ActiveHouse

Project website https://irwintr.github.io

Prepared by Trenton Irwin

March 28, 2017

Declaration of Sole Authorship

I, Trenton Irwin, will be completing all of the work related to this project including the hardware, database, web service, and mobile application. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included. It is mandatory that all sources of information be acknowledged in the SRS. Plagiarism is unethical and a candidate suspected of plagiarizing may be referred to the Complaints Committee.

Proposal

Proposal Table of Contents

The ActiveHouse Project would be a complete home monitoring system consisting of multiple Arduino based sensor hubs in each room of your home wirelessly connecting to a Raspberry Pi based base station which sends the data to the server where it can be read by the app. Each sensor hub would be capable of reading temperature, humidity, luminosity, carbon monoxide and other gas levels, water consumption, power consumption, and it would have the ability to toggle the built-in light. The power and water consumption sensors are detachable as they would not necessarily be used in each room, thus creating a more modular design. The app would allow users to view the current settings in each room and over the complete house, toggle the lighting, and set a lighting schedule.

Executive Summary

As a Computer Engineering student, I'll be using all of the skills I have learned in the last 2 years in this program to complete this project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The sensor hub hardware will include a custom PCB with sensors and XBee wireless adapters and an Arduino inside a custom printed box. The Base station hardware will include an XBee wireless adapter connected to a Raspberry Pi. The database will store the sensor data from each room. The mobile device functionality will allow users to view the current settings in each room and over the complete house, toggle the lighting, and set a lighting schedule. I may end up collaborating with the greater Active House team later in the semester but I will not be working with any companies. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project.

Background

The problem solved by this project is that it provides a home monitoring solution so that users homes can be controlled and monitored remotely through an intuitive app. It will help people monitor their resource consumption as well which may help drive down consumption.

There are several other home monitoring solutions on the market now(Lien, Bai, & Lin, 2007), but they are usually integrated into either the thermostat or home security system. This is one of the first projects I've seen where water and power consumption are also monitored on a residential scale.

In the Computer Engineering Technology program I have learned about the following topics from the respective relevant courses:

- Electronic Devices and Circuits – Soldering experience
- Database with Java – Setting up database
- Electronic Circuits – Voltage Divider calculations
- Technical Workplace Writing – Creating write-ups and technical drawings
- Software Engineering – Creating project schedules

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:
Phase 1 Hardware build.

Phase 2 System integration.

Phase 3 Demonstration to future employers.

```
*Phase 1 Hardware build*
```

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16” x 6” x 2 7/8” (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

```
*Phase 2 System integration*
```

The system integration will be completed in the winter term.

Phase 3 Demonstration to future employers

This project will showcase the knowledge and skills that we have learned to potential employers.

The tables below provide rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

Labour Estimates	**Hrs**	**Notes**
Phase 1		
Writing proposal.	9	Tech identification quiz.
Creating project schedule. Initial project team meeting.	9	Proposal due.
Creating budget. Status Meeting.	9	Project Schedule due.
Acquiring components and writing progress report.	9	Budget due.
Mechanical assembly and writing progress report. Status Meeting.	9	Progress Report due (components acquired milestone).
PCB fabrication.	9	Progress Report due (Mechanical Assembly milestone).

Interface wiring, Placard design, Status Meeting.	9	PCB Due (power up milestone).
Preparing for demonstration.	9	Placard due.
Writing progress report and demonstrating project.	9	Progress Report due (Demonstrations at Open House Saturday, November 7, 2015 from 10 a.m. - 2 p.m.).
Editing build video.	9	Peer grading of demonstrations due.
Incorporation of feedback from demonstration and writing progress report. Status Meeting.	9	30 second build video due.
Practice presentations	9	Progress Report due.
1st round of Presentations, Collaborators present.	9	Presentation PowerPoint file due.
2nd round of Presentations	9	Build instructions up due.
Project videos, Status Meeting.	9	30 second script due.
Phase 1 Total	**135**	
Phase 2		
Meet with collaborators	9	Status Meeting
Initial integration.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Meet with collaborators	9	Status Meeting
Incorporation of feedback.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Prepare for demonstration.	9	Progress Report

Complete presentation.	9	Demonstration at Open House Saturday, April 9, 2016 10 a.m. to 2 p.m.
Complete final report. 1st round of Presentations.	9	Presentation PowerPoint file due.
Write video script. 2nd round of Presentations, delivery of project.	9	Final written report including final budget and record of expenditures, covering both this semester and the previous semester.
Project videos.	9	Video script due
Phase 2 Total	**135**	
Phase 3		
Interviews	TBD	
Phase 3 Total	**TBD**	
Material Estimates	**Cost**	**Supplier**
Phase 1		
Cannakit Raspberry Pi v3	$99.99	Amazon.ca
Arduino Uno SMD Rev3	$31.08	Digi-key
Xbee S1 Trace Antenna (x2)	$53.68	Digi-key
Xbee Shield for Arduino	$22.77	Digi-key
3.5mm audio jack	$2.09	Canada Robotix
Non-Invasive Current Sensor	$13.19	Canada Robotix
Adafruit Gas Sensor	$21.12	Digi-key
Adafruit Light Sensor	$8.40	Digi-key
Water Flow Sensor	$11.89	Canada Robotix
3 Pin Molex header	$0.39	Canada Robotix
12mm Buzzer	$1.99	Canada Robotix
Adafruit Temperature and Humidity Sensor	$19.70	Digi-key
Shipping	$6.95	
Tax	$37.22	

Phase 1 Total	**$330.46**
Phase 2	
Materials to improve functionality, fit, and finish of project.	N/A
Phase 2 Total	**TBD**
Phase 3	
Phase 3 Total	**TBD**

Concluding remarks

This proposal presents a plan for providing an IoT solution for a home monitoring system. This is an opportunity to integrate the knowledge and skills developed in the Computer Engineering program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects. I request approval of this project.

Abstract

This report contains information outlining the ActiveHouse home monitoring system project. The project aims to create a complete home enviromental monitoring system to measure temperature, humidity, light levels, water consumption, power consumption, smoke and gas detection, and the ability to toggle lights. There is an Android mobile application for displaying the data and making changes. The hardware consists of a Raspberry Pi, an Arduino, xBee wireless adapters, and a mirriad of sensors. The web service and database to store all the data is hosted on Munro.humber.ca. The total cost for the prototype is about $330. The project will take me about 8 months to complete but can be redone in much less time.

Table of Contents

List of Illustrations or Diagrams

Introduction

The ActiveHouse Project is a complete home monitoring system consisting of multiple Arduino based sensor hubs in each room of your home wirelessly connecting to a Raspberry Pi based base station which sends the data to the server where it can be read by the app. Each sensor hub is be capable of reading temperature, humidity, luminosity, carbon monoxide and other gas levels, water consumption, power consumption, and it would have the ability to toggle the built-in light. The power and water consumption sensors are detachable as they would not necessarily be used in each room, thus creating a more modular design. The app allows users to view the current settings in each room and over the complete house, toggle the lighting, and set a lighting schedule.

