CALIFORNIA POLYTECHNIC STATE UNIVERSITY, SAN LUIS OBISPO

Smart Water Meter

Final Project Report

Instructor:	Thomas Katona
Class Section:	ENGR 459
Prepared By:	Jeffrey Hufford - jrhuffor@calpoly.edu
	Eric Adler - eadler@calpoly.edu
	Garrett Hall - gmhall@calpoly.edu
Submitted On:	6/5/2015

Disclaimer

Since this project is a result of a class assignment, it has been graded and accepted as fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is done at the risk of the user. These risks may include catastrophic failure of the device or infringement of patent or copyright laws. California Polytechnic State University at San Luis Obispo and its staff cannot be held liable for any use or misuse of the project.

Statement of Confidentiality

The complete senior project report was submitted to the project advisor and sponsor. The results of this project are of a confidential nature and will not be published at this time.

Table of Contents

Executive Summary	Page 5
Chapter 1: Introduction	Page 6
Chapter 2: Background	Page 9
Chapter 3: Design Development	Page 16
Chapter 4: Description of Final Design	Page 20
Chapter 5: Manufacturing and Design Verification (Testing)	Page 28
Chapter 6: Conclusions and Recommendations	Page 34
Acknowledgements	Page 35
Appendices	
Section A: Gantt Chart	Page 36
Section B: Microcontroller Comparison	Page 37
Section C: Transmission Throughput	Page 41
Section D: WiFi vs. Bluetooth LE Comparison	Page 43
Section E: Bill of Materials	Page 45
Section F: Vendor Supplied Component Schematics and Data Sheets	Page 46
Section G: Design Development	Page 49
Section H: Magnetometer Test Graphs	Page 54
Section I: Microcontroller Comparisons	Page 57

Executive Summary:

For three months, our team aggressively pursued the original idea of repurposing an ultrasonic flow meter in order to create a "smart water meter" that would be accurate, reliable, durable, easy to install and, most importantly, competitively priced in the marketplace. After repeatedly discovering holes in our original design and attempting to patch them, we ultimately decided that there were simply too many issues to create a functional, manufacturable product. We went back to the drawing board and looked at the big picture once again for a different solution. After much research and basic testing, we ultimately discovered that piggybacking off of the current municipal water meters will provide us with a far superior solution to our original problem of monitoring residential water usage.

The necessary components to create a working prototype and manufacturable product are much less expensive, widely available and have been proven to work. Articles have already been written by Do-It-Yourselfers on how to utilize the magnetic field created by a positive displacement water meter to monitor your residential water usage. In order to create a fully functional prototype, the only physical components we needed were two microcontrollers, a digital magnetometer, antenna and a battery. Additional, simple code was written to interpret the magnetic field signal and output quantifiable water usage data in a volumetric unit.

By piggybacking off of the municipal water meter, we can ensure incredibly high accuracy (99.9%), while only measuring a change in magnetic field strength, as opposed to the much more difficult task of measuring a physical flow rate. Municipalities in the United States most commonly use positive displacement water meters, all of which generate a magnetic field as a small magnet travels along a circular path. One magnetic cycle corresponds to one volumetric unit of water entering and leaving the flow meter. This amount of water per cycle changes with manufacturers, but can easily be determined by a simple calibration procedure upon installation.

This device is completely non-invasive, compatible with 90% of metered homes in California and requires maintenance only once per year. Our device resides inside the meter pit generally located on the sidewalk in front of most homes. It is a small device that clamps to the side of the meter, with an antenna directed towards the home. The device will require low power and be put into sleep mode when not in use, allowing the battery pack to last for about a year with no recharging necessary.

Chapter 1: Introduction

The Smart Water Meter project was created in response to a lack of individual household water information in the face of a global water shortage. Currently there is no simple, user-friendly interface that allows homeowners to monitor their water usage across intervals shorter than one month. Based on surveys we have conducted on 20 homes San Luis Obispo, we have determined that homeowners have a strong desire for this information. We will develop a fully integrated water measurement and reporting system that relays household water usage data to a server and is readily accessible to customers. Our team consists of four founders in total who committed to split the investment costs of the project and ownership equally amongst ourselves.

Customer Requirements

Household Water Usage Monitoring Device:

- *Invasiveness*

  The product must be non-invasive in order to provide an easier installation, avoid piping safety hazards, and bypass costly professional installation.

- *Power*

  The product must not require professional electrical installation. Renewable energy sources are preferred, but it is acceptable to run on a battery for one year without maintenance, or be located within 10 feet of a power outlet. Professionally installed electrical lines would be expensive to install, but would never require battery maintenance.

- *Accuracy and Reliability*

  The flow must be measured with greater than 1% ± 5% accuracy and be repeatable to a degree of 0.2% ± 5% in order to provide useful and reliable data to the consumer.

- *Microcontroller Compatibility*

  The product must be capable of communicating with a microcontroller. Digital communication is preferred, but analog devices may also be used provided the current draw is within an acceptable limit.

- *Ease of Calibration*
  The device must be able to be set up and calibrated by the customer. Automation (via software) should be utilized wherever possible.

Data Transfer:

- *Reliability*
  The product must reliably transmit data as to avoid data loss. Data stored on the server should match device readings completely. Fail-safes must be put in place in anticipation of possible connection issues that might arise.

- *Distance from Base*
  The device must be able to relay data to the customer's router, which may be up to 200 feet away and behind obstacles.

- *Security*
  The device must not compromise the security of the customer's home network. Customer information must be sent and received in a secure manner.

Power Supply:

- *Ease of Installation/Maintenance*
  Power installation must meet the total customer installation time requirement. A user-friendly guide must walk customers through installation procedures and account for all varieties of installation types. Batteries must be easily replaceable, and take less than 10 minutes to replace. If a rechargeable battery is to be used, the charging apparatus must be safe and user-friendly.

- *Low maintenance*
  The customer does not want to be bothered with replacing components or troubleshooting power issues. The power supply should be installed once and not need any further attention during the 1 year $\pm$ 1 months expected lifespan.

Webapp/Web Server:

- *Usage Data and Graphs*

  Customers need to be able to see yearly, monthly, weekly, and hourly usage data. This data must be supplemented with graphs and charts that show usage over time, and comparisons to previous time periods. Other desired comparisons include -but are not limited to- house size, number of family members, national and regional averages.

- *Accessible from any location.*

  The customer must be able to see their water usage from their phone, tablet, or laptop whether they are at home, out running an errand, or on vacation.

- *Notification functionality*

  Customers must be able to set up text and email notifications if desired. Desired notification types include -but are not limited to- specific usage thresholds, water usage that is deemed 'unusual' (abnormal spikes, large quantities at night, etc), and daily/weekly/monthly updates.

- *Usage trends*

  The webapp must be able to analyze customer usage trends and alert customers to abnormalities via notification.

Chapter 2: Background

Water is becoming increasingly scarce in the world, particularly in California. We are currently in our third consecutive year of a drought, and with reservoirs drying up, the forecast is not looking promising for California's water supply.[(1)] Although residential water usage is not the main consumer of water in California, it still accounts for 20% of water use and has a significant impact on the state's reserves.[(2)] Currently, there is no effective and simple solution to monitor how much water residents are using in real time, or even daily, without stepping outside, walking over to your water meter, recording readings and compiling data.

There are several products on the market that monitor water use in homes. Most however, target specific appliances or are designed to monitor for leak detection and operate as emergency shut-off valves. There is one product that is currently in the prototyping and pre-order phase which monitors water usage in real-time and relays data over a wifi network to an app on your phone; this company is our most direct competitor. Below is more information regarding the most relevant competing products.

Multiple companies have products on the market that target leaks and employ an emergency shut-off valve. One company, DynaQuip, created a product called the WaterCop that accomplishes this goal. The WaterCop utilizes a central module attached inline to your homes water main that functions as a Wi-Fi unit and the emergency shut-off valve; this unit draws power from a wall outlet. Battery powered sensors are then placed in locations around the home that are susceptible to leaks, such as the bathroom, laundry room, kitchen, basement, etc. These sensors will send a signal to the central shut-off valve if they detect water, reducing the amount of water damage that your home incurs. The total cost for just the base model of this device, excluding the sensors is $370.[(3)]

Another device specifically targets water usage related to sprinkler and irrigation systems.[4] Hydrawise "intelligently adjusts your sprinklers watering schedule to suit your local weather conditions". The unit replaces your current sprinkler controller, relays information over your wifi connection to an app on your phone and allows you to control the timing and amount

[1]California State Government Drought Information: http://ca.gov/drought/
[2]Wikipedia - Water Usage in California: http://en.wikipedia.org/wiki/Water_in_California
[3]Leak Detection & Emergency Shut-Off Valve: http://www.watercop.com/
[4] HydraWise sprinkler and irrigation monitoring https://hydrawise.com/irrigation-features/

of water used as well as which region of your lawn you wish to water. With the flow meter included, this system costs upwards of $400.

There is currently a very simple, easy to use and inexpensive water monitor device on the market that has received praise by the press and customers. The device is basically a timer that sits on the floor of your shower and begins timing when it first detects water. A display illuminates a green light for the first minute that water is detected, informing the user that they are being environmentally friendly and using small amounts of water. As the device continues to detect water, the LED switches to orange and then finally to red to indicate that you are using too much water. This device is a real-time water monitoring tool in its most basic form and lacks many features, but is a bargain to the consumer at only $13.[5]

These devices have aspects of their design that are similar in nature to ours, but only monitor certain areas of your residential plumbing and do not record overall residential water usage. There are however two companies that have the potential to be significant competitors.

The first competitor is a company named Driblet which was launched at the beginning of 2014.[6] They have designed a self-powered in-line water meter that connects to your wifi and relays real-time water usage information to an app on your smartphone; this device has not yet been brought to market, but is available on pre-order with no specified price. This device solves the same problem as ours, but the main difference is that the water meter is in-line and designed to be used with PVC piping. This would require homeowners to hire a plumber to cut their main inlet pipe and install the device. Through multiple conversations with plumbers, we have determined an average price of roughly $200 to cut a pipe and insert a new section. Our device will be non-invasive and will not require professional skills to install--a portion of the design that we believe has a significant impact in the marketplace.

Second, a company named Water Hero[7] is measuring the magnetic field from the municipal water meter to monitor home water usage and has incorporated an emergency shut-off valve into the design. Their product is by far the most similar in design to ours. However, since their product must be plugged into a wall outlet and is not designed for outdoor use, it can only be used in homes that have the municipal water meter located in a basement; a rare occurrence in California.

---

[5] Water Pebble Shower Monitor: http://www.waterpebble.com/
[6] Driblet Smart Water Meter: http://driblet.io
[7] Water Hero: http://www.waterhero.us/

Our devices will be non-invasive by piggybacking off of the municipal positive displacement water meter. It will utilize the changing magnetic field created by an oscillating magnetic inside of the meter. One complete revolution of the magnet corresponds to a finite amount of water passing through the meter. This change in magnetic field can be detected by a Hall effect sensor, which will then be connected to a microcontroller, where the signal will be analyzed to note every time a complete cycle has occurred.

There are currently no city or state laws that prevent homeowners or residence from recording their own water usage data. However, a device that measures a residents data remotely has never before been implemented and therefore it cannot be guaranteed that no legal issues will arise farther down the road.

Nearly all municipalities use positive displacement water meters to measure residential usage, meaning that a magnetic field will be present for our device to detect. There are however multiple meter manufacturers that the municipalities source from, resulting in one magnetic field cycle corresponding to a different amount of water, depending on which meter is used. Because of this, a calibration procedure will need to be performed during installation. Fortunately, the user will not need to know any information about their water meter, a very basic calibration can be performed via the applications software and the user filling up a specific amount of water, no physical calibration will be necessary.

The American Water Works Association (AWWA) holds the standards in the industry for flow meter qualifications. Certain aspects of our product, such as accuracy, will be based upon the accuracy of the specific water meter installed by the city. Further testing procedures are described in the appendices.

The WiFi module and Bluetooth shield will be bound by the standards and codes of IEEE section 802.11 and section 802.15, respectively.

Objectives:

As previously stated, developed a low-cost, non-invasive water usage monitoring system that can be installed in residential homes. Since 70% of California homes are metered by the city [8], a 90% compatibility goal has been set within these metered homes. This yields an overall compatibility of 63% of California homes. Key objectives for the device are listed below, followed by a more detailed and exhaustive specification list.

Low Cost: A key requirement comes from the fact that our customers will have enough disposable income to afford this product. We surveyed homeowners in San Luis Obispo and discovered that an ideal price for the device would be around $100, while many homeowners said that they would be willing to pay up to $150. Currently, $100 per unit is our goal at a 1000 units per year, but this value may change as research, prototyping and development continues. We believe this initial estimate is a reasonable balance of where cost to produce and customer demand could be reached. A detailed cost analysis is in Appendix Section E.

