

School of Engineering

Department of Electrical and Computer Engineering

**Smart Home Device**

Submitted By:

Neil Surti

Imad-Uddin Siddiqui

Thorson Dai

Austin Su

Team Project Number: 36

Advisors:

Hana Godrich

Samuel Ramrajkar

March 18, 2018

Submitted in partial fulfillment of the requirements for senior design project

**Electrical and Computer Engineering Department**

**Rutgers University, Piscataway, NJ 08854**

**Project Abstract:**

Most smart home devices provide limited applications, such as just security alerts or general information, such as temperature or humidity. We wish to design and create a proof of concept for a multi-purpose smart home device that, when placed inside a room, has multiple applications that, at the moment, is intended to be used by one or more users. The first application would be to gather information such as temperature, humidity, and light levels. The sensor information can then be viewed remotely on any device with internet connectivity through a web application. The end goal is to provide a smart home device form which different room conditions can be monitored and controlled. For example, when a person enters a room with the device, the lights will automatically turn on. The device will then know that the user is in the room and disable any security features. The device will be able to use motion detection to act as the security trigger for when the user is not in the room. Along with security measures, we wish to implement some type of control over appliances.

**Table of Contents**

1. INTRODUCTION 4  
2. METHODS / RESULTS / APPROACH 5  
 2.1. Methods 5  
 2.2. Use of Standards 8  
 2.3. Product Results 9  
3. COST AND SUSTAINABILITY ANALYSIS 10  
4. CONCLUSIONS / SUMMARY 11  
5. ACKNOWLEDGMENTS 12  
6. REFERENCES 13

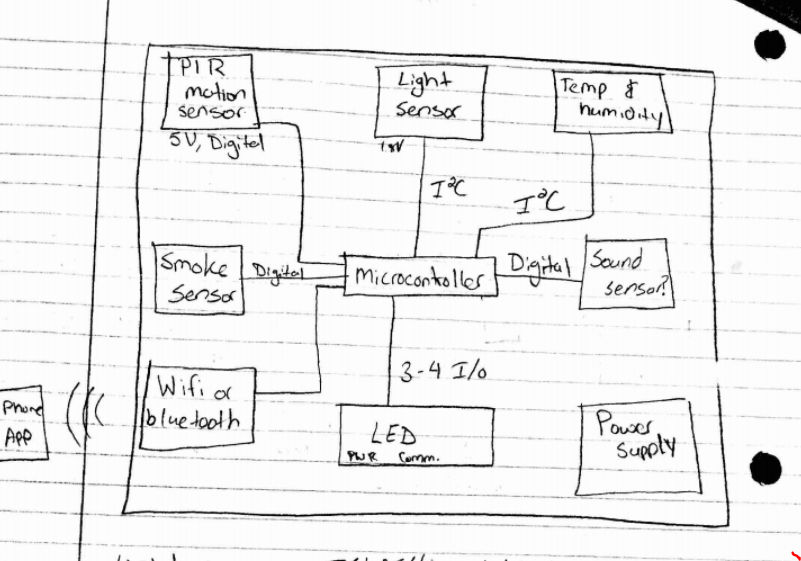
**1. INTRODUCTION**

In the 21st century, smart home devices have become a normal part of everyday life for many in the Western world. By 2020, the technology consulting firm Gartner estimates that 21 billion devices connected in the Internet of Things [1]. Even now, they have become the most common devices in a household, with electronics such as smartphones and smart home devices. Devices like Alexa, Nest, and Google home all provide a assistant at home that can do various things ranging from entertainment to control of environment. These devices, however, are expensive, have limited integration, are connected but not integrated, and are proprietary and inflexible. Our objective with our capstone is to create a smart home device platform that is cheap, multi-functional, easily integratable to other smart devices, an open-concept and flexible.