Project Description

The ActiveHouse project consists of two main pieces of hardware:

Raspberry Pi based base station with an Xbee wireless antenna connected to it.

Arduino based sensor hub which includes the following:

–Xbee Wireless Antenna

–Light Sensor

–Gas Sensor

–Current Sensor

–Water Flow Sensor

–Temperature Sensor

–Humidity Sensor

–Remote Controllable Light

–Buzzer

You can have as may sensor hubs as you want hooked up to one base station, they jut need to be within the 300 foot range for the XBees.

There are 5 main software components at work in the Active House system; the Arduino code running on the Sensor Hubs, the C++ code running on the Raspberry Pi base station, the PHP Web Service Running on Munro, the MySQL database on Munro, and the mobile Android app.

Sensor Hub Code

Arduino Specifications

The Arduino program communicates with the Base station over the XBee serial communication to send sensor data and toggle the light status.

Code Breakdown

The program is written to initialize all of the sensors. After that, it waits and listens on the XBee serial port for communication from the base station. When it receives a message from the base station that includes that sensor hub's unique ID, it replies with all of the sensor data.

Base Station Code

Raspberry Pi Specifications

The Raspberry Pi based Base Station is used to communicate with the Sensor Hubs over the XBee serial communication to gather all of the sensor data and send it to the web service running on Munro.

** Code Breakdown**

The code that is running on the Raspberry Pi is written in C++ using the ArduPi.cpp library which makes it easier to communicate using the Xbee. The service has a list of the registered sensor hubs, and goes in order, broadcasting each RoomID over the Xbee. When the corresponding sensor hub receives this message, it will reply with all of its sensor data. The Base Station code then parses that response and sends it back to the web service that is running on Munro.

Database Specifications

Database Type

For the ActiveHouse project I am using a MySQL database hosted on Munro. It can be administered through phpMyAdmin.

Database Tables

There are four tables that are used in the Firebase.

1. USERS
2. HOUSE
3. ROOM
4. ROOM_DATA

USERS is the table in which all users who sign up for the ActiveHouse application are stored. They are given a key as a unique identifier.

HOUSE is the table in which the names of houses and their data are stored. Each house is unique via an HOUSEID identification key within the table.

ROOM is the table in which the rooms in each house are stored. It contains the HOUSEID that it corresponds with, a unique ROOMID and the room name.

ROOM_DATA is the table where the sensor readings for each room are stored. It is identified by the unique ROOMID.

Work Breakdown

There is no further work that needs to be done to the database.

Mobile Application Specifications

Graphical User Interface Specifications

MainActivity: The login screen of the application is the first presented upon launching the application. The user may enter already existing credentials into EditText fields and login, or may choose to go to the sign up activity, both navigations via Buttons. There is a checkbox to save the entered username to local storage to be auto populated in future sessions.

RegisterActivity: The sign up screen of the application can be accessed from the login screen. It has EditText fields for gathering new user information including their HOUSEID. Without a HOUSEID, they cannot create a new account. Incorrect EditText entries will be communicated via Toast message asking for re-entry.

HomeActivity: This is the main screen of the app where the user can view the current settings of the user's house. There is a hamburger menu with links to the other pages including one for each room in the home. Users can logout from the menu bar.

RoomActivity: This page has the sensor data from an individual room displayed. The user can interact with the page by toggling the light or light schedule or setting the time on or off for the light. There is a hamburger menu with links to the other pages including one for each room in the home. Users can logout from the menu bar.

RoomViewActivity: The room view activity is a listview with all of the rooms in it. Users can select the room they wish to view and be taken to the RoomActivity page for that room.

Application Work Breakdown

There are a few small bugs that I need to patch, but the app is working and pretty much complete.

Web Specifications

Web Service Specifications

The web service is written in php and hosted on Munro. It can be accessed at munro.humber.ca/~n01046059/ActiveHous

Web Work Breakdown

The side of the web service that interacts with the mobile app is complete and does not require any changes. I do need to write the side of the web service that connects to the Raspberry Pi to send and receive sensor data. Additionally I need to write a program to run on the Raspberry Pi to send the data to the server.

Build Instructions

My Active House project is a complete home monitoring system. The idea is that you would have one Sensor Hub in each room of your home to collect data You would have one Raspberry Pi based base station which all of the sensor hubs send their data wirelessly to. The base station then sends the received sensor data to the database for use in my app.