Invasiveness: The Smart Water Meter will reside in the meter pit, where the municipal water meter is located. A small wire with a Hall effect sensor on the end (< 1 cm^2 in size) will be attached to the side of the meter. The microcontroller and battery will be attached to the wall of the meter pit, ensuring that the meter counter is not unobstructed. An antenna will fed along the wall of the pit and stuck to the shelf where the concrete lid rests, ensuring that it does not reach about the lid. This product will be easily implemented without tampering with existing pipe systems.

Accuracy: Assuming our device will function without electrical or software errors, the accuracy of our device will be then directly related to the accuracy of the municipal water meter. Positive displacement flow meters provide high accuracy (±0.1%)[9] resulting in our device having the ability to attain 99.9% accurate water usage data. This high accuracy will allow the user to detect small leaks, enabling them to take the necessary action to fix the leak, saving water and money.

Ease of Installation/Calibration: One of the most important aspect of this product is ensuring that the user has the ability to easily install the device without professional assistance. The end goal is to create a product than can be setup and operated by the average homeowner. Calibration will be necessary and likely require the user to run a calibration mode on the application, while filling up a container with a specified volume (as simple as filling up a gallon jug). While the user is filling the container, the Hall effect sensor and microcontroller will be

[8] Percent of homes metered: http://pacinst.org/wp-content/uploads/sites/21/2014/09/pacinst-metering-in-california.pdf
[9]Flow & Level Measurement: http://www.omega.com/literature/transactions/volume4/t9904-08-mech.html

recording magnetic cycles. After the water is shut off, the device will know how much water corresponds to one magnetic cycle, and the device will be successfully calibrated.

Ease of Use: The software will be the face of the product and must be easy to use as well aesthetically pleasing. A simple, easily navigable software interface will drastically increase the customer's experience. The customer will be able to calibrate their flowmeter through this software on their mobile device. Real time data will be relayed and recorded. The user will be able to see their usage trends and diagnose any unexpected results. Leak detection features will alert the user if an abnormal amount of water is being used or if the meter is steadily running over an extended period of time. A warning system will be incorporated into the app, informing the user that they are getting within a certain range of the next tier of water unit pricing. Also, the user can choose if they want to be alerted after a set number of gallons have been used. This will be useful for customers who are limited to a certain amount of water a month. We will be working closely with James Fazio, our CSC and app developer, to ensure that the mechanical, electrical, and software sides of the project are integrating successfully.

Table of Requirements

No.	Parameter Description	Requirement/Target	Unit	Tolerance	Risk	Compliance
Microcontroller						
1.1	Input Voltage	5	V	± 5%	H	A
1.2	Current Draw	5 mA	mA	± 10%	H	A T
1.3	PCB Construction	FR-4 A	-	TBD	L	A
1.4	Solder	Lead-Free	-	TBD	L	A
1.5	Memory	32	KBytes	N/A	M	A T
1.6	Processor Speed	16	MHz	N/A	L	A
1.7	Dimensions (LxWxH)	4x4x2	Inches	TBD	L	I
1.8	Case Material	ABS	-	TBD	L	T
1.9	Construction	Water Tight / Potted	-	IP57	H	A T S I
Hall Effect Sensor						
2.1	Sensitivity	0.1	µT	± 0.01	M	A T
2.2	Magnetic Range	0.4 - 5	Gauss	± 0.2	M	A T
2.3	Dimensions (LxWxH)	0.5x0.5x0.5	Inches	TBD	L	I
2.4	Material	Plastic	-	TBD	L	T
2.5	Construction	Water Tight / Potted	-	IP57	H	A T S I
Battery						
3.1	Dimensions (LxWxH)	5x10x2 in	Inches	TBD	L	I
3.2	Capacity	25	Amp-Hour	TBD	H	T
3.3	Voltage	6	V	± 1 V	M	A T
3.4	Type	Lithium Ion	-	N/A	L	I
3.5	Construction	Sealed	-	IP57	H	A T S I
Measurement						
4.1	Repeatability	0.50%	-	5%	H	T
4.2	Accuracy	0.10%	-	5%	H	T
Operational						
5.1	Safety	No Risk of ESD	-	0	M	A T I
5.2	Temperature	-20 - 60	℃	±1	H	T
5.3	Humidity (RH)	0% to 100%	-	0	H	T
5.4	Installation	15	min	± 15	M	A T

Cost						
6.1	***Cost/unit at high volume****	***$40***	***USD***	***±$5***	***H***	***A***
6.2	***Price/unit at high volume****	***$100***	***USD***	***±$5***	***H***	***A***
Website						
7.1	***Uptime***	***99.90%***	***-***	***1%***	***H***	***A I***
Other						
8.1	***Lifetime***	***5 Years***	***-***	***± 6 Months***	***M***	***A T***

See Appendix for Specification Details

Chapter 3: Design Development

Thorough conceptual design selection and analysis was performed over that last four months and highlighted in our Conceptual Design Report. We performed basic "proof-of-concept" analysis on our current design in order to prove that the main hurdle of measuring the changing magnetic field in a positive displacement water meter has not only been done before by others, but can be easily repeated on all types of positive displacement water meters.

Measuring Flow:

There are a variety of flow measurement devices available on the market. The majority of these measurement techniques are in-line, requiring the user to cut their pipe to retrofit the device. We analyzed both in-line flow measurement as well as non-invasive options. Our original design utilized a non-invasive ultrasonic flow meter, but after months of testing, we discovered that this technology was simply not designed to be used at such a low flow rates and on such a small pipe. After comparing and contrasting a multitude of other meters, we discovered that the best option was to simply use the municipal flow meter and measure a magnetic field created as a byproduct of positive displacement meters--a much simpler, cheaper, and liability free solution. See Appendix G for detailed consideration breakdown.

Sensor:

In order to detect changes in the magnetic field generated by the positive displacement meter, several types of Hall Effect sensors were considered: threshold, linear, and magnetometer.

Threshold:

Threshold sensors work by producing a voltage once a certain magnetic field strength has been detected. This type of analog sensor would not work well for our project because we want it to work for a wide range of positive displacement meters. Threshold sensors must be set to activate at a fixed value and so a sensor that works for one meter may not work with another and vice versa.

Linear:

Linear Hall Effect sensors are analog sensors that output a voltage that is proportional to the detected magnetic field. We originally decided that this type of sensor would be ideal for prototyping as it would allow us to create an algorithm and/or circuit that would normalize sensor output and get reliable data across many types of meters, all while being fairly simple to implement.

The most important requirements in choosing our sensor were power consumption and sensitivity. We found that most linear Hall Effect sensors require 1.3-10mA of current and are capable of running on a 5 Volt supply, well within the operating range of the Uno. We ended up choosing the DRV5053EA chip because it consumes 2.7 mA (on the low-end of the range), and because its sensitivity rating was the best that we could find.

Initially we chose the DRV5053EA linear Hall Effect sensor. As per its datasheet, the 5053EA has a typical sensitivity of 45mV/mT, which is the greatest sensitivity we could find among linear analog sensors. After some initial tests with the sensor and a positive displacement meter, we determined that an amplification circuit would need to be constructed in order for the Arduino's Analog-to-Digital converter to better read the sensor output.

Amplifying the circuit caused the measured resolution to increase seven-fold. This increase in resolution was important as otherwise it would have been difficult to differentiate peaks from the operating voltage. While this was a step in the right direction, there were several negatives: the resolution was still on the low side for detecting peaks, and the addition of an amplification circuit meant that there would be more current required to operate the circuit, even during 'sleep' periods.

Magnetometer:

A third type of low-powered, digital sensor exists and is called a 'magnetometer'. This type of device seems to be fairly new (2012), and somewhat of a niche product, as we were only able to find one of its kind. These devices have been in smartphones for a while, providing data for the standard 'compass' app, but are not available to consumers to experiment with. One such sensor, the AK8936, is used in the iPhone 5 and has a sensitivity range of 0.15 to 0.6 μT[10]. The MAG3110 magnetometer, manufactured by Freescale Semiconductor, has a sensitivity of 0.1 μT and is capable of running at 2 μA in standby and as low as 10 μA in active mode. The power draw and sensitivity of this device blow analog sensors out of the water and it is available as a breakout-board that hobbyists can experiment with.

We did not select this chip at first because we had not tested it yet and we were concerned that it would be "too good to be true". However, we eventually received a prototyping board and were able to verify it's sensitivity claims.

Microcontroller:

For more information regarding our microcontroller selection, please see Appendix Section B.

[10] http://www.akm.com/akm/en/file/datasheet/AK8963.pdf

Data Relay:

In comparison, the latest 802.11n WiFi standard has a similar range and 100x the throughput, but peak current consumption can reach into the hundreds of milliamps when transmitting. Because this product will be sending miniscule amounts of data (See Appendix C.), such a high throughput is unnecessary. Because the ranges of these technologies are comparable, Bluetooth LE was selected for it's low power consumption.

Other RF technologies were considered, including the Moteino R4 which utilizes the 915 MHz band. [11] While the Moteino has excellent range performance and comparable power performance, it is a lesser-used protocol and does not benefit from a huge user base and developer community. Bluetooth LE is a set-in-stone standard and thus any Bluetooth LE enabled device could communicate with our device. Lastly, RF protocols are being updated all the time. By choosing a long-running standard we will maximize our device's chances of being backwards-compatible with future devices and be able to adapt to new Bluetooth standards with minimal effort.

Web Server: *(Backend)*

For the web server, **Apache 2.2** was chosen. This is because it is the most widely used server on the net and consequently has the most support for various web frameworks and databases. This support is is crucial as it will integrate seamlessly with the database and web framework selections. It will also provide us with the most hosting options when it becomes time to migrate to a production server.

Connection Broker: *(Backend)*

During server design and development, it was discovered that a 'broker' would be required to handle client-server interactions. For this we chose **Redis**: an open-source, fully-networked data structure server. Redis is used by a number of web-based companies such as Twitter, Github, and Craigslist, and is supported by cloud hosting services like Amazon Web Services.[12]

Task Distribution: *(Backend)*

Because Django does not (yet) support asynchronous tasks or take advantage of Python 3.4's new asynch library, a backend task handler must be implemented in conjunction with

[11] https://lowpowerlab.com/shop/index.php?_route_=moteino-r4
[12] http://redis.io/topics/whos-using-redis

Django. For this, we have chosen **Celery**, a Django-compatible Python module that efficiently handles task distribution.

Task API Routing: *(Backend)*

Celery does not natively implement an HTTP API and so **Flower** was installed in conjunction with Celery. Flower allows Celery to accept and process HTTP POST requests, which is what the base Arduino will be sending via Ethernet.

Database: *(Backend)*

MySQL was selected for the database. Like Apache, it is one of the most popular databases and has a lot of online support and documentation. It was also integrate seamlessly with the chosen web framework. PostgreSQL was ruled out due to its inferior performance and because the additional features it offers are not required for this project. Similarly, SQLite was ruled out for its relatively poor write performance. All other database configurations were ruled out for not being compatible with the chosen web framework.

Web Framework: *(Frontend)*

The web framework chosen was Django, which is written in Python. This framework was chosen because it has the biggest collection of available libraries and is written in a language the developer is familiar with. This will cut down on development time significantly. Performance-wise, the differences between the frameworks were negligible in comparison to the benefits gained by decreased development time. The framework is also a driving force behind the server and database selections.

