD Drawing of Base of Case

3. Design Verification

3.1 Accelerometer Verification

Testing Setup: The ADXL335 was connected to an Arduino, and the setup was placed on top of a washer, as shown in the photo below

Figure 10: Setup of ADXL335 Verification

The z-axis value was outputted to a terminal and graphed across three states: when the washer was on, when the washer was off and the dryer was on, and when both appliances were off.

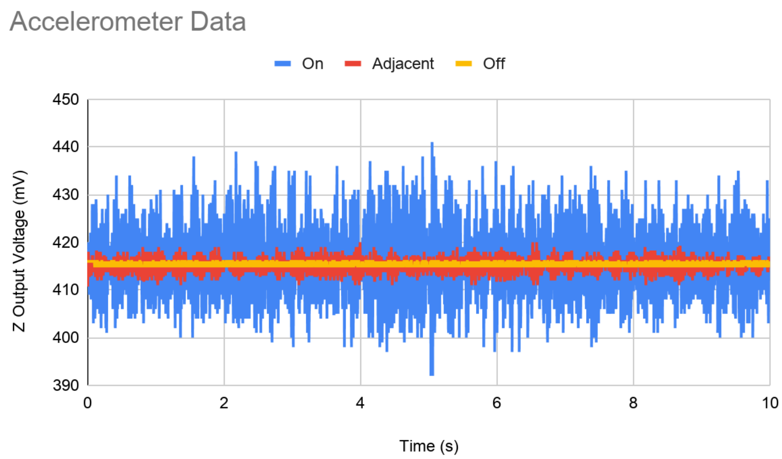

Figure 11: Z-Axis Acceleration Data Across 3 States

Results: The accelerometer was able to successfully differentiate between the three states. Because of its precision, it was used in the project without any issues.

3.2 Transceiver Verification

Testing Setup: The nRF24L01+ was connected to an Arduino, and the setup was placed next to a microwave, as shown in the photo below.

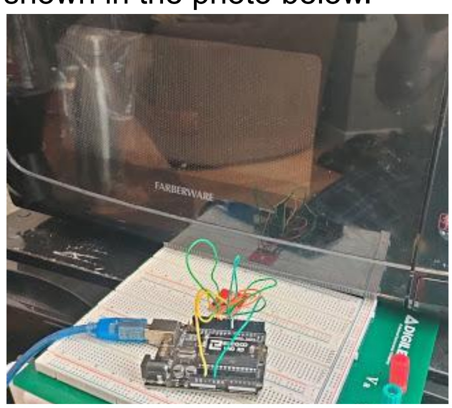

Figure 12: Setup of nRF24L01+ Verification

The 2.450 GHz channel would be open for a brief moment of time, and the number of hits received was outputted to a terminal and graphed across two states: when the microwave was on and off. The data was then averaged across the last 10 values, to remove random noise present.

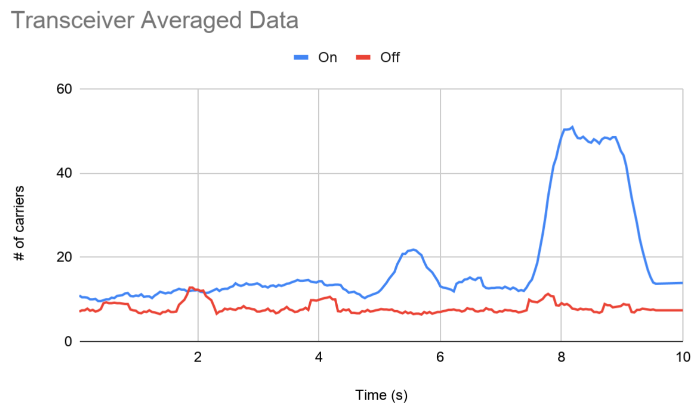

Figure 13: Channel Hit Data Across 3 States

Results: The transceiver was able to successfully differentiate between the two states. Because of its precision, it was used in the project without any issues.

3.3 Humidity Sensor Verification

Testing Setup: The AHT20 was connected to an Arduino, and the setup was placed in a room with a humidifier, as shown in the setup below.

Figure 14: Setup of AHT20 Verification

The humidity value was then outputted to a terminal and graphed across three states: when the humidifier outputted 20% RH, 50% RH, and 80% RH.