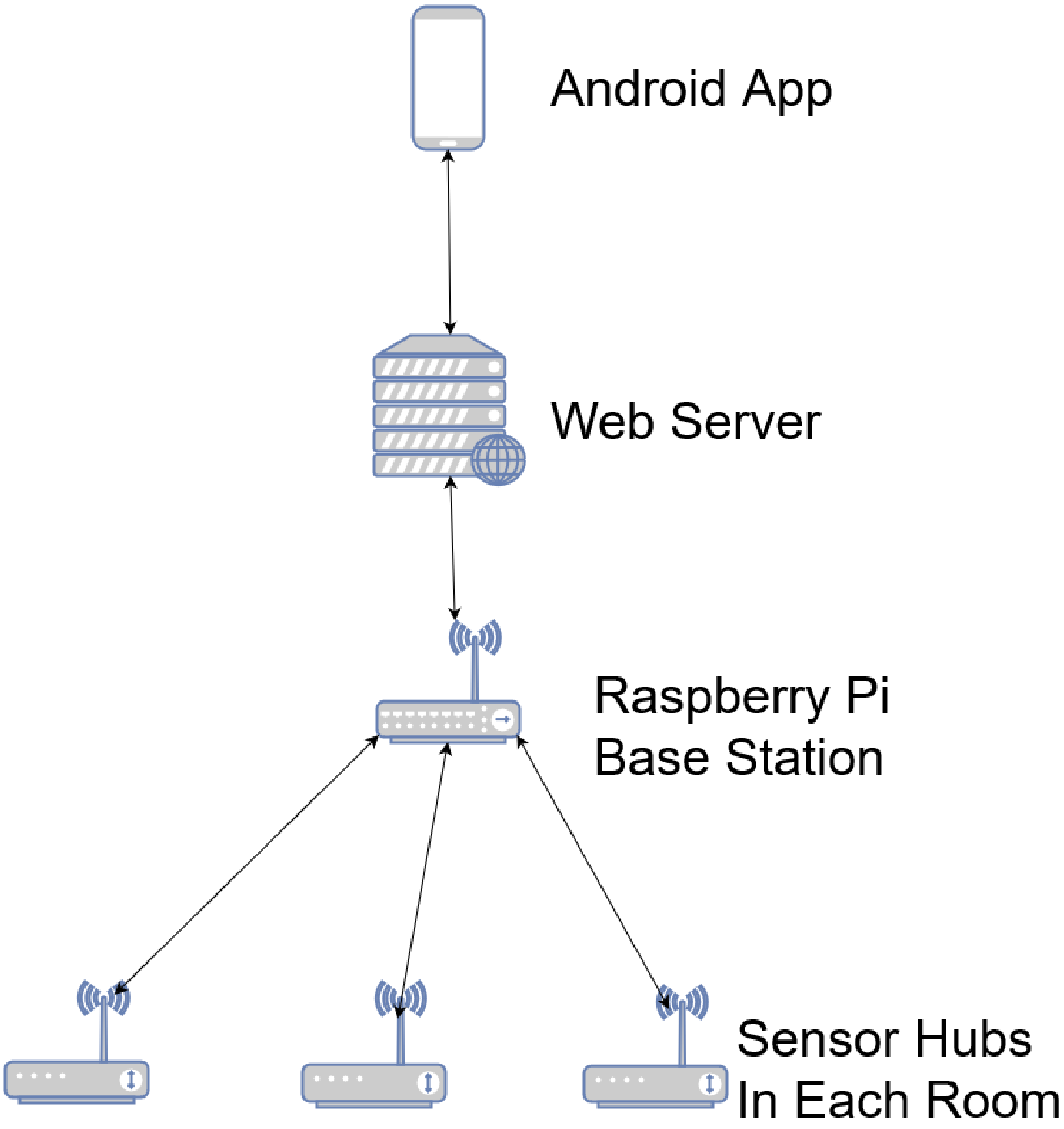

Figure 1:

Parts: Here is a list of all the parts and aproximate costs for one Raspberry Pi base station and one Sensor Hub:

Part	Quantity	Unit Cost	Total Cost	Supplier
Cannakit Raspberry Pi v3	1	$99.99	$99.99	Amazon.ca
Arduino Uno SMD Rev3	1	$31.08	$31.08	Digi-key
Xbee S1 Trace Antenna	2	$26.84	$53.68	Digi-key
Xbee Shield for Arudino	1	$22.77	$22.77	Digi-key
3.5mm audio jack	1	$2.09	$2.09	Canada Robotix
Non-Invasive Current Sensor (30A)	1	$13.19	$13.19	Canada Robotix
MiCS5524 (Gas sensor)	1	$21.12	$21.12	Digi-key
Adfruit i2c Light Sensor	1	$8.40	$8.40	Digi-key
Water Flow Sensor (1/2" Pipe)	1	$11.89	$11.89	Canada Robotix
3 Pin Molex header	1	$0.39	$0.39	Canada Robotix
12mm Buzzer	1	$1.99	$1.99	Canada Robotix
Adafruit SHT31-D Temperature / Humidity Sensor	1	$19.70	$19.70	Digi-key
Shipping			$6.95	Canada Robotix
Tax			$37.22	
Phase 1 Total			$330.46	

Figure 2:

This list does not include parts from the Humber electronics parts kit. The parts you will need from the kit are:

- 220ohm Resistor
- 33ohm Resistor
- 2x 10Kohm Resistors
- LED
- 50uf Capacitor
- Single Core Wire

You will also need to print out the Acrylic case which can be found here and the 2 PCBs which can be found here and here.

Time Commitment While it took me probably upwards of 30 hours to design the project and the circuit boards, put it all together, and code it, I'd expect it to only take around 5 hours to build the project again.

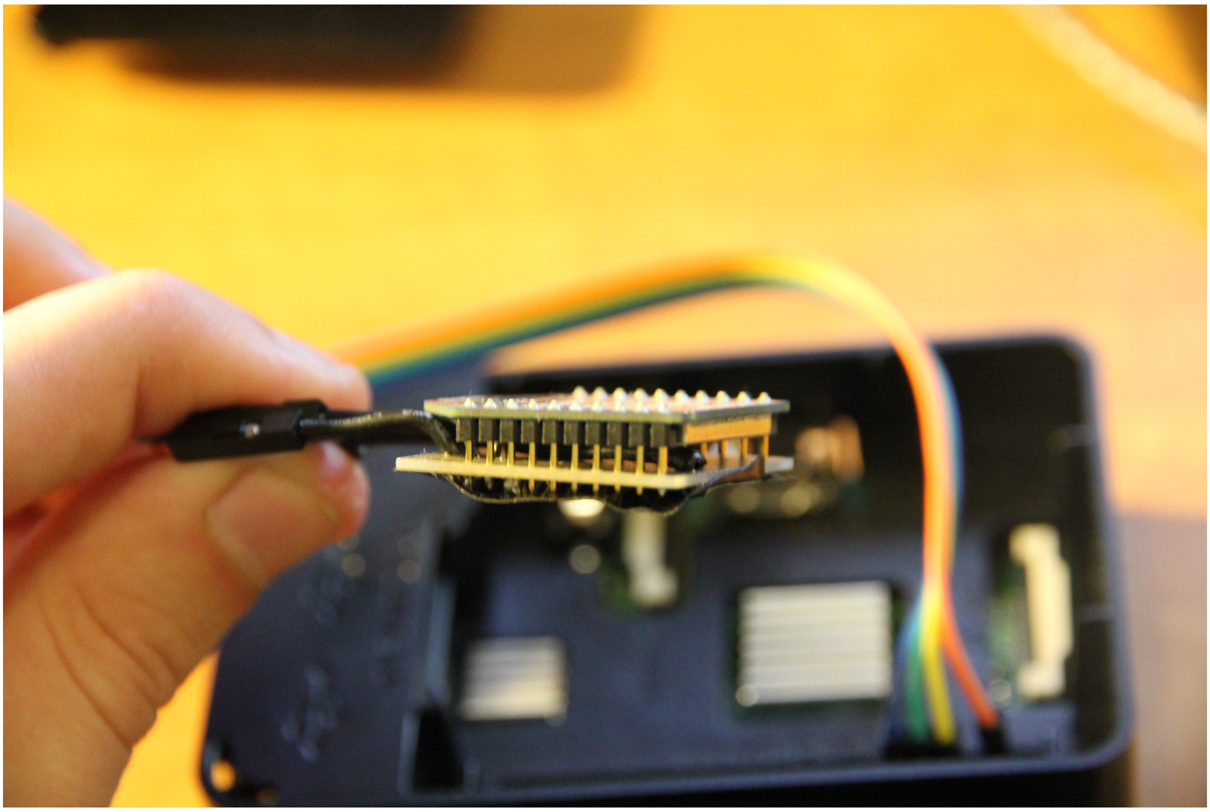

Figure 3:

PCB/Soldering The first board you will need to solder is the XBee Breakout board. You only need to solder the 3.3v, GND, RX and TX pins as seen in this diagram:

Then you will need to solder the Arduino Shield which includes all of the sensors and connections. You can view the board and schematic files for this board here which will explain where all of the components and connections go. Beware, some of the solder pads on the Arduin Shield are very small and you will need a very steady hand to solder them. Below are a collection of images which will help you understand where everything goes and how it is connected:

Mechanical Assembly First, you will need to unbox your Raspberry Pi and set it up. You can view my unboxing video here. After that, connect the wires from the Xbee breakout board to the appropriate GPIO pins.