Chapter 4: Description of Final Design

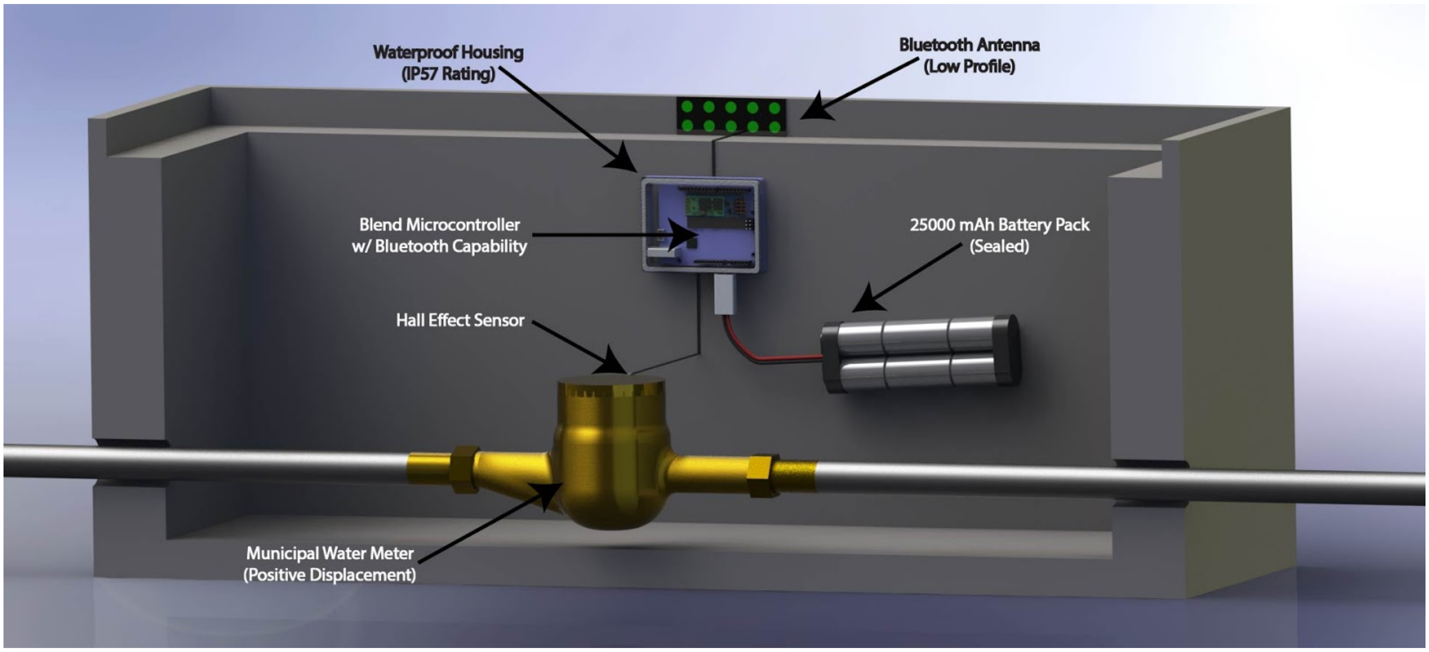


Figure 1: Conceptual layout of components incorporated into our final design.

Final Component List:

After thoroughly analyzing our design goal and researching current components on the market, we arrived at a concept prototype that incorporates these components, satisfies our requirements and performs optimally for our application. Our final design concept will utilize the following electrical components:

Meter Unit:

- **Microcontroller:** Arduino Uno
- **Magnetometer Sensor:** Freescale Semiconductor MAG3110
- **Antenna:** Rufa 2.4 GHz 2.5dB Bluetooth Antenna
- **Housing:** Custom 3D Printed (ABS Plastic)
- **Power Supply:** 27.6 Amp-hour Ni-MH Rechargeable Battery Pack
- **Data Relay:** RedBearLabs *BLE Mini* (Bluetooth Low-Energy)

<u>Base Unit:</u>

- **Microcontroller:** Arduino Uno
- **Connectivity In:** RedBearLabs *BLE Mini* (Bluetooth Low-Energy)
- **Connectivity Out:** Ethernet Cable to Home Router
- **Housing:** Custom 3D Printed (ABS Plastic)
- **Power Supply:** 120 VAC Wall Outlet to 5V DC Adapter

<u>Server:</u>

Front End:
- **Framework:** Django (Python)
- **Graphing:** MatPlotLib

Back End:
- **Web Server:** Apache 2.2
- **Database:** MySQL
- **Connection Broker:** Redis
- **Task Management:** Celery, Flower (Python)

Housing of Electronics:

Base Unit:

The base unit will always be inside the house and therefore exposed to minimal environmental factors. However, we have decided to protect the base unit electronics with an ingress rating of IP57, ensuring that objects >1mm cannot enter the case and dripping water will have no harmful effects. For prototyping purposes, a waterproof case with an ingress rating of IP57 was used.

Meter Unit:

The meter unit will reside in the meter pit generally located on the sidewalk in front of most homes. It will will be exposed to a variety of environmental factors and must be capable of withstanding complete submersion, therefore it has been assigned an Ingress Rating of IP57. The meter unit housing has been designed for robustness and optimal performance for various environments. The Red Bear *mini* is the most difficult piece of electronics that must be protected. The board was placed in a waterproof case with an ingress rating of IP57 as well.

For future versions, brick electronics potting will also be used, providing an Ingress Protection of 57. This method of waterproofing not only provides superior protection, but is low cost and easy to manufacture. The 3D printed housing is not critical for weather resistance, only to make the potting process easier and aesthetics. Therefore, it has been assigned an ingress rating of 14 which will sufficiently keep the already protected components separated from dirt particles and light water exposure. The magnetometer sensor wire and antenna will require weatherproof sheathing along with a small amount of potting epoxy near the sensors. The battery housing selected is rated to IP68 which surpasses the requirement of IP57. All components in the meter unit will have the ability to withstand all types of weather exposure anticipated in the United States. Unexpected, extreme exposure circumstances will require lab testing to find the device's limits of operation.

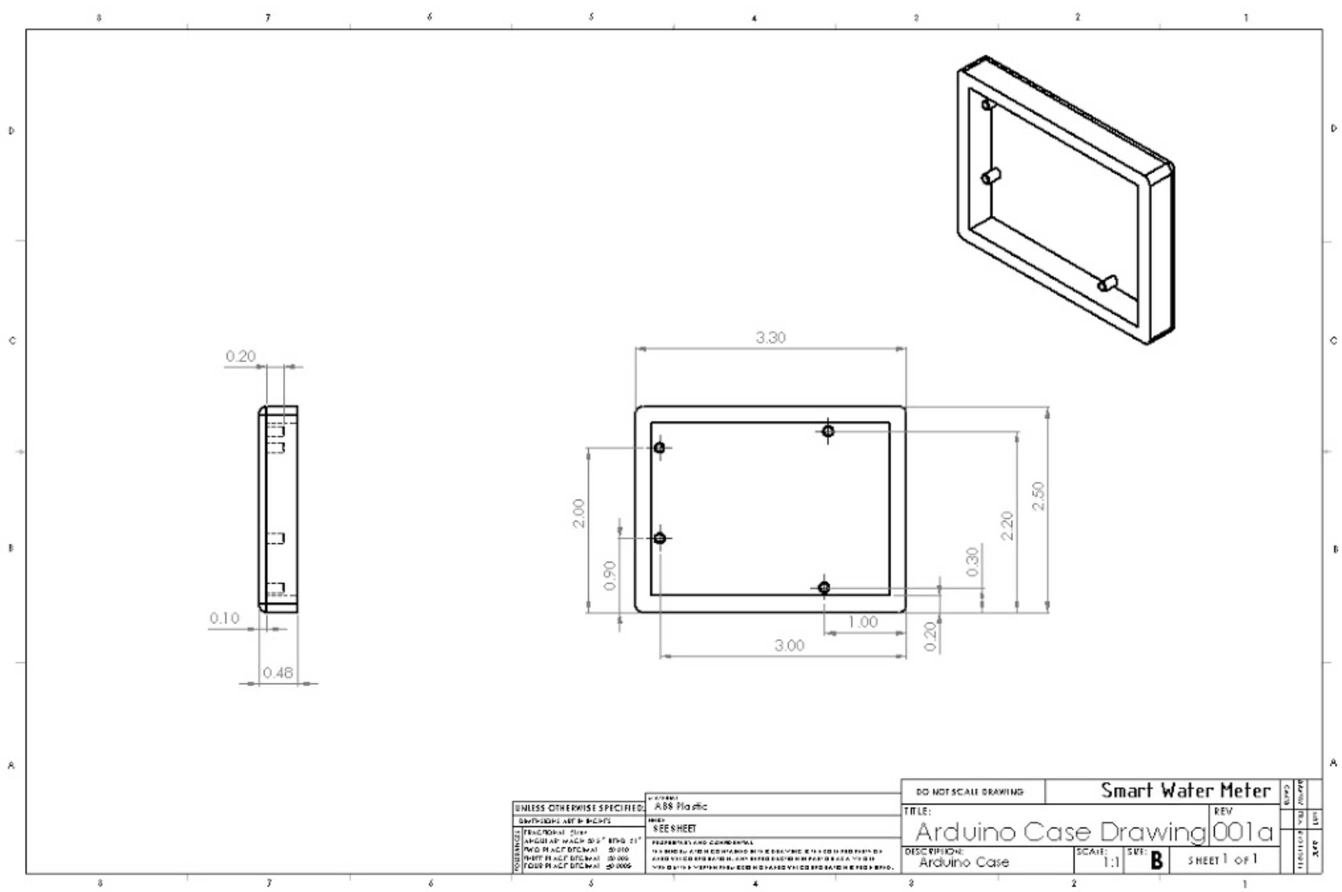


Figure 2: Detail drawing of meter unit housing.

Sensor:

For our Hall Effect sensor, we chose the MAG3110 magnetometer, manufactured by Freescale Semiconductor. It has a sensitivity of 0.1 µT and is capable of running at 2 µA in standby and as low as 10 µA in active mode[13]. The power draw and sensitivity of this device are both an order of magnitude better than the available analog sensors, making it an easy choice. It is also available as a breakout-board that hobbyists can experiment with. Lastly, these sensors are extremely cheap, costing only $1.00 per chip.

We purchased a break-out version of the chip and tested it on one of our home water meters. As expected, the MAG3110 far out-performed our experiments with analog devices. These results can be seen in **Appendix G**.

Power Supply:

To power the meter unit device, 12 Ni-MH rechargeable AA batteries with a total capacity of 27.6 Amp-Hours was chosen (2300 mAmp-Hours per AA battery). After power optimization and sleep modes are implemented, sources indicate that the system's current draw may be as low as 3mA on average. If this level of power consumption is achieved, the battery is estimated to last 383 days.

Microcontroller:

The Arduino Uno was chosen for its versatility, ease of implementation, and low power consumption. The Uno has more than enough processing power for our design and is compatible with a large assortment of sensors and add-on modules. In addition, the Uno has a large, diverse developer community which offers additional support and troubleshooting advice. Finally, Arduino is an open-source platform and allows developers to build customized boards based on its architecture.

BLE Mini prototype boards were chosen for both the base meter unit Bluetooth communication. These microcontrollers are designed to be compatible with the Arduino framework and have built-in Bluetooth 4.0/LE support. The BLE Mini can act as both a slave and a master for communication, and so it can easily be reconfigured as a base or peripheral.

Data Relay:

In order to relay meter data to our web server, we have chosen Ethernet and Bluetooth components which are peripheral to the Arduino Unos. A functional block diagram demonstrating the flow of information between components and devices can be seen in **Figure 4** below.

---

[13] http://cache.freescale.com/files/sensors/doc/data_sheet/MAG3110.pdf

This solution was chosen mainly because of its potential to be reliable and extremely energy efficient. The new Bluetooth LE standard (also known as Bluetooth 4.0) features a range of unobstructed 100+ meters, a peak current consumption of 15 mA, and data transfer rate of 0.27 mbps.[14]