**2. METHODS / RESULTS / APPROACH**

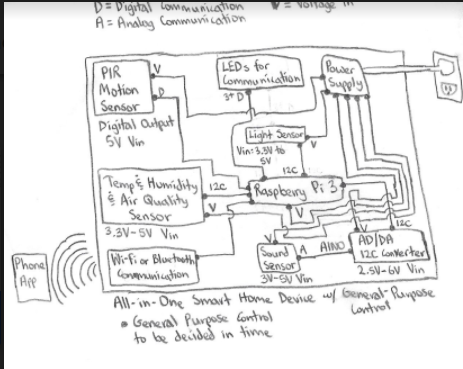
*2.1. Methods*

In order to begin with our project of creating a smart home device, we first began with creating a basic block diagram of what sensors will be included along with the interfaces for each sensor. This allowed us to figure out what our requirements will be for the type of microcontroller which must include the number of input/output ports, wifi capability, and GPIO pins. Also the scope of our project since our time constraint is only 3-4 months we are forced to only include features that we will be able to implement in this time period.

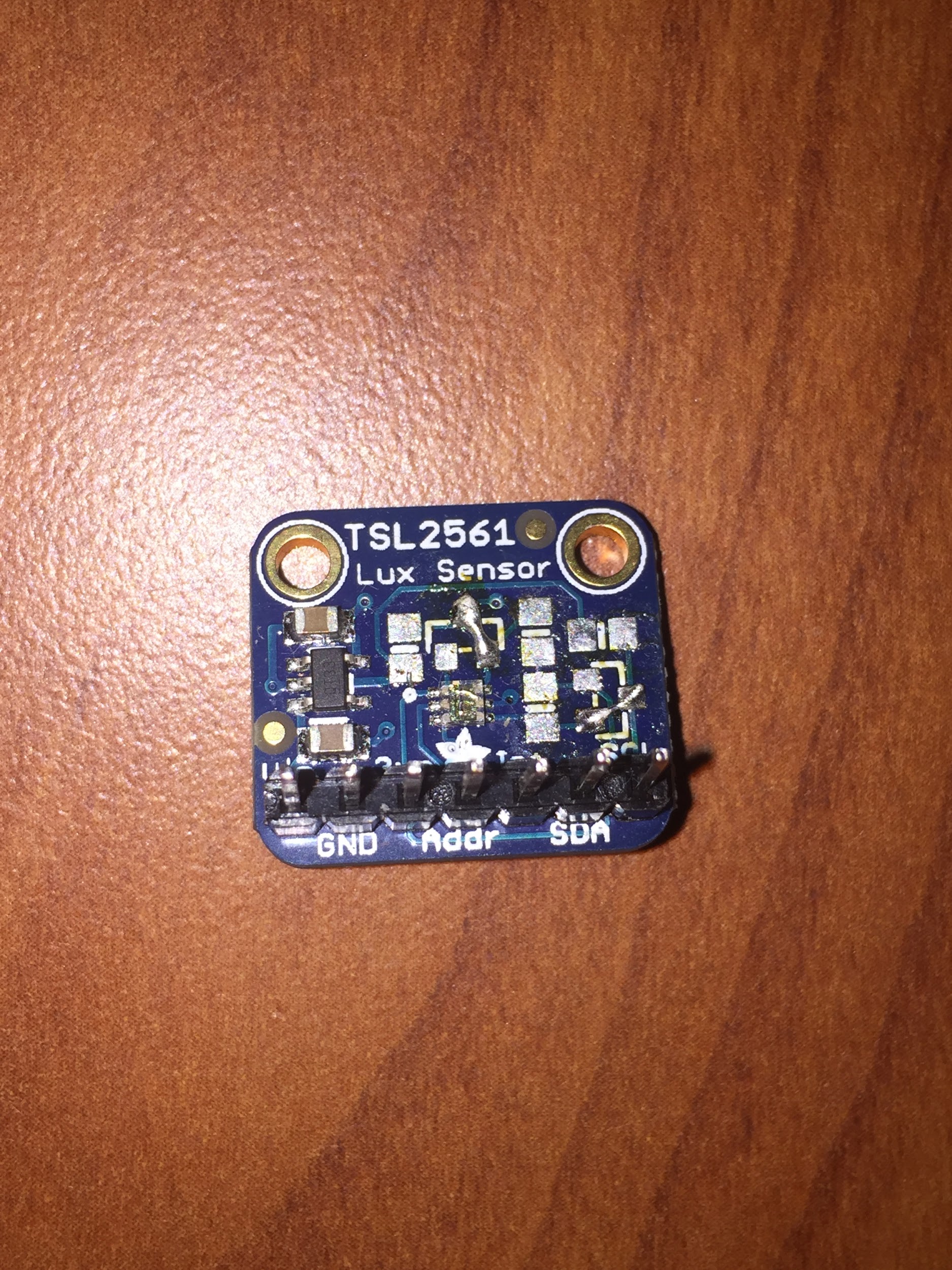


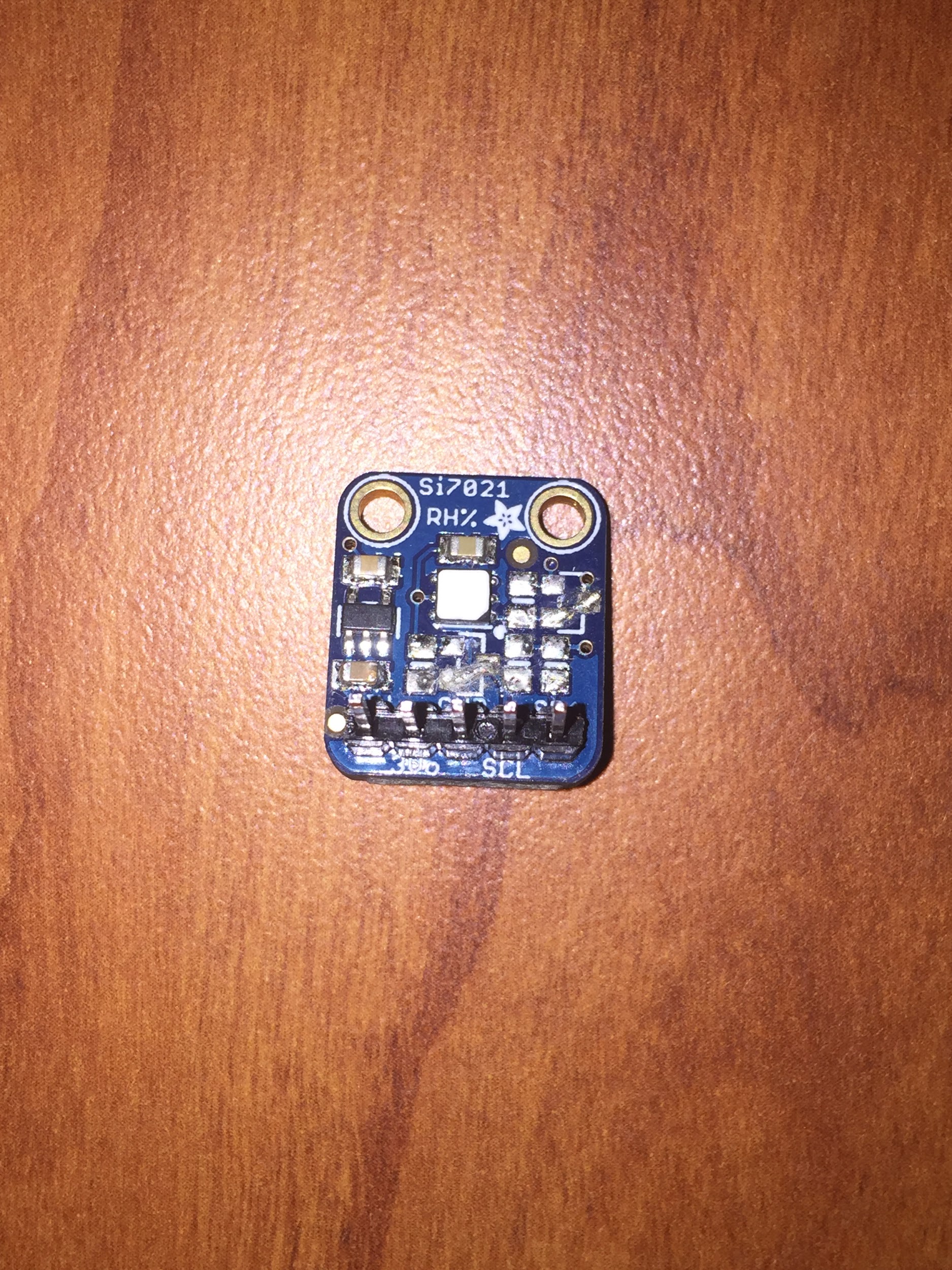
By: Neil Surti

By starting with this basic block diagram we were able to figure out the microcontroller and communication that will be used between the sensors and microcontroller. After doing some research, we decided that the Raspberry PI would be the best microcontroller to use since it already has WiFi capabilities and includes numerous serial communication protocols By using I2C(Inter-integrated Circuit) compatible sensors, we are able to have one master and as many as 128 sensors (slaves) under this bus. Thus allowing us to add as many sensors we feel necessary with little effort. After determining the bill of materials and ordering our parts we created a more comprehensive block diagram that shows everything in more detail (created by Imad).