Next you will need to build your acrylic case for your Arduino. Glue the all but the back side together and glue the ports for the current and liquid sensors onto the rear panel of the case.

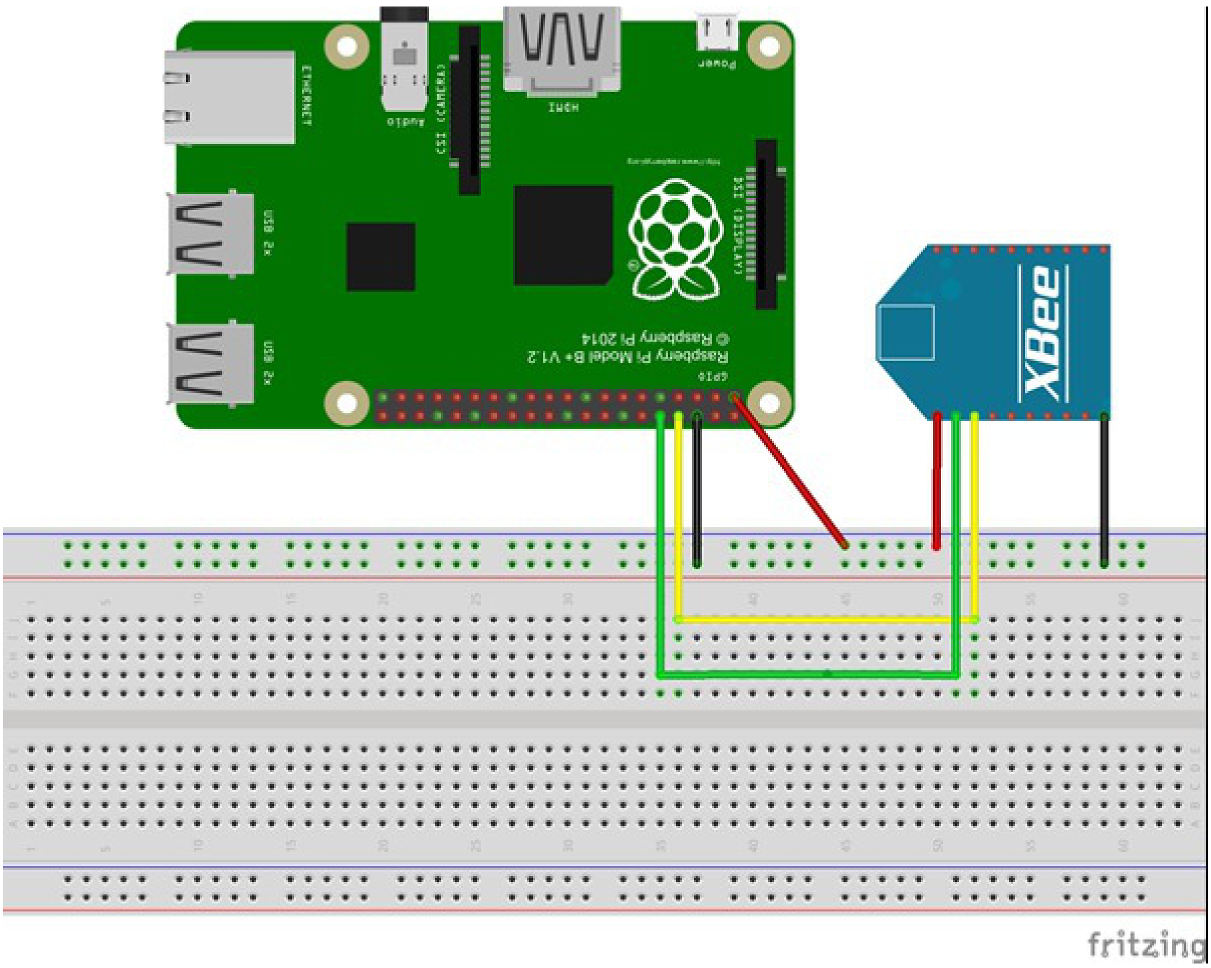

Figure 4:

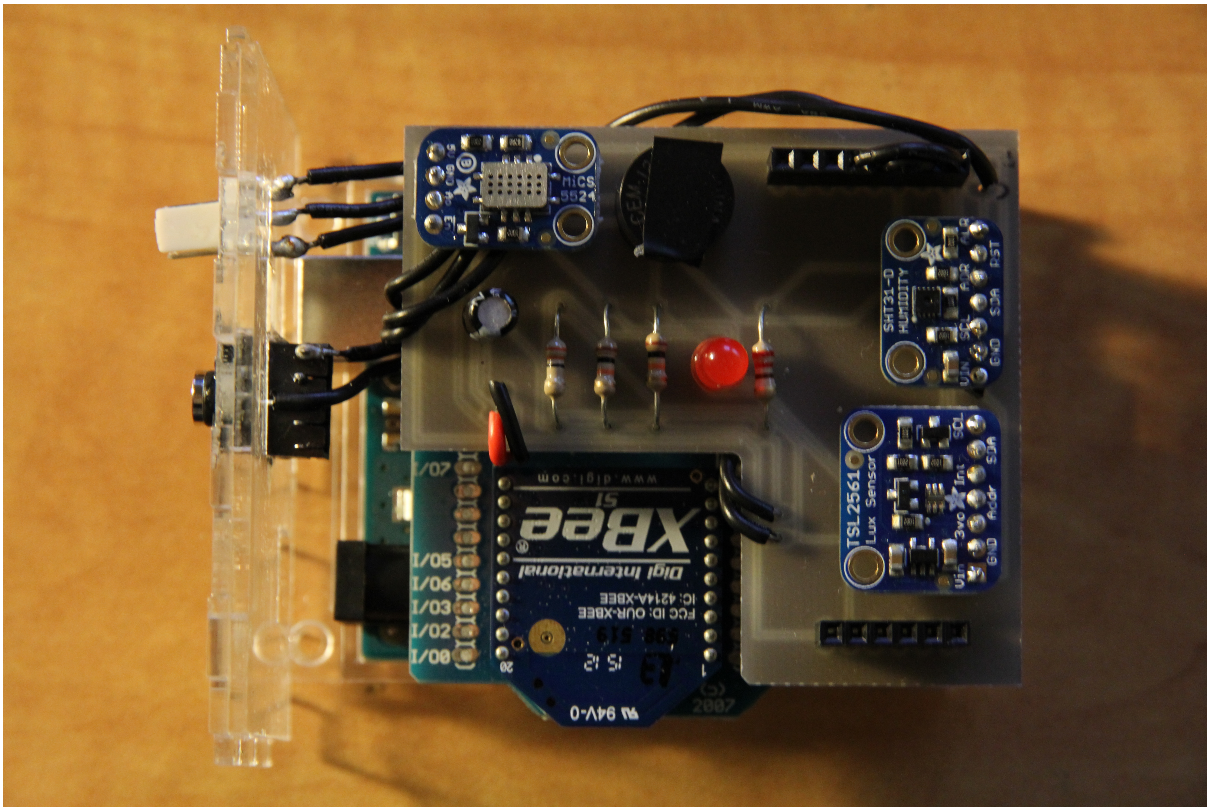

Figure 5:

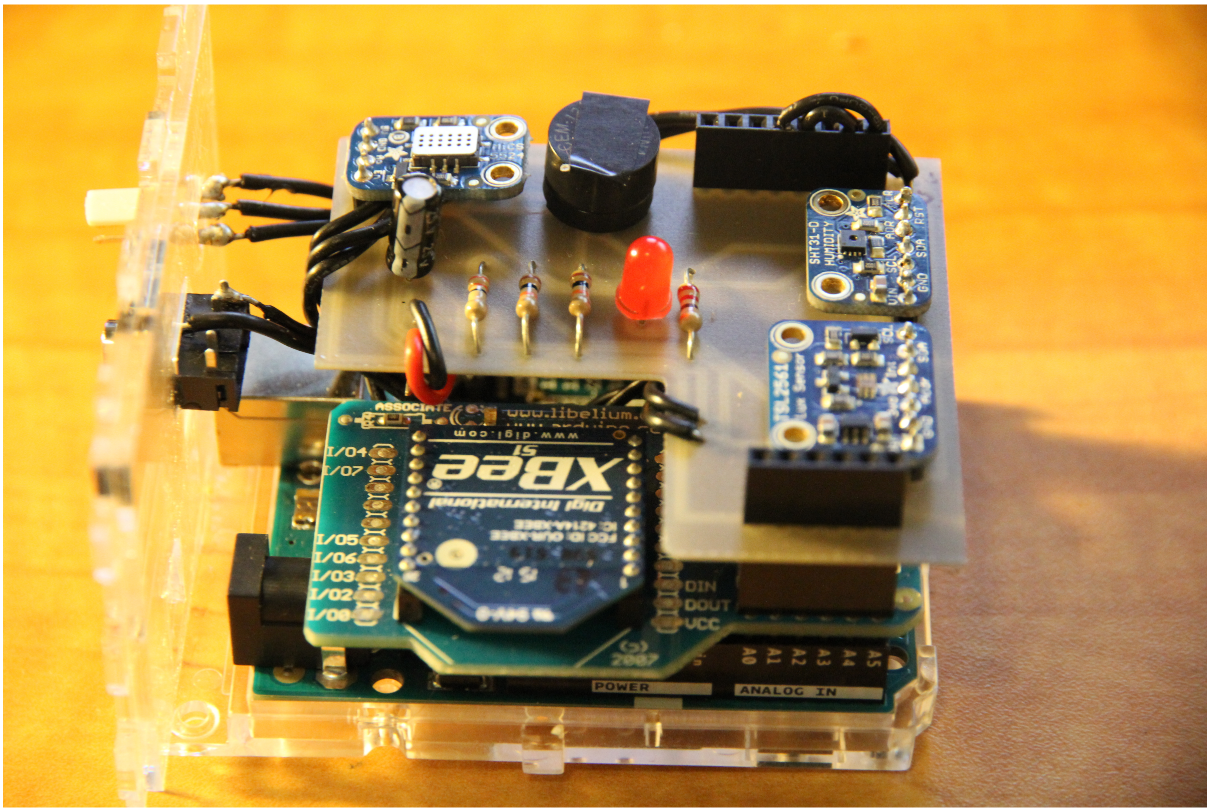

Figure 6:

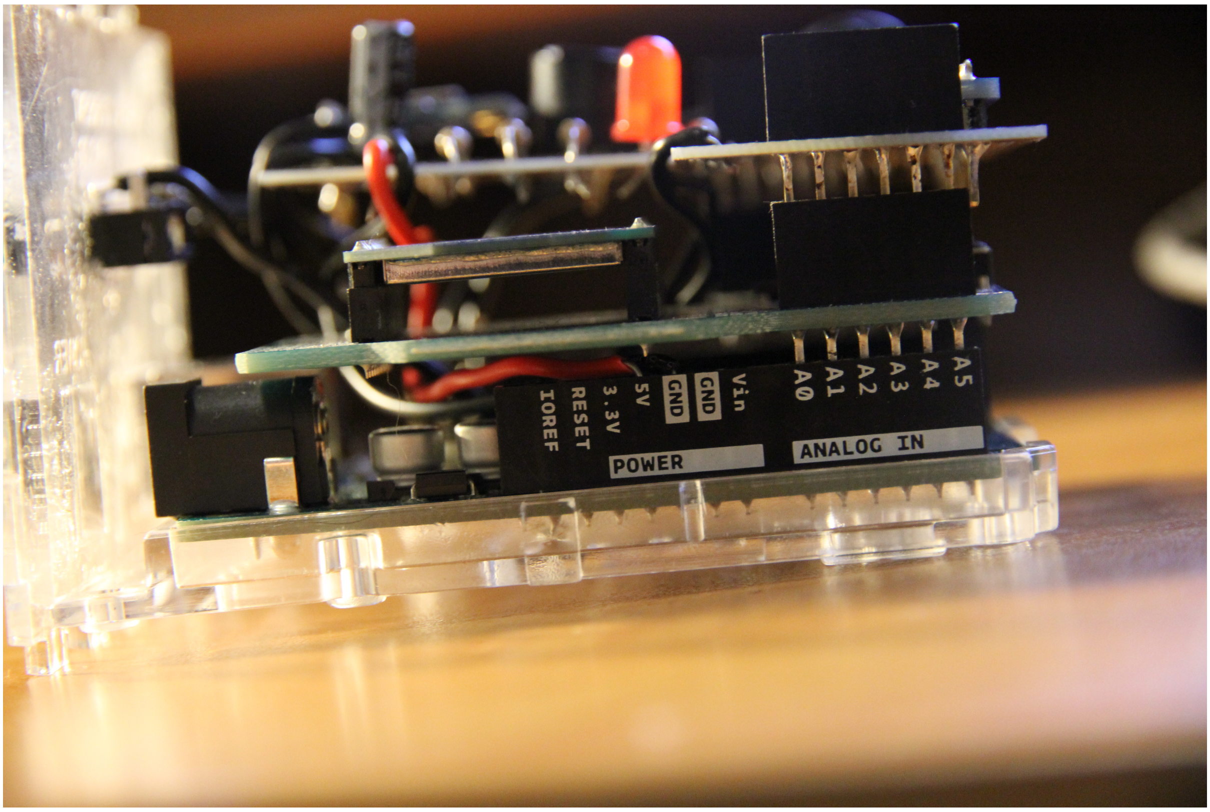

Figure 7:

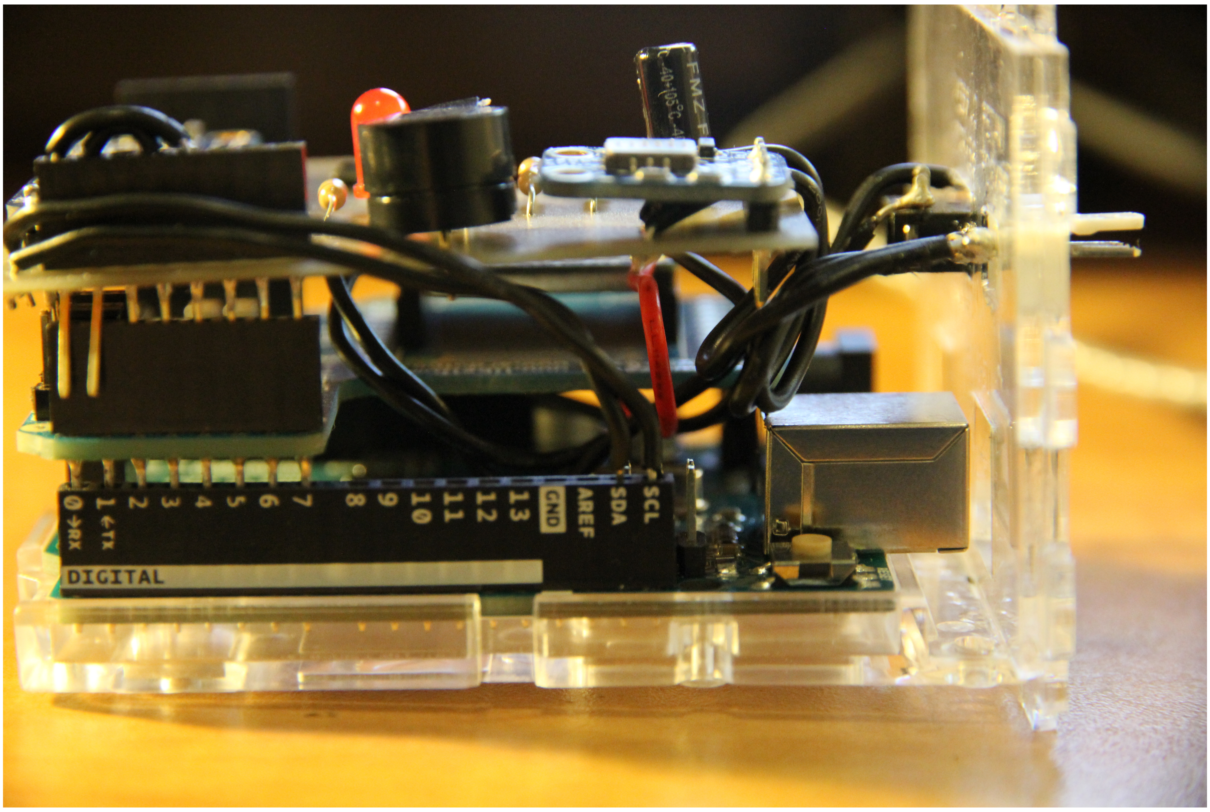

Figure 8:

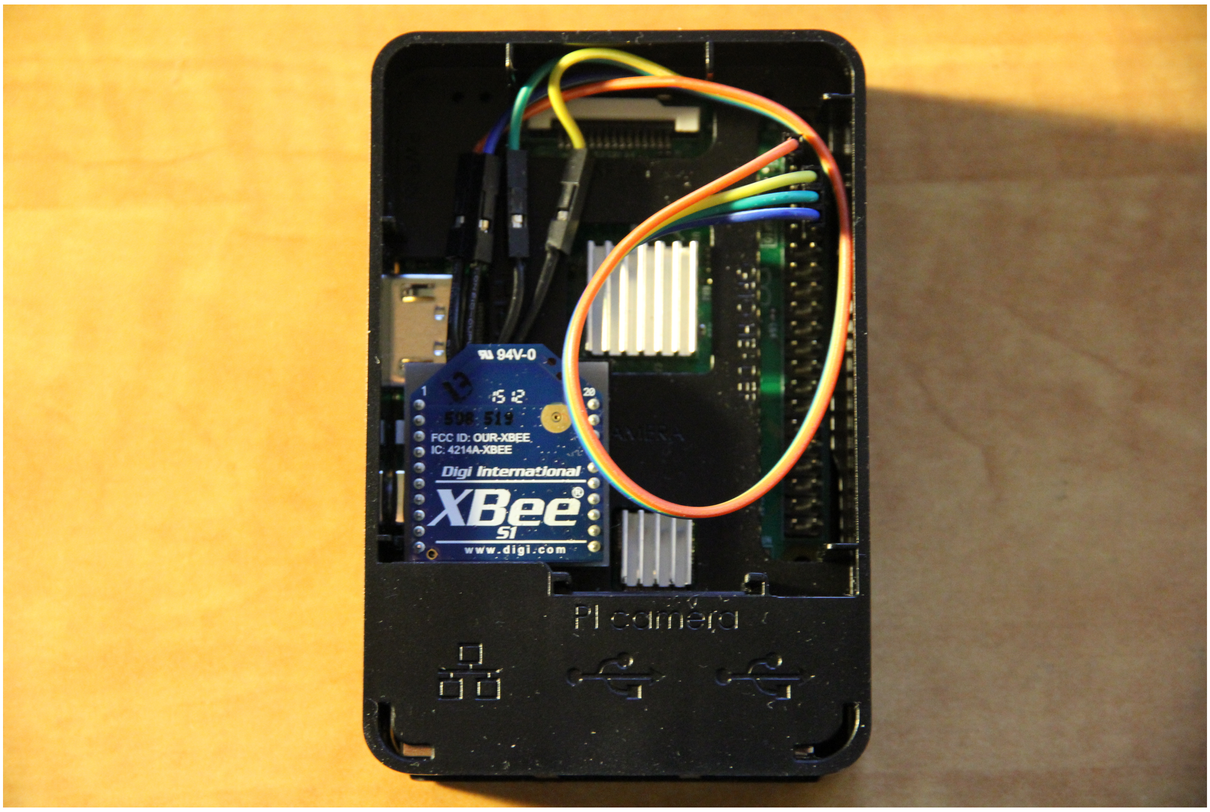

Figure 9:

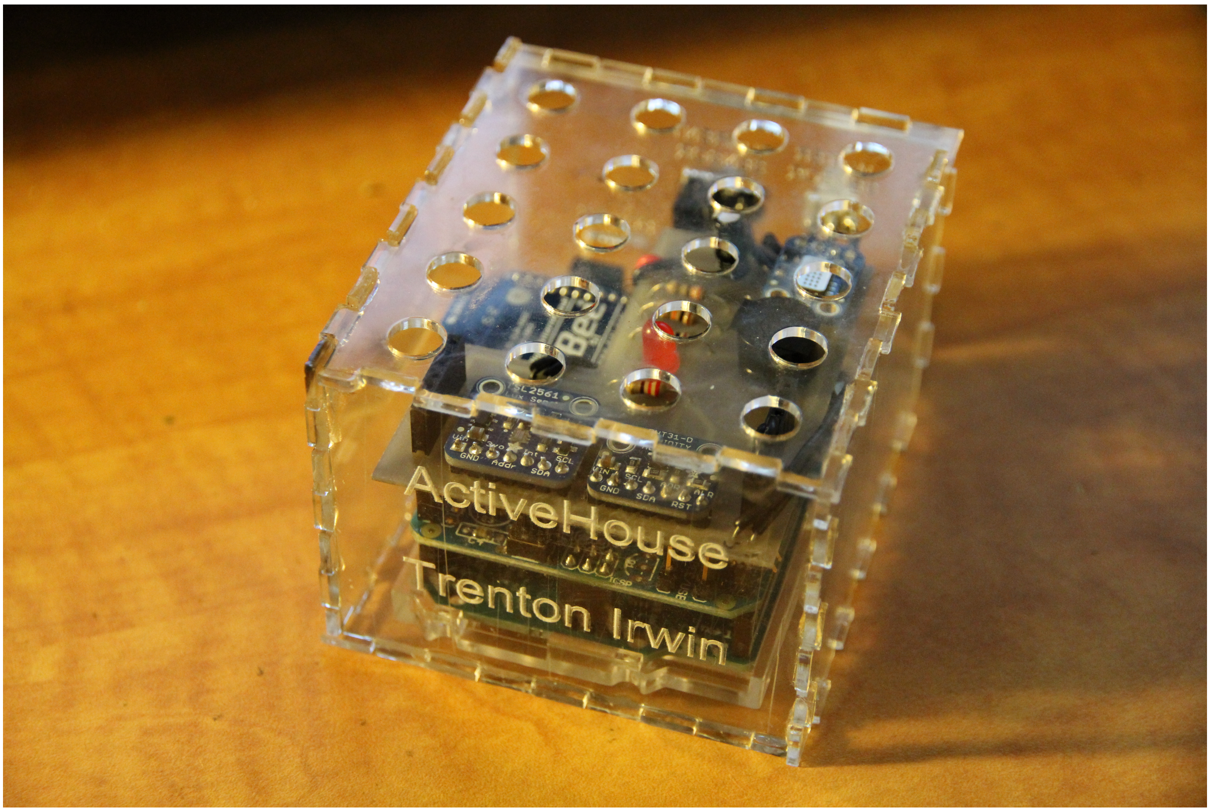

Figure 10:

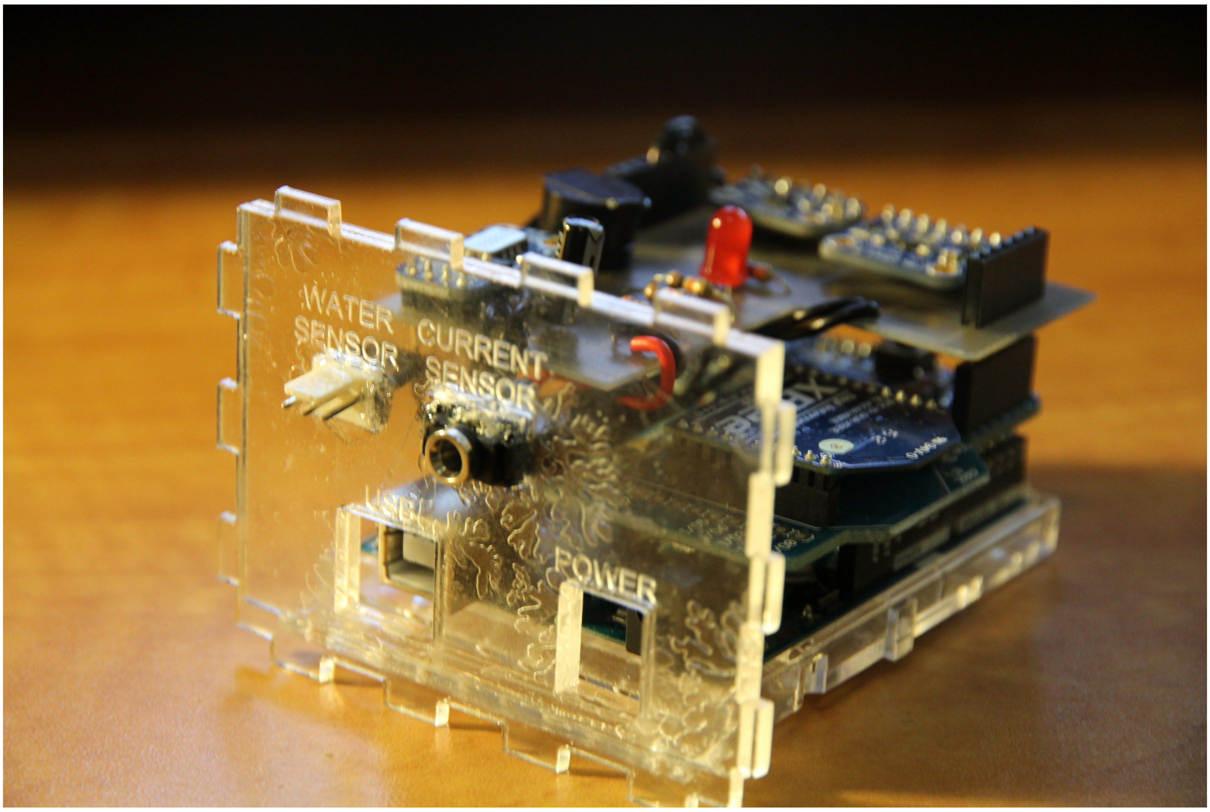

Figure 11:

Code You can view the Arduino Code here to read all of the sensor data and output it to Serial. You will also need to include the Libraries folder. Due to a limitation with the Arduino Uno and the Arduino XBee Shield, you cannot upload code to the Arduino while the XBee shield is plugged in, so it needs to be removed from the arduino board when uploading. Alternativly, you could flash the Arduino Uno's bootloader with the Duemilanove which supports uploading code over XBee and send the code to the Arduino wirelessly from the Raspberry Pi. You can read about how to flash the bootloader here

Power Up The output from the Arduino is below. You can also view my power up in my Build Video.