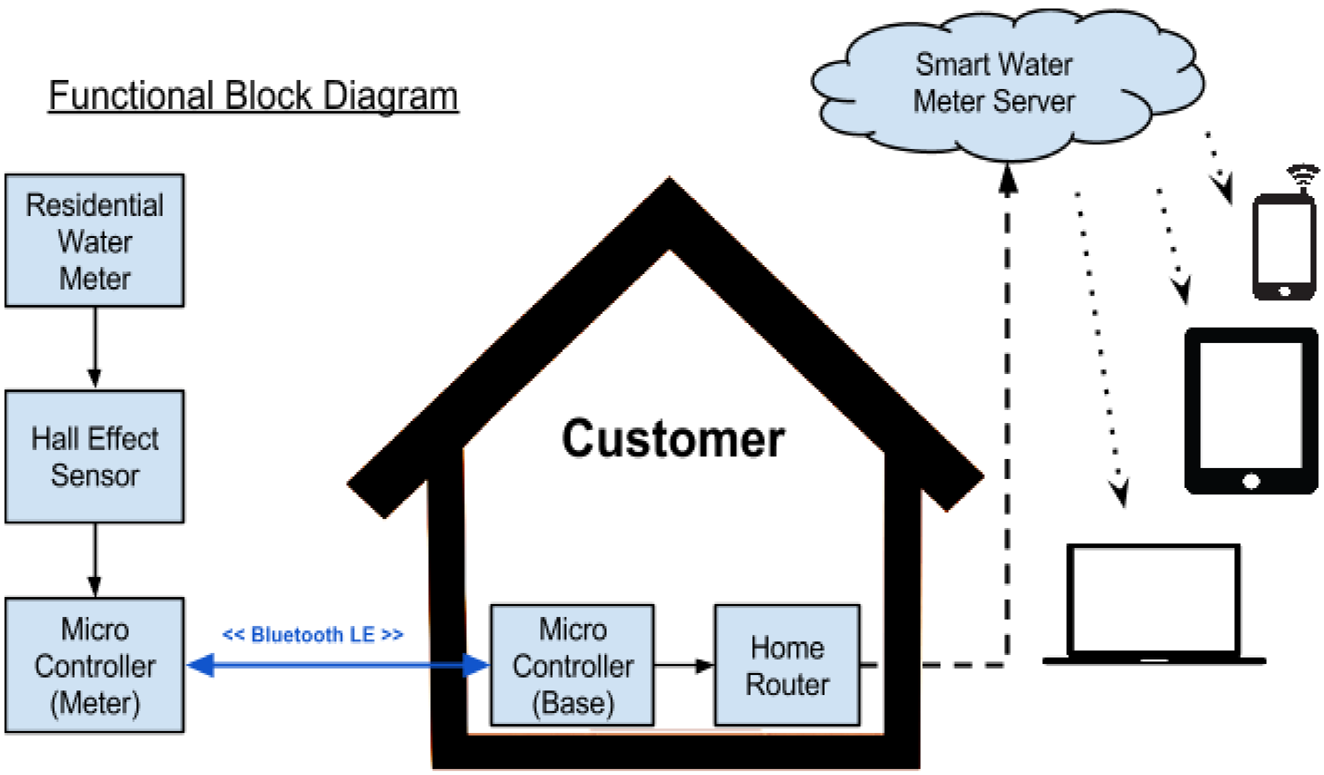


Figure 3: Data Relay Flowchart

---

[14] http://en.wikipedia.org/wiki/Bluetooth_low_energy

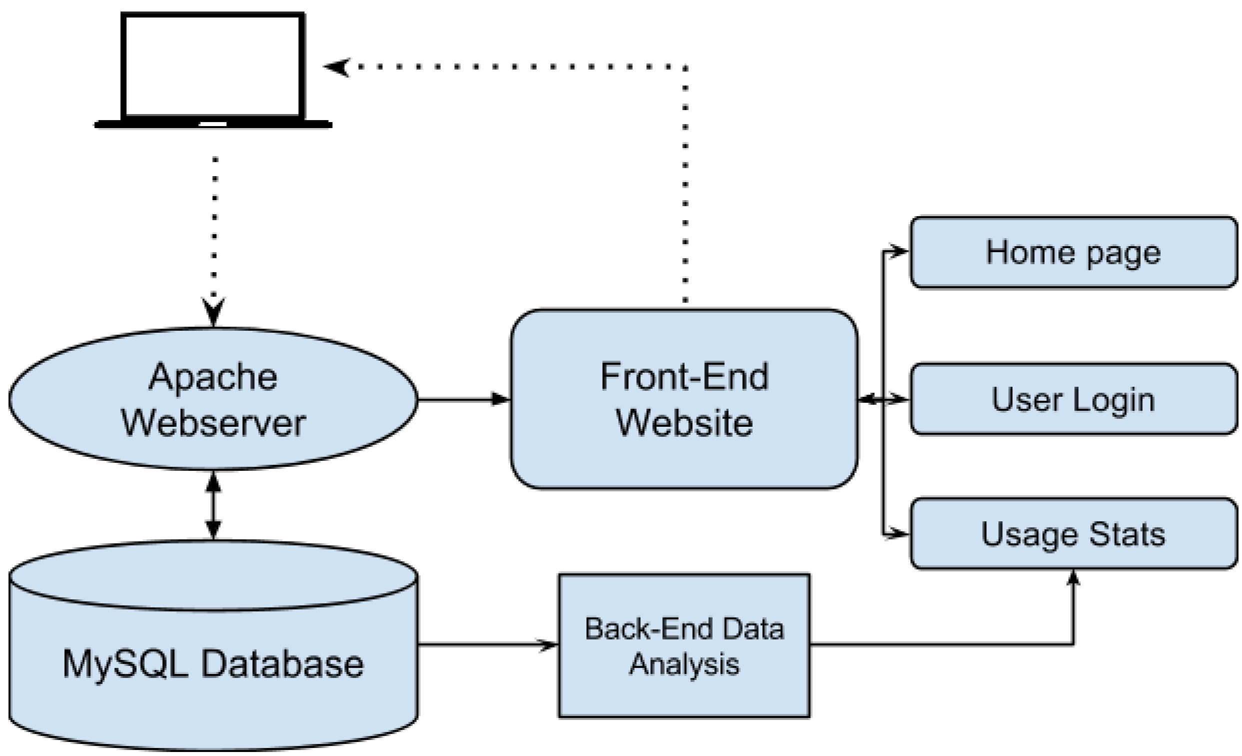


Figure 4: Webapp Diagram

Cost:

Overview of Total Cost Estimation of First Working Prototype: $197

Arduino Uno + BLE Mini Module ~ $40
Arduino Uno + BLE Mini Module ~ $40
Arduino ethernet shield ~ $12
Bluetooth antenna (2) ~ $9
MAG3110 sensor & breakout board: ~ 17$
Arduino housing (2) ~ $5
Battery housing: ~ $30
Rechargeable AA batteries: ~ $23
Encapsulating potting epoxy ~ $20
Miscellaneous wires ~ $1

Consumer Price Point Analysis:

Since there is currently no product on the market that measures your residential water usage in real time and makes that data available to homeowners, it is difficult to determine to what extent water consumption will be reduced. On average, Californian household uses 360 gallons per day,[15] yet studies have shown that individuals only need to use around 27 gallons per capita per day.[16] Considering this fact, it is reasonable to assume that with increased awareness, our product will help reduce water consumption by 20%.

In the city limits of San Luis Obispo, a gallon of water costs $0.00925, resulting in average household residents spending on average $100 per month for incoming water.[17] The sewer is roughly equal to the water fee, resulting in a rough estimate of $200, excluding base fees that will be unaffected by reduced consumption. If our product were to reduce the homeowner's water consumption by 20%, an average of $40 would be saved per month or $480 per year.

Another objective of our product is to enable our our customer to save money by making them more aware of costly leaks. According to the EPA, approximately 13.7%[18] of water used in homes is a result of a leak. If people were made aware of this and salvaged 10% of their water leaks, that would equate to, in San Luis Obispo, 36 gallons per day. Over the course of a year this would equate to over 13,000 saved gallons of water. Based on the water rates inside the City of San Luis Obispo[19], this would result in a savings of roughly $122 each year. Combining these

---

[15] KQED News: http://blogs.kqed.org/lowdown/2014/01/23/how-much-water-do-californians-use-each-day-and-what-does-a-20-reduction-look-like/
[16] Environment Victoria: http://environmentvictoria.org.au/content/leave-target-155-dead-how-use-less-100-litres-water-day#.VNnB6kfF98E
[17] San Luis Obispo Water Utility: http://www.slocity.org/utilities/billing.asp#Water1
[18] http://www.epa.gov/WaterSense/pubs/indoor.html
[19] http://www.slocity.org/utilities/billing.asp#Water1

leak detection savings with reduced consumption and extrapolating over the course of a year, our product would have the ability to save a homeowner on average $600 and that is only with a 20% reduction water use. The product would pay for itself in merely two months.

NOTE: These are rough estimates and a more detailed price-cost analysis will be needed in the future.

Chapter 5: Manufacturing and Design Verification (Testing)

Ingress Protection Rating:

In order to obtain a certified Ingress Protection we will have to send our product away to a testing facility where the necessary procedures will be performed, ensuring that the product meets our standards. However, for prototyping purposes, there are a variety of rough testing procedures that can be done to determine whether or not we are in the ballpark of our standard.

An Ingress testing facility utilizes water jets and depth tanks to test how waterproof a product is. Our hardware may encounter situation where it will be fully submerged. We want to exceed these requirements to provide a factor of safety, so we have chosen to reach for a liquid ingress protection rating of 7:

> *"Ingress of water in harmful quantity shall not be possible when the enclosure is immersed in water under defined conditions of pressure and time (up to 1 m of submersion)."*
>
> *"Test duration: 30 minutes*
>
> *The lowest point of enclosures with a height less than 850 mm is located 1000 mm below the surface of the water, the highest point of enclosures with a height equal to or greater than 850 mm is located 150 mm below the surface of the water ."* [20]

This testing procedure was replicated with a tank of water large enough to submerge the device for the specified time and conditions. Typical water meters are located 1-3 feet below the surface depending on the region and location of the frost line.[21] This test was performed with just the case to ensure that our prototype was damaged. The case did not leak and the test was deemed a success.

Although our product will not endure physical impacts during its application, we must create a housing that can withstand human error impacts from installation, such as accidental drops. Each component will be reviewed and investigated before testing to determine the amount of acceleration force it can withstand. Following data collection from component spec sheets, the prototype hardware will be drop tested within its housing from various heights to determine functionality following an impact. We have set a drop height target of five feet as our nominal goal.

[20] Wikipedia - IP Code: http://en.wikipedia.org/wiki/IP_Code
[21] Water Line Depth: http://www.irrigationtutorials.com/faq/new-water-main.htm

Temperature Testing Procedure:

We must ensure that our product is capable of operating in a range of temperature between 0℉ and 140℉. The Arduino, Hall sensor, batteries, housings and miscellaneous components must be all rated within this temperature range.[22][23][24][25] However, testing must be performed before the prototype is complete. A basic oven with a temperature probe will be used to confirm the high end of the temperature range and a freezer will be utilized to determine the low end of the temperature range.

Procedure:

High Temp Test

1. Heat oven to 140℉.
2. Record temperature for 5 minutes to confirm steady state temp has been obtained
3. Place potted microcontroller, magnetometer, antenna and battery pack in oven
4. Manually connect to the controller
5. Send data from test unit to base unit
6. Ensure base unit is fully connected and receiving data
7. Run test for one hour

Low Temp Test

1. Reduce temperature of freezer to 0℉
2. Record temperature for 5 minutes to confirm steady state temp has been obtained
3. Place potted microcontroller, magnetometer, antenna and battery pack in freezer
4. Manually connect to the controller
5. Send data from test unit to base unit
6. Ensure base unit is fully connected and receiving data
7. Run test for one hour

Humidity Testing Procedure:

A variety of DIY plans are available for creating a simple and effective humidity chamber that will be sufficient for preliminary testing of our product. We will construct a basic

[22] Arduino Spec Sheet: http://www.atmel.com/Images/doc8161.pdf
[23] Ni-MH Batteries http://en.wikipedia.org/wiki/Nickel%E2%80%93metal_hydride_battery
[24] ABS Plastic http://en.wikipedia.org/wiki/Acrylonitrile_butadiene_styrene
[25] See DRV5053 Hall Effect Sensor Data Sheet

humidity testing chamber with an incorporated humidity gauge, which are inexpensive and readily available online and at local hardware stores. Testing procedures similar the the nist.gov will be followed as a guide to properly test.[26]

<u>Humidity Test</u>

1. Set relative humidity in test chamber to 100%
2. Run device for four hours in test chamber
3. Perform system diagnostics
4. Pinpoint what aspect of system was affected by moisture
5. Reiterate design to protect device from moisture

Safety Considerations:

The product has been designed to minimize risk to both the user and the product itself. The battery packs are housed in IP57 cases in order to prevent shorting in the case that water enters the pit. Additionally, the wires running from the battery packs have been heat-shrinked and epoxied to prevent shorting wires and/or fraying wires. Similarly, the microcontroller housing has an ingress rating of IP57 which will eliminate this risk while the housing is properly sealed. The unit has not been sealed yet, however, to allow for continued testing and debugging.

Water Accuracy Testing:

A closed loop water recirculation apparatus was designed and built to test our magnetometer sensor accuracy and placement location. The system included a submersible variable bilge pump, inline electronic flow meter, inline municipal flow meter and manual shut off valve. The variable bilge pump allowed us to change the flow rate of the system, mimicking a residents dynamic water usage. A positive displacement municipal flow meter has been donated by the water company, providing us with an exact replica of the flow meters used throughout California cities.

Flow rates between 0 and 30 gpm were tested and the magnetometer consistently gathered magnetic field data without ever missing a peak. This information was then analyzed via our peak detection software, which was able determine amount of magnetic revolutions based on the peaks and return volumetric usage data.

Testing Methodology:

1. Plug pump into an analog switch that can be easily turned on or off
2. Begin recording data with the microcontroller + sensor

[26] Testing procedures: http://www.nist.gov/itl/vote/upload/OperatingTemperatureandHumidity.pdf

3. Turn on the pump and run the microcontroller calibration script to obtain high/low peak thresholds.
4. Once the dial on the meter nearly reaches the 12:00 on the gauge, switch off the pump.
5. Clear the data set on the microcontroller, saving the previously obtained thresholds
6. Turn on the pump and wait for the dial to turn three full revolutions, stopping the pump once the meter reaches 12:00 for the third time.
7. Compare the meter's specified tick:water-volume ratio to the number of peaks counted by the microcontroller.
8. Repeat steps 1-7, adjusting the ratio each time until the accuracy of the peak count reaches 95%.