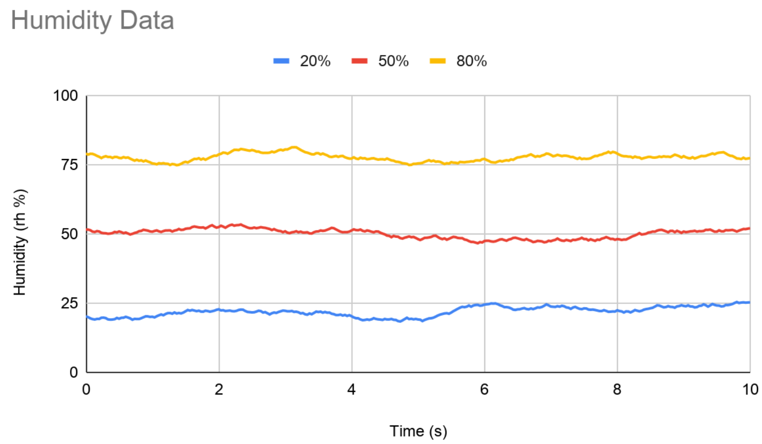

Figure 15: Humidity Data Across 3 States

Results: The humidity sensor was able to successfully differentiate between the three states. Because of its precision, it was used in the project without any issues.

4. Costs

4.1 Parts

Part	Quantity	Cost (prototype)	Cost (bulk)	Purchase
SparkFun Purchase 1				
ADXL335 Accelerometer	1	$14.95	$12.71	Link
nRF24L01+ Transceiver	1	$20.95	$17.81	Link
PNP Transistor	3	$0.50	$0.17	Link
3.3V Voltage Regulator	3	$1.95	$1.95	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
5V Micro-USB Power Adapter	3	$5.95	$5.95	Link
Shipping + taxes		$12.69		

Adafruit Purchase 1				
ESP32	3	$20.95	$10.99	Link
AHT20 Temperature and Humidity Sensor	1	$4.50	$3.60	Link
Shipping + taxes		$8.93		
Mouser Purchase 1				
Micro-USB Breakout	1	$1.98	$1.50	Link
Shipping + taxes		$8.99		
Mouser Purchase 2				
Micro-USB Breakout	1	$1.98	$1.50	Link
Shipping + taxes		$8.99		
PCBway Purchase 1				
Printed Circuit Board + Shipping + taxes	1	$127.00	$12.00	
Shipping + taxes				
PCBway Purchase 2				
Printed Circuit Board + Shipping + taxes	1	$127.00	$12.00	
Total Cost		$434.51		

4.2 Labor

Our simulated labor costs were set to have a wage of $50/hour, with 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 on the project, this totals to be $30000 in labor costs.

5. Conclusion

5.1 Accomplishments

Throughout the course of the semester, we accomplished every major goal that we set for the project. We successfully created three different nodes for the washer/dryer, microwave, and humidity of a room. These nodes were all capable of connecting to the internet, and relaying their data to the web server. They also all had the ability to function with the speaker and LED when disconnected from WiFi. Our web app was also successful, being easy to use and safe in the final design. We even decided later on to add a live graphing feature to the app, which was outside of the original design, but was a welcome addition nonetheless. Our nodes were also all well under our $50 price target, when looking at bulk prices.

5.2 Ethics

Our project was relatively light on ethical concerns, the main one being user privacy and security. IoT related 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.[4] These issues generally fall into 2 categories, physical attack and remote attacks. Handling physical attacks is 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, was a bigger task. To do this, we made sure all communication was mediated by the server through HTTPS encryption standard with proper authorization headers.

The web app and accompanying user data was 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[5]. To facilitate secure login and access while still providing ease of access, We set up secure password requirements with a session ID system to protect user data behind safe login credentials.

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.”[6] 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.

5.3 Future Work

Our project was left ready for future work based on our focus on modularity. The final node design we used has multiple sets of GPIO pin through hole ports on the board, allowing for a wide range of sensors to be used. This design choice not only reduced our costs by being able to manufacture only a single type of PCB, but also allows the potential use of sensors outside of our original plans. The web app is also able to account for these new nodes with its flexible design. Any amount of nodes can be registered on a user, with any name and value configuration the user desires.