Testing In all of the testing I have done so far, everything works as expected. I've checked the read values from the temperature, humidity, light, and gas sensors and they all seem to be accurate. I have noth been able to find a 1/2" hose to connect to my liquid flow sensor so I've been unable to calibrate it, but tripping it by blowing air trough it works as expected. I've tested the current sensor with a few different appliances and its's read values seem to be close to the expected range, howeer I still need to completely calibrate it with an clampmeter. The buzzer and LED can both be toggled from within the

```
RMS Current: 0.26 amps
Apparent Power: : 31.53 watts
Gas level: 34
Light level: 24.00 lux
Temperature: 15.16*c
Humidity: 48.08%
Flow rate: 0.7L/min
Current Liquid Flowing: 12mL/Sec
Output Liquid Quantity: 12mL

RMS Current: 0.06 amps
Apparent Power: : 7.42 watts
Gas level: 33
Light level: 24.00 lux
Temperature: 15.16*c
Humidity: 47.98%
Flow rate: 8.6L/min
Current Liquid Flowing: 143mL/Sec
Output Liquid Quantity: 155mL

RMS Current: 0.01 amps
Apparent Power: : 1.75 watts
Gas level: 32
Light level: 23.00 lux
Temperature: 15.19*c
Humidity: 47.85%
Flow rate: 0.1L/min
Current Liquid Flowing: 3mL/Sec
Output Liquid Quantity: 158mL

RMS Current: 0.00 amps
Apparent Power: : 0.41 watts
Gas level: 31
Light level: 15.00 lux
Temperature: 15.21*c
Humidity: 47.79%
Flow rate: 0.0L/min
Current Liquid Flowing: 0mL/Sec
Output Liquid Quantity: 158mL
```

Figure 12:

code.

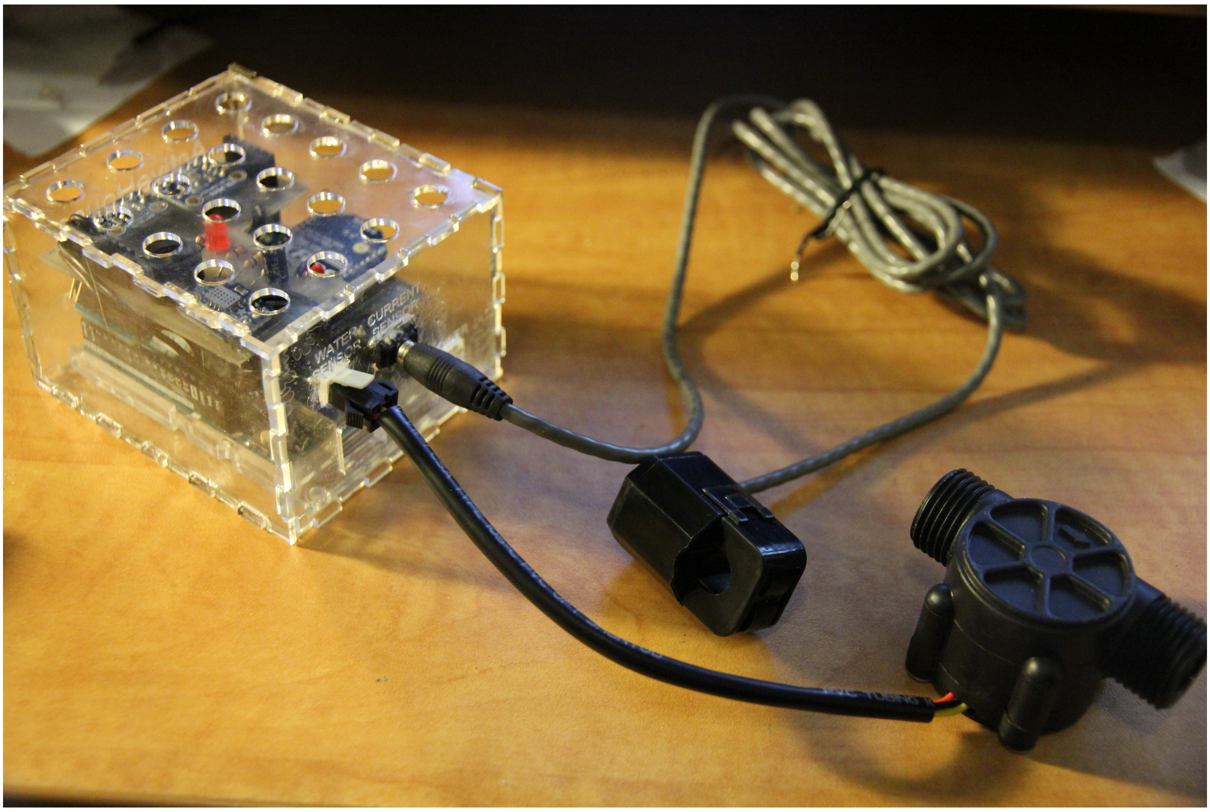

Figure 13:

Conclusions

The ActiveHouse project is nearly complete and will be within the next week. It was definately a learning experience, but I fell like it was a sucessful project. There are a few things I might do differently next time including adding a Sync button to both the base station and the sensor boxes to pair them automatically, however my current method of pairing works just fine. All of the components of the project are working flawlessly together as expected, there is just a bit of coding left to be done. The ActiveHouse project has great potential as a home monitoring system as it has many features included such as water and power consumption monitoring that aren't included in commercial solutions that are currently on the market. Unfortunately, the fact that it's build using mainly consumer hardware such as the Raspberry Pi and the Arduino makes it not financially viable to birng to market without compltetely rebuilding it all using cheaper microcontrollers. It was still a very good learning experience and the skills that I have learned on this project will keep me tinkering with electronics for years to come.

Appendicies

Bibliography

Lien, C. h., Bai, Y. w., & Lin, M. b. (2007). Remote-controllable power outlet system for home power management. IEEE Transactions on Consumer Electronics, 53(4), 1634–1641. https://doi.org/10.1109/TCE.2007.4429263