Results:

Our initial test showed us that our original formula of 0.1 gallons of water for every peak was off by nearly 50%. After running the test several times, we arrived at a new ratio of 0.043 gallons of water for each peak counted by the sensor.

Magnetometer Testing:

The MAG3110 sensor has been tested to work well in a variety of different locations within the meter pit. Other factors tested were the sensor sampling rate, the data output rate, as well as the flow rate through the meter. See Appendix H. for all associated test graphs.

Testing Methodology:

1. Attach MAG3110 to the meter
2. Turn on pump and run at 0.1 gpm
3. Begin recording magnetometer data using the microcontroller attached to PC
4. Increase flow rate in 0.1 gpm increments until 1 gpm is reached
5. Increase flow rate in 0.5 gpm increments until 30 gpm is reached
6. Save all data sets in separate files
7. Run Python script to process data and output graph
8. Repeat steps 1-7, placing the sensor on different parts of the meter
9. Compare all graphs. Select the best graph based on the following criteria:
    a. Signal noise
    b. Measurement delta
    c. Smoothness of curve
    d. Accuracy of peak count
    e. Consistency of peak location

Results:

By comparing the output of all the graphs, we determined that the best placement for the sensor was directly beneath the meter, with the face of the sensor flush with the base of the meter. Electrical tape was used to secure the sensor and prevent any components on the board from shorting due to contact with the meter.

Data Relay/Server:

To test data relay for reliability and consistency, the sensor was coupled with an internet connected microcontroller and server results were compared with actual peak data outputted directly to the computer's Serial interface.

Data Relay (Peripheral to Base):

For the peripheral (meter) to base connection, we experimented with different antenna lengths and locations to determine which variation was the most reliable.

Testing Methodology:

1. Connect microcontrollers (with BLE Modules attached) to two separate computers.
2. Load test BLE communication software
3. Begin moving one of the laptops further and further away, making sure that data can still be sent/received at each point
4. Stop once data can no longer be sent/received and record the distance.
5. Repeat steps 1-4 in the following environments:
    a. Open space
    b. Through walls (One, then two.)
    c. Different antennas (No Antenna, Omni-directional/Patch, 2.5db - 6dB gain)
    d. With one of the units placed inside a meter pit (lid on & off)

Results:

Our original test was conducted without antennas. We were able to reach a distance of 56 feet in open space, but only 34 feet with a wall in-between the units. Next we tested with omni-directional 2.5dB antennas and the range was increased to 125 feet in open space, 90 feet through a wall, 60 feet through, and up to 40 feet with one of the units inside a covered meter pit.

Power Consumption:

To test the power consumption of our device, we removed all unnecessary LEDs and configured the microcontroller and sensor to sample at the lowest rate possible. We used a multimeter to measure the current draw during sleep, active, and data transmit stages on various battery configurations, ranging from 4 Volts to 9 Volts.

Results:

Our initial test revealed the following consumption characteristics:

Device Mode	Current Draw (mA)
Active	40
Sleep	15
Active (Transmitting)	60

Current Draw:

Active: 40mA
Sleep: 15mA
Active (Transmitting): 60mA

Future Steps:

When testing, we discovered that our MAG3110 does not provide sufficient functionality to put the device into sleep mode when the meter is not running (and wake it back up when it is running). After some research, we determined that a newer magnetometer sensor, the LIS3MDL, which is produced by STMicroelectronics, will give us this functionality without any sacrifices to power consumption. We recently received a prototype board and have successfully established communication with the board. Our next step is to configure the device's interrupt pin and integrate the board with the rest of the system.

Chapter 6: Conclusions and Recommendations

We have performed a thorough, complete analysis and have determined that changing course from designing a proprietary ultrasonic flow meter to instead piggybacking flow data off of the municipal meter is advantageous in all respects. A fully functioning proof of concept prototype has been developed and tested. We have successfully proven that it is possible to monitor residential water usage wireless through a concrete lid via a magnetometer, microcontroller and antenna all placed inside the meter pit. Our team members are currently working on application development as well as manufacturability to ensure that taking our product to market will be as seamless as possible.

A team of three Cal Poly graduates, Eric Adler, James Fazio and Jeff Hufford have committed to continue on with the project through the SLO HotHouse via the Accelerator Program. Throughout the summer of 2015 we will not only develop the product further, but create a business model canvas, assess market viability and begin discussions with manufacturing companies.

Acknowledgements

We thank Professor Thomas Katona for his assistance, contribution, insight and enduring patience throughout this project's development. Without his belief in our idea and his acceptance of our concept into the senior project program, this device would likely had never been developed.

Appendix A: Gantt Chart

[Multidisciplinary Senior Project]

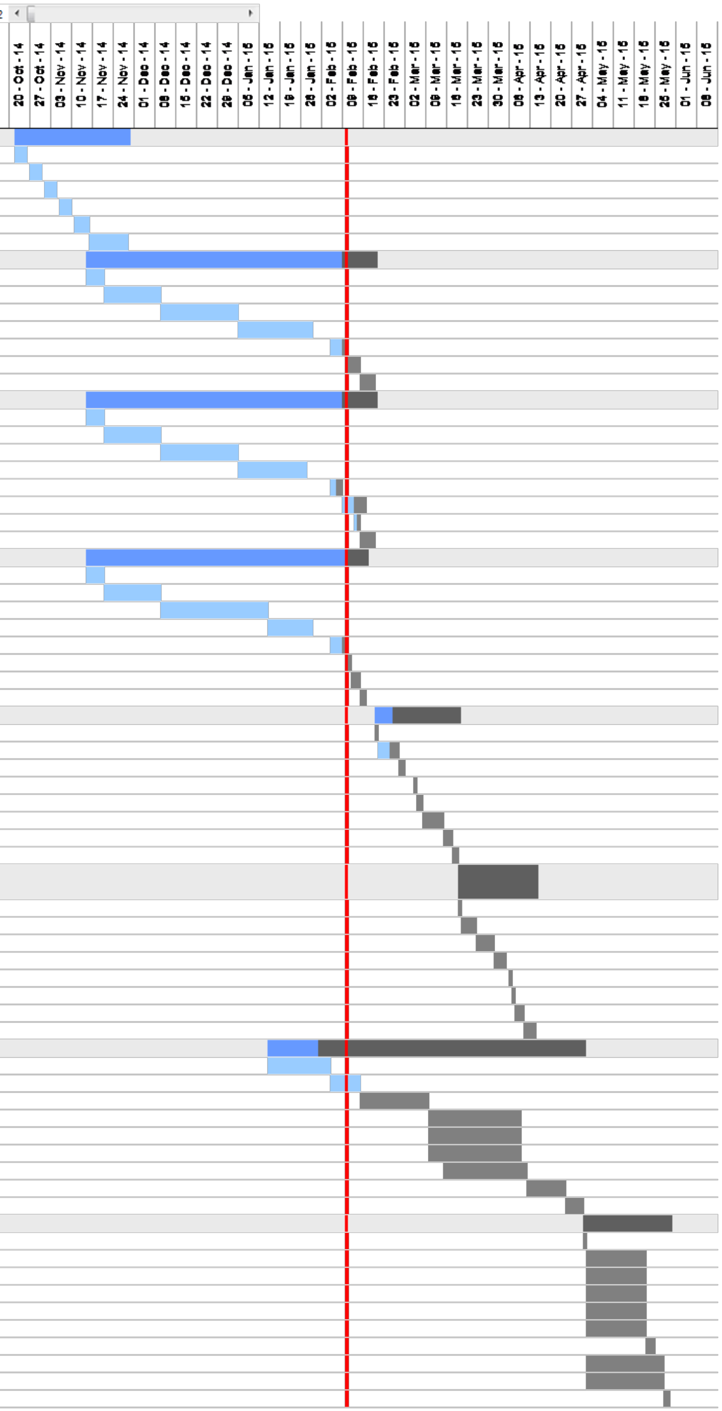
Today's Date: 2/10/2015 Tuesday
(vertical red line)

Project Lead: [Garrett Hall, Eric Adler, .
Start Date: 10/22/2014 Wednesday

[42]

First Day of Week (Mon=2): 2

WBS	Tasks	Task Lead	Start	End	Duration (Days)	% Complete	Working Days	Days Complete	Days Remaining
1	Design Questionaire, Survey,		10/22/14	12/15/14	39	100%	39	39	0
1.1	Design Questionaire Creation		10/22/14	10/26/14	4	100%	3	4	0
1.2	Implement Survey Around SLO		10/27/14	10/31/14	4	100%	5	4	0
1.2.1	Housing Demographic 1		11/1/14	11/5/14	4	100%	3	4	0
1.2.2	Housing Demographic 2		11/6/14	11/10/14	4	100%	3	4	0
1.3	Design Decision Matrix		11/11/14	11/16/14	5	100%	4	5	0
1.4	Conclude Decision Matrix		11/16/14	11/29/14	13	100%	10	13	0
2	Electrical Components		11/15/14	2/20/15	98	89%	70	86	12
2.1	Purchase Devices		11/15/14	11/21/14	6	100%	5	6	0
2.2	Disassemble / Examine		11/21/14	12/10/14	19	100%	14	19	0
2.3	Test Microprocessor and		12/10/14	1/5/15	26	100%	19	26	0
2.4	Troubleshoot		1/5/15	1/30/15	25	100%	20	25	0
2.5	Conclude Test Results		2/5/15	2/11/15	6	70%	5	4	2
2.6	Generate Iteration Plan		2/11/15	2/15/15	4	10%	3	0	4
2.7	Finalize Components		2/15/15	2/20/15	5	0%	5	0	5
3	Flow Sensing Components		11/15/14	2/20/15	98	88%	70	86	12
3.1	Purchase Devices		11/15/14	11/21/14	6	100%	5	6	0
3.2	Disassemble / Examine		11/21/14	12/10/14	19	100%	14	19	0
3.3	Test Accuracy, Reliability,		12/10/14	1/5/15	26	100%	19	26	0
3.4	Troubleshoot Testing		1/5/15	1/28/15	23	100%	18	23	0
3.5	Conclude Test Results		2/5/15	2/9/15	4	60%	3	2	2
3.6	Generate Iteration Plan		2/9/15	2/17/15	8	60%	7	4	4
3.7	Iterate Design If Needed		2/13/15	2/15/15	2	50%	1	1	1
3.8	Finalize Components		2/15/15	2/20/15	5	0%	5	0	5
4	Web Server Set Up		11/15/14	2/17/15	95	93%	67	87	8
4.1	Research Options		11/15/14	11/21/14	6	100%	5	6	0
4.2	Select Initial Method		11/21/14	12/10/14	19	100%	14	19	0
4.3	Test with Simulated Data		12/10/14	1/15/15	36	100%	27	36	0
4.4	Troubleshoot		1/15/15	1/30/15	15	100%	12	15	0
4.5	Conclude Test Results		2/5/15	2/11/15	6	80%	5	4	2
4.6	Generate Iteration Plan		2/11/15	2/12/15	1	60%	2	0	1
4.7	Iterate Design If Needed		2/12/15	2/15/15	3	0%	2	0	3
4.8	Finalize		2/15/15	2/17/15	2	0%	2	0	2
5	Integrate Flow Sensing and		2/20/15	3/20/15	29	22%	21	6	23
5.1	Initial Assembly		2/20/15	2/21/15	1	60%	1	0	1
5.2	Test Microprocessor and Flow		2/21/15	2/28/15	7	60%	5	4	3
5.3	Test Response on Actual Home		2/28/15	3/2/15	2	30%	1	0	2
5.4	Troubleshoot		3/5/15	3/6/15	1	0%	2	0	1
5.5	Conclude Test Results		3/6/15	3/8/15	2	0%	1	0	2
5.6	Generate Iteration Plan		3/8/15	3/15/15	7	0%	5	0	7
5.7	Iterate Design If Needed		3/15/15	3/18/15	3	0%	3	0	3
5.8	Finalize		3/18/15	3/20/15	2	0%	3	0	2
6	Integrate Working Flow Sensing and Electrical		3/20/15	4/15/15	27	0%	19	0	27
6.1	Initial Assembly		3/20/15	3/21/15	1	0%	1	0	1
6.2	Test Distance, Connection		3/21/15	3/26/15	5	0%	4	0	5
6.3	Test on Actual Home		3/26/15	4/1/15	6	0%	5	0	6
6.4	Troubleshoot		4/1/15	4/5/15	4	0%	3	0	4
6.5	Conclude Test Results		4/6/15	4/7/15	1	0%	2	0	1
6.6	Generate Iteration Plan		4/7/15	4/8/15	1	0%	2	0	1
6.7	Iterate Design If Needed		4/8/15	4/11/15	3	0%	3	0	3
6.8	Finalize		4/11/15	4/15/15	4	0%	3	0	4
7	Manufacture Electrical Housing		1/15/15	5/1/15	107	16%	77	17	90
7.1	Brainstorm Designs		1/15/15	2/5/15	21	100%	16	21	0
7.2	Create Design in Solidworks		2/5/15	2/15/15	10	100%	7	10	0
7.3	Prototype Housing		2/15/15	3/10/15	23	0%	17	0	23
7.4	Temperature Testing		3/10/15	4/10/15	31	0%	24	0	31
7.5	Force Testing		3/10/15	4/10/15	31	0%	24	0	31
7.6	Potting Testing		3/10/15	4/10/15	31	0%	24	0	31
7.7	Generate Iteration Plan		3/15/15	4/12/15	28	0%	20	0	28
7.8	Remake if Needed		4/12/15	4/25/15	13	0%	10	0	13
7.9	Finalize		4/25/15	5/1/15	6	0%	5	0	6
8	Final Assembly		5/1/15	5/30/15	30	0%	21	0	30
8.1	Combine All Components		5/1/15	5/2/15	1	0%	1	0	1
8.2	Ingress Rating Test		5/2/15	5/22/15	20	0%	15	0	20
8.3	Humidity Test		5/2/15	5/22/15	20	0%	15	0	20
8.4	Temperature Testing		5/2/15	5/22/15	20	0%	15	0	20
8.5	Drop Test		5/2/15	5/22/15	20	0%	15	0	20
8.6	Troubleshoot		5/2/15	5/22/15	20	0%	15	0	20
8.7	Conclude Test Results		5/22/15	5/25/15	3	0%	2	0	3
8.8	Generate Iteration Plan		5/2/15	5/28/15	26	0%	19	0	26
8.9	Redesign Components If Needed.		5/2/15	5/28/15	26	0%	19	0	26
8.10	Finalize		5/28/15	5/30/15	2	0%	2	0	2