After creating the bill of materials (BOM) and receiving our sensors, we were able to begin our design for the smart home device. In order to use these sensors we first had to solder the level translators off of them and solder the proper connections for serial data/clock lines for the I2C bus. Figuring out the exact pins from the schematic was a bit challenging along with desoldering the components. In order to use these sensors we first had to create drivers for both of them that has the proper functions that will call the values from the register locations. After creating the drivers for both of these sensors, we created a python script that calls the functions from these sensors every minute and updates them using a pipe which connects to the javascript file. Creating the pipe between the two process was our first road block, since it took the hardware team some time to figure out how pipes work and how to implement them. After successfully implementing the pipe we were able to connect the sensor values from the device to our website. We went on to have our scripts run in the background on boot-up and after successfully connecting to WiFi to finalize and improve upon on our functionality through the usage of the rc.local script. We then went on to create a PCB to have our sensors connected efficiently in one piece and thus easily use our sensors for testing and data collection. Finally, to finish off our capstone, we designed and 3D printed an enclosure for our Raspberry Pi and sensors so everything can be properly and neatly displayed.





By: Imad By: Neil

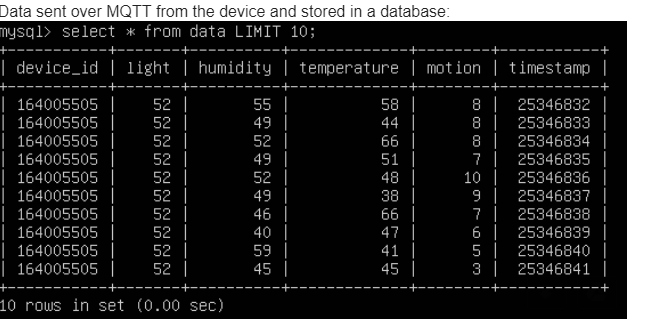
[Device Enclosure] [PCB Design and Layout]

By: Neil and Thorson By: Neil

This image shows successful communication between the device with sensors connected and the database on the server using MQTT.

[Images of the timer\_modified.py sending sensor information and of the pubsub.js subscribing to the client and receiving sensor information]

By: Imad, Neil, and Thorson



By: Thorson

This image is from our most up-to-date website interface for the smart home device. The UI and the sensor triggers were implemented on the website.

[Images of the website and the sensor triggers]

By: Thorson and Austin

*2.2. Use of Standards*

Standardized Network Technologies: IEEE 802.15.4, IEEE 802.11 a/b/g, IPv4 and IPv6, TCP, []

Standardized Software Development Tools: Java Software Development Kits, Python, []

Hardware Standards: I2C, IPC standards, []

Operating Systems, Open Source Standards, and Software: Raspbian (Unix), []

*2.3. Product Results*

Trail data will be collected this week, 3/19/18 and will be used to display all data on website along with testing the schmitt trigger.