Some potential ideas we had for future additions to our ecosystem included a temperature sensor, a motion triggered sensor to detect passerbys, and a gyroscope to detect the tilt of an object. We also have realized that the current sensors we already use can be calibrated to read a whole host of different value ranges, and thus be used on multiple products. For example, the accelerometer could be recalibrated to sense the motion of a garage door. Or the humidity sensor could be set to notify the user within a range of humidity suggesting that the shower has been turned on. The possibilities for this platform are numerous with some imagination.

References

[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] 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

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

[5] 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

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

Appendix A

Requirements and Verifications Table

Requirement	Verification	Status
1. Power unit 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.	Complete
2. Power Unit Outputs >500 mA at 5V	2. 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.	Complete
3. Power Unit Outputs > 200 mA at 3.3V	3. 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.	Complete

	d. Ensure that the voltage is 3.3V and the current > 200 mA.	
4. Wifi Module is Able to communicate over the IEEE 802.11 WIFI network standard with the Web Server with sub 1 second latency.	4. 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 the timer is less than 2 seconds(Due to the round-trip).	Complete
5. Wifi Module Receives data over I2C from Humidity Sensor with a latency of < 50 ms	5. 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	Complete
6. Button is Able to be easily pressed, with no more than 1 Newton of	6. We will put an moderately sized apple on the button and see if it depresses	Complete

Force necessary.		
7. Able to clearly differentiate solid green, blinking green and unpowered LED.	7. 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	Complete
8. Able to hear speaker from another room, we estimate at >= 90 dB.	8. a. Activate Speaker in kitchen b. Walk into bedroom ~40 ft c. Check if the speaker is still audible	Complete
9. Able to receive internet failure signals and inform the user.	9. 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	Complete
10. Able to allow voltage to flow from the switch collector to the emitter when a 12mA, 3.3V	10. a. Hook collector up to a 3.3V source b. Connect GND to base c. Use voltmeter to ensure	Complete

signal is applied to the base.	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	
11. Accelerometer must have a measurement resolution of .01 m/s^2 at a maximum	11. a. Connect accelerometer to breadboard with power and connect external accelerometer (e.g. OnePlus 7 Pro) to setup b. Drop breadboard setup and measure data on accelerometer c. Ensure data is within .01 m/s^2 with external accelerometer's data	Complete
12. Accelerometer should have a total measurable range of at-least -1 g to 1 g	12. a. Connect accelerometer to breadboard with power and connect external accelerometer (e.g. OnePlus 7 Pro) to setup b. Drop breadboard setup and measure data on accelerometer c. Ensure data is within .01 m/s^2 with external accelerometer's data	Complete
13. Microphone can measure the difference between 5 dB sounds.	13. Play two sounds at 5 dB apart and compare recorded voltages	Complete
14. Microphone can record up to 60 dB at a minimum.	14. Play sound getting increasingly louder until 60 dB and compare data recorded	Complete
15.	15.	Complete

Temperature sensor Working range of 0-40°C with Accuracy error of ±2°C.	a. Place sensor in 0°C and 40°C environments and verify correct voltages b. Place sensor in 20°C environment, wait for sensor 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	
16. Humidity sensor Working in a range of 0-80% with minimum Accuracy error of ±5%.	16. 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	Complete
17. Server receives HTTPS Request from Node with less than 200ms latency	17. Have Node device send out HTTP request, with timestamp. Log the request on server with timestamp on receival. Verify the difference is under 200ms	Complete
18. Server Receives HTTPS Request from User with less than 200ms latency	18. Have User device send out HTTP request, with timestamp. Log the request on server with timestamp on receival. Verify the difference is under 200ms	Complete
19.	19. Set up test GET and POST API	Complete

Server Reads from and Writes data to MongoDB Instance	calls. POST a piece of data remotely, then retrieve it via the GET call.	
20. Server performs under load, can handle up to 1000 API Requests a minute	20. 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	Complete
21. User is able to log in and out of the application	21. Access the website, and enter account credentials. Verify that the user is taken to the home screen, and the correct node data is displayed.	Complete
22. User can receive push notifications	22. On the home screen, select the option for notifications. Trigger one of the nodes manually, and verify that the user received a push notification.	Complete
23. User is able to see data from their nodes	23. 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.	Complete
24. Internal components are not moved when case is moved	24. 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	Complete
25. Case is resistant to dust and water up to	25. a. Splash water on case, power on device and	Incomplete

IP54	ensure it is still functional b. Throw dust on case, power on device and ensure it is still functional	