Appendix B: Alternative Microcontroller Comparison

Requirements:

WiFi:
- WiFi 802.11b/g
- Open/WEP/WPA/WPA2 security support

I/O:
- Digital Input support to communicate with Flow Meter
- Digital Output may be desired for LCD output
- Digital input may also be desired for keypad input of some kind

OS:
- Any OS

Memory:
- Large enough for WiFi and digital i/o to function.

Power:
- AC Power Adapter

Desired:

Power:
- Lowest Power Consumption

Cost:
- Cheapest solution

Complexity:
- Least amount of components

	Option 1: **Raspberry Pi + Dongle**	**Option 2:** **Uno + CC3000**	**Option 3:** **Uno + Eth + Router**
Cost	$50	$52	$48
Power Consumption	~200mA	~50mA	~250mA/50mA
Complexity	moderate	moderate	easy

Detailed:

Alternative Option 1: Raspberry Pi + WiFi Dongle

Item	**Price** (less shipping)	**Power Consumption**	**URL**
Raspberry Pi (Model A)	$35	~120mA	http://amzn.com/B00BC0ZL88
Edimax EW-7811Un	$9	Varies	http://amzn.com/B003MTTJOY
Voltage Regulator	~$6	Varies	
Total	$50	~200mA	

Raspberry Pi: Model A must be chosen because it can run on about 120mA, as opposed to the Model B which requires 500mA or more.

Voltage Regulator: A voltage regulator is needed to output a steady 3.3v to the Pi.

- (Alternate 1) MCP1825S Linear Regulator: http://amzn.com/B00LQP7UP6

Less efficient. Only advantage is it requires less batteries, and thus less weight.

- (Alternate 2) LM2596 Switching Regulator: http://amzn.com/B00EZA2G6W

2-3x more efficient. Preferred option.

WiFi Dongle: The Raspberry Pi will require a WiFi dongle to connect to a home network. These are cheap, readily available, and require little to no configuration.

Comments: This option will be the easiest in terms of configuring the device to connect to WiFi. However, the Pi consumes much more power than the Arduino and requires a voltage regulator in order to function at all. Lastly, the Pi does not have an onboard ADC, so an additional chip will have to be purchased if that functionality is ever required.[27]

[27] http://www.daveakerman.com/?page_id=1294

Alternative Option 2: Arduino Uno + Adafruit CC3000

Item	**Price** (less shipping)	**Power Consumption**	**URL**
Arduino Uno	$0/$25	~45mA	http://amzn.com/B006H06TVG
Adafruit CC3000	$37	Varies	http://amzn.com/B003MTTJOY
Total	$37/$52	~100mA	

Arduino Uno: I already have one, but otherwise they are $25, have an onboard ADC, and support digital IO. They also consume much less power. A drawback, however, is that they have limited memory available and the Adafruit CC3000 WiFi libraries would take up a good amount of that. We shouldn't need too many other libraries though so it shouldn't be an issue. No voltage regulator is needed and it can run on a standard 9 Volts.

WiFi Dongle: Adafruit CC3000 is more primitive compared with the Pi's robust WiFi libraries. However it does support WPA2, TCP, and can relay integers to a server. Ideally this is all that's needed.

Comments: Apparently the Adafruit CC3000 is difficult to work with and will require some debugging. However, the Uno would be nice as it consumes much less power. This option would require less components and have a better form factor. It is rumored that cheaper, better-functioning WiFi dongles are on the way for the Uno, so there is a possibility that this option could be upgraded in the future.[28]

[28] http://www.adafruit.com/products/1469

Alternative Option 3: Arduino Uno + Ethernet Shield ENC28J60 + TP-LINK WR702N

Item	**Price** (less shipping)	**Power Consumption**	**URL**
Arduino Uno	$0/$25	~45mA	http://amzn.com/B006H06TVG
Ethernet ENC28J60	$6	negligible	http://www.newegg.com/Product/Product.aspx?Item=9SIA3ZB1CB5437
TP-LINK WR702N	$17	~200mA	http://amzn.com/B007PTCFFW
Total	$23/$48	~250mA	

Arduino Uno: I already have one, but otherwise they are $25, have an onboard ADC, and support digital IO. They also consume much less power. No voltage regulator is needed and it can run on a standard 9 Volts.

Ethernet Shield: The ENC28J60 would be connected to the Uno and then communicate with the nano-router.

Nano-router: The TP-LINK WR702N would have to run on a separate power supply and make this option the highest in terms of base power requirements. One thing that would be cool about using the nano-router is that it could act as an AP. That way you could connect to it with your laptop or phone and set up the Uno that way.

Comments: This option is the least complicated and would allow us to experiment the most. However, it would consume a lot of power and separate supplies would be required.[29]

[29] http://redd.it/1xnygi

Appendix C: Transmission Throughput

Data Structure

At the most basic level, the device must record the rate of flow using some metric (such as gallons/minute). For the purposes of this project, it will be acceptable to average data collected over a 1 second interval. Each data point must also include a time-stamp. This is because data will be transmitted every minute (in order to conserve power) and therefore precludes us from using the timestamp in the packet header. This will also prove useful in the case of a transmission error. Data can accumulate on the device and be sent once the connection has been re-established.

Timestamp:

The timestamp will take the following format: YY|MM|DD|HH:MM:SS. The millenium can be assumed. Since no single value will ever exceed 99, the timestamp can be accomplished in as little as 6 bytes of data.

Flow Rate:

The rate of flow can be accomplished using a 4 byte floating point number which will represent the rate for that timestamp in gallons/minute.

Combined Structure:

In order for the receiving device to interpret these values, a special byte will placed before and after each time stamp. Thus, each second worth of usage data will be recorded and transmitted using 12 data bytes. This would result in 720 bytes being sent every minute, or just under 1 MB per day per device.

Packet Size

The packet size will differ depending on the technology used. Here we will examine the possible packet structures with both WiFi and Bluetooth LE.

WiFi:

In the case of the Adafruit CC3000 (WiFi), each packet may contain between 46 and 1470 data bytes[30]. Thus a minute of usage data can be relayed with one transmit. It is unknown how long it takes to transmit one packet, so a safe assumption of about a fifth of a minute will be used.

Bluetooth LE:

In the case of Bluetooth LE, each packet may contain between 8 and 27 bytes[31]. BLE is capable of transmitting up to 270000 bits/second or 33750 bytes/second[32]. Thus, sending 720 bytes of data could be accomplished in under 25 milliseconds. It is unknown how long it takes to transmit one packet, so a safe assumption of about a fifth of a minute will be used.

[30] http://e2e.ti.com/support/wireless_connectivity/f/851/t/184093
[31] http://www.litepoint.com/wp-content/uploads/2014/02/Bluetooth-Low-Energy_WhitePaper.pdf
[32] http://en.wikipedia.org/wiki/Bluetooth_low_energy

Appendix D: WiFi vs. Bluetooth LE Power Comparison

WiFi

An estimation of the average power requirement of the device is described by:

$$P_{Total} = \frac{t_{off}}{t_{total}} P_{Controller(sleep)} + \frac{t_{on}}{t_{total}} P_{Controller(wake)} + \frac{t_{on}}{t_{total}} P_{WiFi(Transmit)}$$

Stated in terms of Current Usage at a given frequency, this gives us

$$Total\ Usage = (I_{Controller(awake)} * t_{awake} + I_{Controller(sleep)}(F - t_{awake}))/F$$

Using power consumption data provided by AdaFruit (WiFi module manufacturer)[33],

$$I_{Controller\&WiFi(sleep)} = 20mA \quad I_{Controller(wake)} = 45mA \quad I_{WiFi(wake)} = 90mA$$

Estimates: *(determined as current per unit-time)*

Assumes: WiFi only transmitting for 1/5th of on time.

$$I_{6-hrs-on} = I_{C(wake)} \frac{6hr}{24hr} + I_{C\&WF(sleep)} \frac{18hr}{24hr} + I_{WF} \frac{6hr}{24hr} * \frac{1}{5} = 30.75mA$$

$$I_{12-hrs-on} = I_{C(wake)} \frac{12hr}{24hr} + I_{C\&WF(sleep)} \frac{12hr}{24hr} + I_{WF} \frac{12hr}{24hr} * \frac{1}{5} = 41.5mA$$

$$I_{18-hrs-on} = I_{C(wake)} \frac{18hr}{24hr} + I_{C\&WF(sleep)} \frac{6hr}{24hr} + I_{WF} \frac{18hr}{24hr} * \frac{1}{5} = 52.25mA$$

$$I_{24-hrs-on} = I_{C(wake)} \frac{24hr}{24hr} + I_{C\&WF(sleep)} \frac{0hr}{24hr} + I_{WF} \frac{24hr}{24hr} * \frac{1}{5} = 63mA$$

Given a 3.3V operating voltage, this gives us an average power requirement of **208mW** for the always-on state. Depending on the elapsed time during which water is flowing in a house, average power consumption will vary. But given that there may be leaks in a home, the device must be able to operate continuously and so the worst case power consumption is listed in the table.

Note: WiFi power is determined separately as data transmission is not constant.

[33] https://learn.adafruit.com/low-power-wifi-datalogging/no-optimizations

Bluetooth LE:

Using standard power consumption data for Bluetooth LE [34],

$I_{Controller\&BLE(sleep)} = 20mA \quad I_{Controller(awake)} = 45mA \quad I_{BLE(awake)} = 15mA$

Estimates: *(determined as current per unit-time)*

Assumes: BLE only transmitting for 1/5th of on time (one transmit every minute).

$$I_{6-hrs-on} = I_{C(wake)}\frac{6hr}{24hr} + I_{C\&BLE(sleep)}\frac{18hr}{24hr} + I_{BLE}\frac{6hr}{24hr} * \frac{1}{5} = 27mA$$

$$I_{12-hrs-on} = I_{C(wake)}\frac{12hr}{24hr} + I_{C\&BLE(sleep)}\frac{12hr}{24hr} + I_{BLE}\frac{12hr}{24hr} * \frac{1}{5} = 34mA$$

$$I_{18-hrs-on} = I_{C(wake)}\frac{18hr}{24hr} + I_{C\&BLE(sleep)}\frac{6hr}{24hr} + I_{BLE}\frac{18hr}{24hr} * \frac{1}{5} = 41mA$$

$$I_{24-hrs-on} = I_{C(wake)}\frac{24hr}{24hr} + I_{C\&BLE(sleep)}\frac{0hr}{24hr} + I_{BLE}\frac{24hr}{24hr} * \frac{1}{5} = 48mA$$

Given a 3.3V operating voltage, this gives us an average power requirement of **160mW** for the always-on state.