**3. COST AND SUSTAINABILITY ANALYSIS**

The cost of our prototype was roughly $85 in parts, with most of the cost coming from our Raspberry Pi 3, which included a casing, a microSD card, and a plug to power it up [2][3][4][5]. The cost of the server itself is mostly negligible and thus wouldn’t really be needed to be passed to the consumer, even with multiple people using it at once. It may add up once thousands upon thousands of people are using the device, however. But, when mass produced, it should be cheaper as we would be ordering our components in bulk and, instead of using a Raspberry Pi 3, we would be using a Raspberry Pi Zero or perhaps a cheaper microcontroller. For instance, buying the Si7021, our light sensor, in the hundreds reduces the price from $6.95 to $5.56, which would go down even further if order in the thousands, if not hundreds of thousands [2]. The same would apply for the costs of our other sensors. In the worst case scenario, if we merely go by the price reduced by ordering in the hundreds, use a Raspberry Pi Zero W instead of a Raspberry Pi 3 from Adafruit, and have the product powered by batteries instead, the price of the hardware would be reduced to roughly $28.50, with the server adding very little to the cost of the product itself [2][3][5][6]. The price would be considerably reduced from there if we order in the thousands and go for a microcontroller cheaper than a Raspberry Pi Zero W. But, going off of the worst case scenario, if we markup the price to $31.99, that should make up for the cost in materials, the almost negligible costs of the server, and provide a small profit per unit sold. That would, of course, still be cheaper than most smart home devices, such as a Google Home or Amazon Echo, which go for over a $100.

Our product would be a key device for general consumers, especially for worried homeowners who want to keep tabs on their homes. For a low price of $31.99, our consumers would have a smart home device that would monitor a key area of their home, such as the living room, alerting them of potential burglars and thieves when they’re away from home while also providing them with the time that activity was detected. Other useful information, such as the inside temperature and humidity, would also be provided and every bit of information would be accessed through a website that they can access on their smartphones, tablets, and computers.

**4. CONCLUSIONS / SUMMARY**

In the end, we managed to finish everything within the scope of our capstone but less than what we may have initially intended to implement. We wanted to go on to implement control settings, such as turning on and off lights if it’s too bright, for example. As everyone in our group became progressively busier throughout the semester and certain tasks taking more time than anticipated, we had to forego adding in control capabilities. Having said that, we are still satisfied with what we pulled off and we were able to properly finish and integrate everything else. On the hardware design side of the team, we have managed to make the drivers for our sensors, de-solder pull-up resistors and level-shifting transistors from the I2C sensors, design and order PCBs for our sensors so we can more easily implement all of them together, and create an enclosure for our device and PCB. On the more software-orientated portion of our tasks, we went on to implement timers to sample information from our sensors at a set rate every second and then average the information once per minute, implement rc.local to have our scripts work on boot-up and after establishing an Internet connection, and work out how to pipe the information through a FIFO pipe so the software design side of the team can make use of the information for the server. The software / server side of the team was able to obtain the information, process and store it into a server, and then make use of the stored information on a website, with sensor triggers and data visualizations completed.

The significance of our device compared to other smart home devices on the market is that it’ll be cheap while still being able to offer and display relevant information, inform the user of potential activity within their house, and be able to control other applications. Overall, it’ll offer great value compared to other contemporary devices on the market.

**5. ACKNOWLEDGEMENTS**

We have all taken effort into this capstone design project. However, it would not have been possible without the kind support of our friends and family and the help of our advisors, Professor Hana Godrich and Samuel Ramrajkar. We would like to extend our sincere thanks to all of them.

**6. REFERENCES**

[1] Boorstin, Julia. “An Internet of Things that will number ten billions.” *CNBC News*. 1 Feb.

2016. Web. 13 Mar. 2018, https://www.cnbc.com/2016/02/01/an-internet-of-things-that-w ill-number-ten-billions.html.

[2] “Adafruit Si7021 Temperature & Humidity Sensor Breakout Board.” *Adafruit*. Web. 14 Mar.

2018, https://www.adafruit.com/product/3251.

[3] “Adafruit TSL2561 Digital Luminosity/Lux/Light Breakout.” *Adafruit*. Web. 14 Mar. 2018,

https://www.adafruit.com/product/439.

[4] “Amazon.com: Canakit Raspberry Pi 3 Kit with Clear Case and 2.5A Power Supply:

Electronics.” *Amazon*. Web. 14 Mar. 2018, https://www.amazon.com/CanaKit-Raspber ry-Clear-Power-Supply/dp/B01C6EQNNK/ref=sr\_1\_6?s=electronics&ie=UTF8&qid=1521073593&sr=1-6&keywords=raspberry+pi+3.

[5] “PIR (motion) sensor.” *Adafruit*. Web. 14 Mar. 2018, https://www.adafruit.com/product/189.

[6] “Raspberry Pi Zero W.” *Adafruit*. Web. 14 Mar. 2018,

https://www.adafruit.com/product/3400.