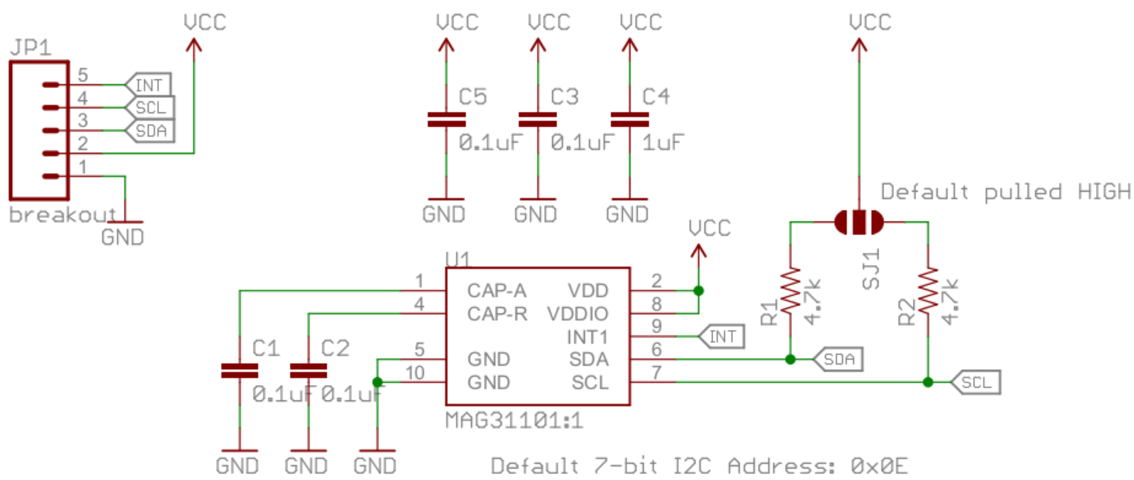
[34] http://en.wikipedia.org/wiki/Bluetooth_low_energy

Appendix E: Bill Of Materials (BOM)

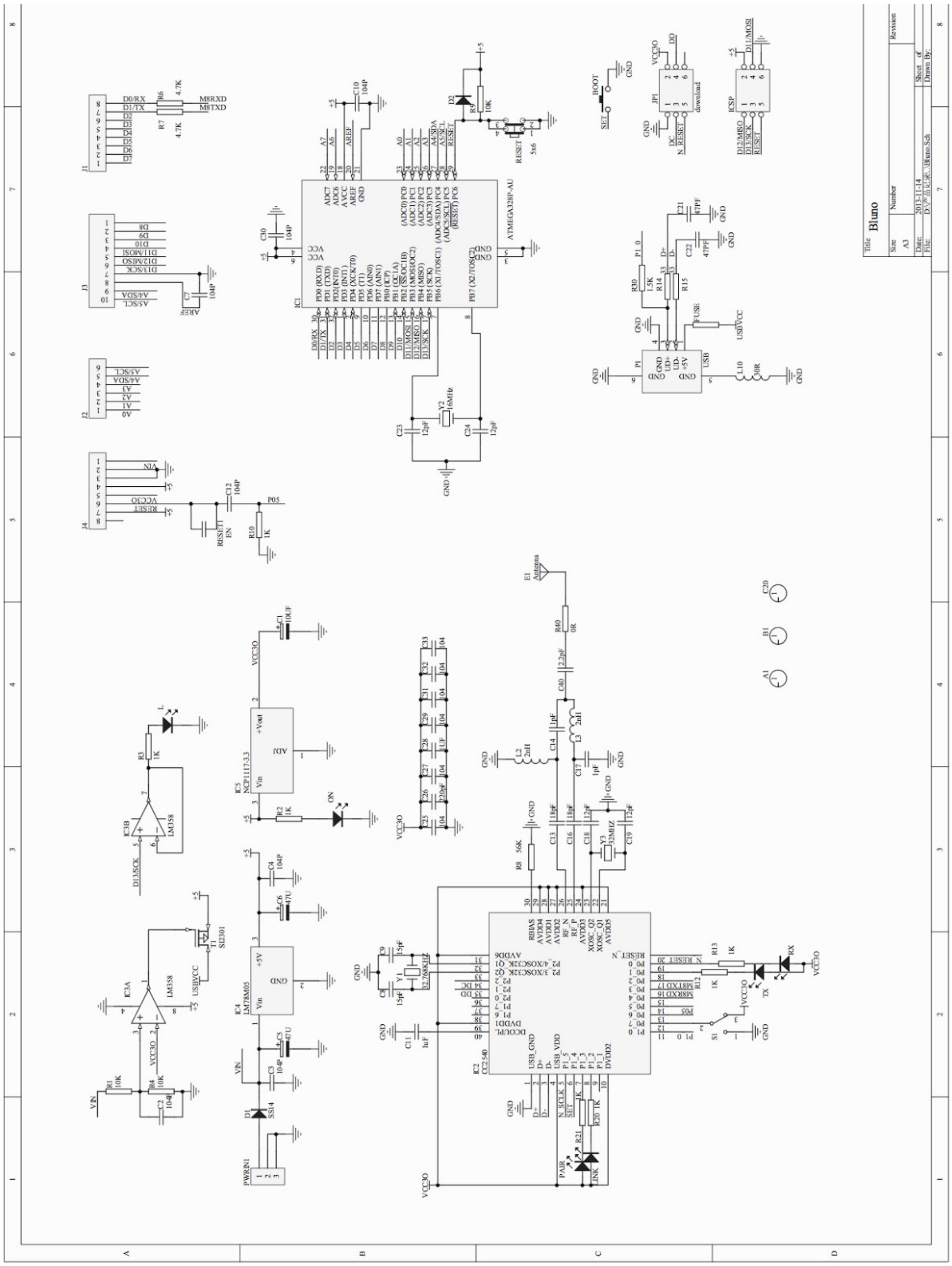
No.	Type	Description	MFG	Price	Qty.
1	Microcontroller	BLE Mini	RedBearLab	$24.99	2
2	Microcontroller	Arduino Uno	Arduino.cc	$14.99	2
3	Ethernet Controller	ENC28J60	Zitrades	$6.67	1
4	Sensor	MAG3110	Freescale Semiconductor	$16.95	1
5	Antenna	2.4 GHz	Rufa	$4.50	2
6	Battery	AA Battery	EBL	$2.00	12
7	Housing	Microcontroller	N/A	$2.50	2
8	Housing	Battery	Philmore	$10.00	3
9	Housing	Antenna	N/A	$3.00	1
10	Routing	Assorted Wires	N/A	$1.00	N/A
11	Routing	Cat5e Cable	N/A	$4.00	1
12	Other	Epoxy	N/A	$20.00	1
Total					
T	-	-	-	$197.58	-

Appendix F: Vendor Supplied Component Schematics and Data Sheets

MAG3110 Sensor Schematic



Arduino Uno Schematic

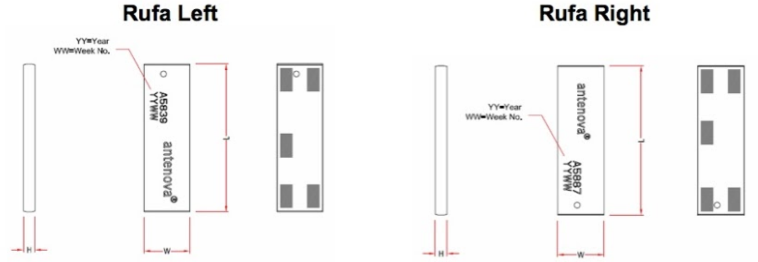


Bluetooth Antenna Schematic

Rufa 2.4 GHz SMD Antenna
Part No. 3030A5839 / 3030A5887

8 Antenna dimensions

Rufa Left **Rufa Right**



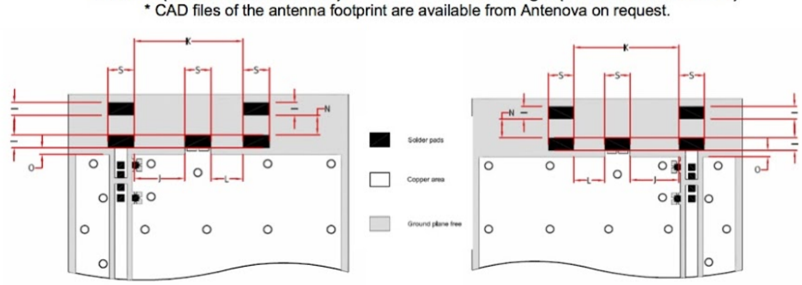
L	W	H
Length	Width	Height
12.8 ± 0.2	3.9 ± 0.2	1.1 ± 0.2

Dimensions in mm

9 Antenna footprint

Rufa Left (Part No: 3030A5839-01) **Rufa Right (Part No: 3030A5887-01)**

* CAD files of the antenna footprint are available from Antenova on request.



Please contact info@antenova.com for further details.

I	S	K	J	N	L	O
1.0 ± 0.1	2.0 ± 0.1	8.1 ± 0.1	3.7 ± 0.1	1.5 ± 0.1	2.4 ± 0.1	0.5 ± 0.1

Dimensions in mm

Integrated Antenna and RF Solutions

5
Product Specification AE020157-M

Appendix G: Design Development

Pro and Con Table for Type of Flow Sensing Method

Ultrasonic ($270)		Turbine Inline ($200+$200)		Installing Magnetic Water Meter ($120+$200)		Pulse Monitoring on City Meters ($60)	
Pro	Con	Pro	Con	Pro	Con	Pro	Con
Non-Invasive	Needs power on the side of house	Self Powered	Need plumber to install (about $200)	Very accurate	Need plumber to install (about $200)	Very accurate	Has to be battery powered
Probably Located within Wi-Fi range	Some home's sprinkler system pipes split away before coming above ground (hoses, sprinklers, and pool water usage would not be taken into account)	Probably Located within Wi-Fi range	Some home's sprinkler system pipes split away before coming above ground (hoses, sprinklers, and pool water usage would not be taken into account)	Probably Located within Wi-Fi range	Some home's sprinkler system pipes split away before coming above ground (hoses, sprinklers, and pool water usage would not be taken into account)	Innexpensive	May need range extenders to reach home's Wi-Fi
Meter has clear output signals to arduino	Innacurate at low flowrates / unreliable	Meter has clear output signals to arduino	Innacurate at low flowrates	No calibration needed	Large pressure drop	No moving parts / Minimal components	Cities may have a problem with attaching device to their water meter.
	Difficult installation and calibration	No calibration needed	Liability of causing a leak and flooding someone's house.		Liability of causing a leak and flooding someone's house.		
	Expensive		Small pressure drop				
Why we are aborting the ultrasonic method					Why we are choosing pulse monitoring on city meters		
- Very unreliable: Spikes and doesn't accurately measure the flow					- Much less expensive		
- Extremely finicky to find optimal set up position. This will vary for different pipe material, sizes, and companies. 0.25mm results in a 5mm changes the suggested spacing. Won't be able to make a mounting rail that successfully spaces the transducers.					- Much more accurate. 1/10gal per pulse		
- Expensive: Already using cheapest flowmeter available and it is over our budget					- Simplicity - using the city's $500 water meter to measure flow		
- Doesn't detect leaks or small flow water usage. If it works 100% above the minimum flow velocity of 0.3m/s then we achieve about 80% water usage data.					- Major aspects of the design have been well documented and proven to work. Hall sensor reliability in spec sheets. City meter works to at least 97% accuracy.		
- Operating at the smallest pipe diameter. The range is 15mm-200mm					- Installation and calibration is much easier than ultrasonic		
- Suggested flow rate minimum is 0.3m/s which doesn't account for leaks or flow lower than 1 L/min. This application is clearly operating outside of the usable zone.					- Still within Wi-Fi range or BLE extender to Wi-Fi		
					- Uses low power. Only has to power arduino, Hall effect sensor, and bluetoon/Wi-Fi attachment		
					Concerns with pulse monitoring city meter		
					- Possible patent infringement with Water Hero - don't have a patent for this technology. This idea has been used and shared on the internet for years.		
					- City may have an issue with attaching device to their meter		
					- Power is still difficult. BLE 4.0 low power and we do not need to power a flow meter anymore. The Hall effect sensor requires a 5v input. Battery power looks much more promising but more research needs to be done		
					- Wi-Fi and BLE will require an antenna to stick above the cement box. These are cheap but will be visible from the outside and be exposed to damage.		

Design Development of Hall Effect Sensor:

DRV5053EA

The graphs below are of our preliminary Hall Effect sensor data from the DRV5053EA placed next to a magnet and a water meter.

Figure 1: Magnet voltage output over ~10 seconds

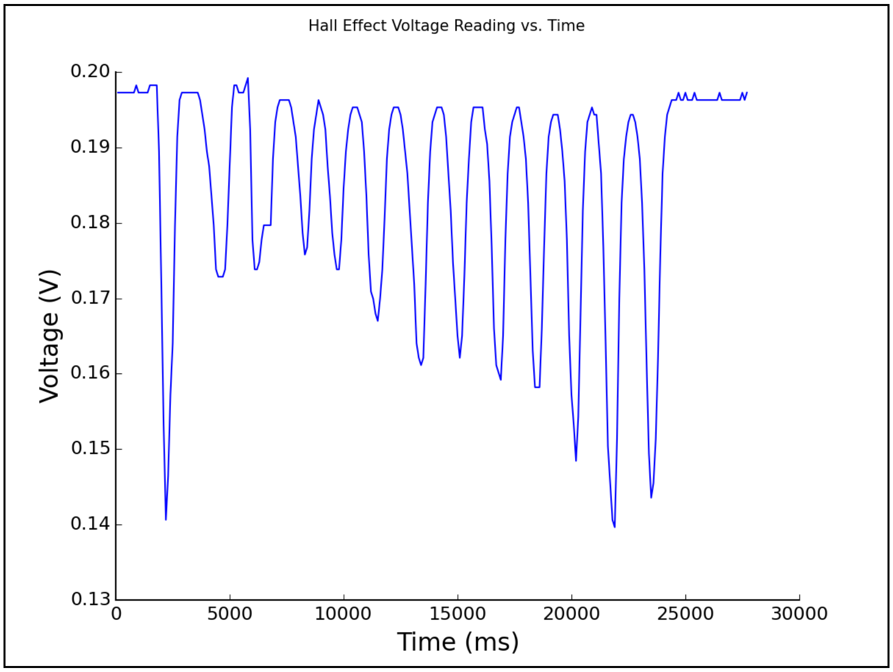


Figure 2: Home Meter - NOTE: Nonamplified resolution is very small (2 points)

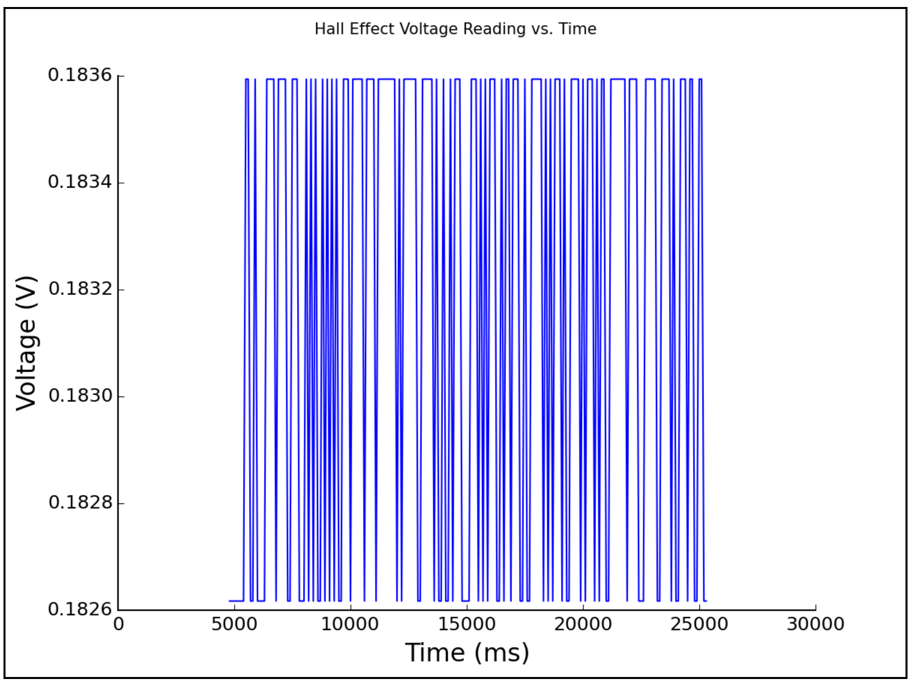
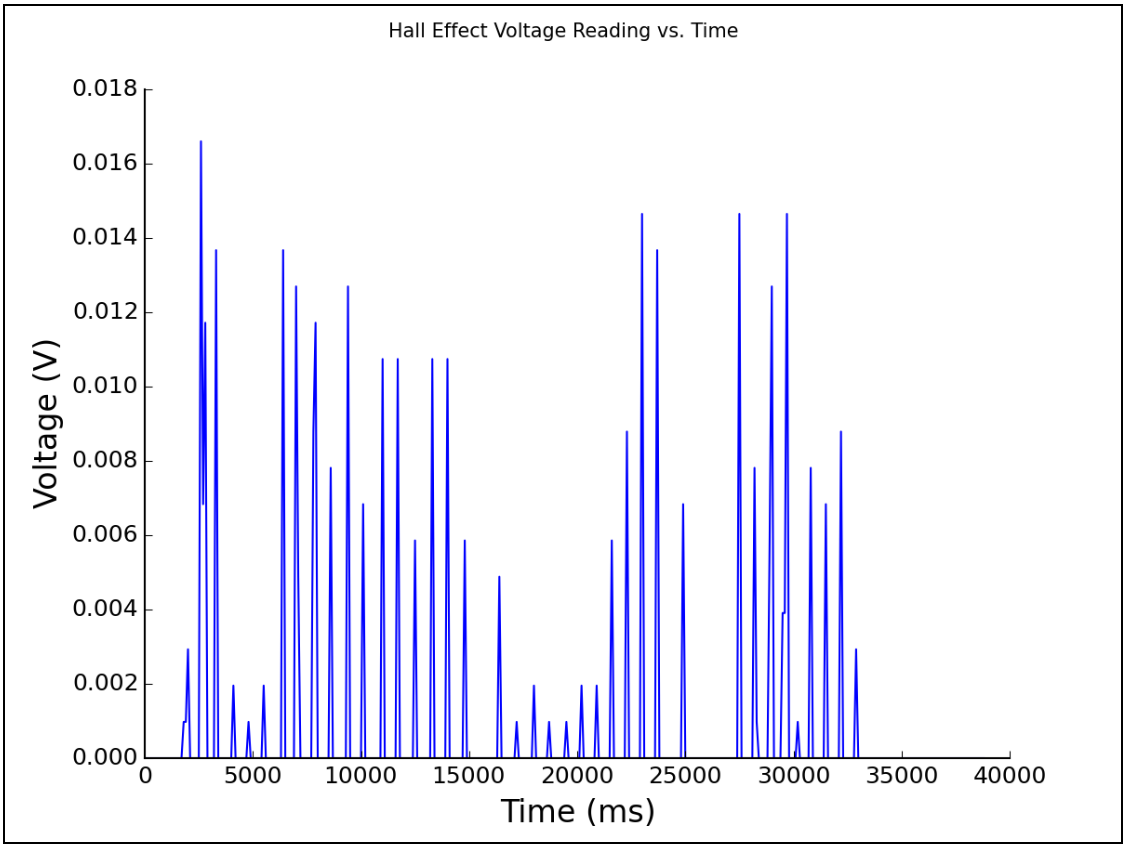


Figure 3: Magnet With Amplifier



MAG3110:

The graphs below are the results of the MAG3110 sensor placed directly above a home water meter.

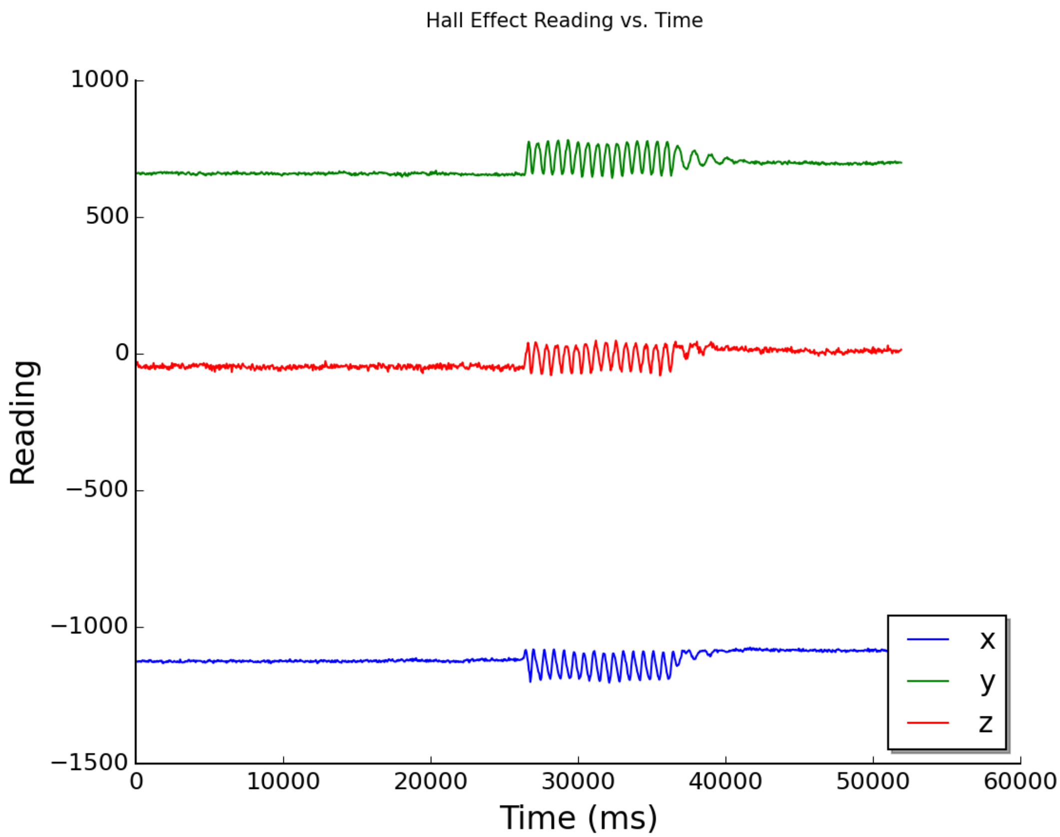


Figure 4: X, Y, and Z axis water meter readings

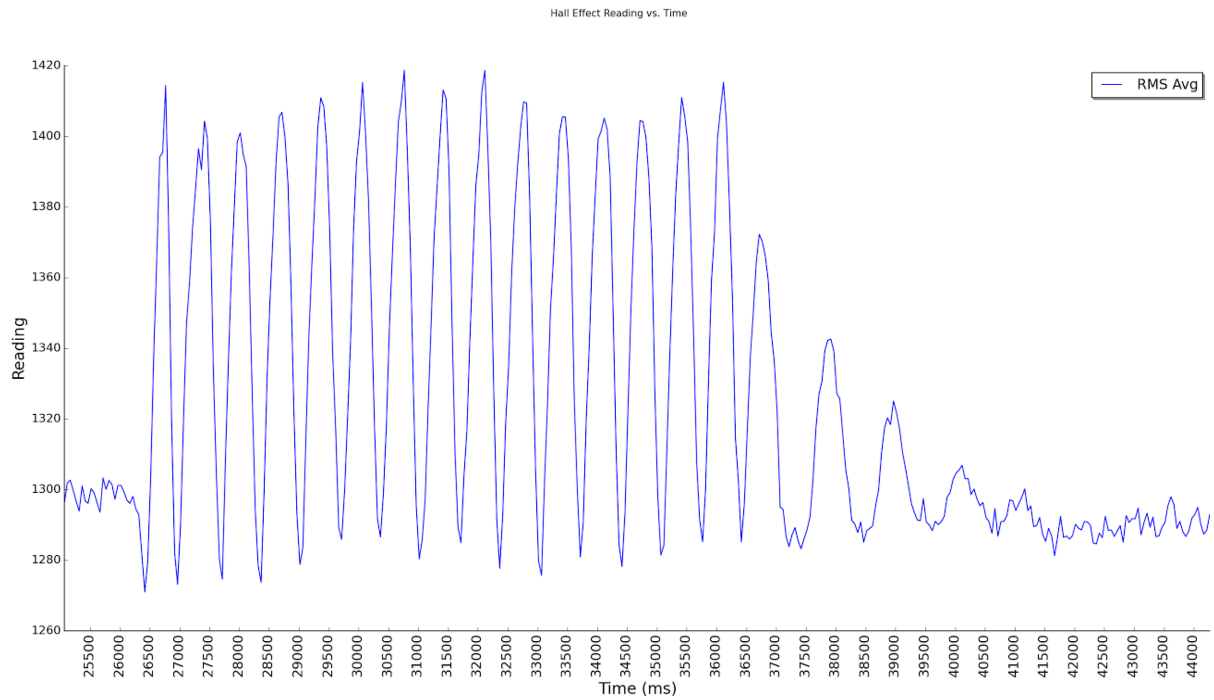


Figure 5: X, Y, and Z RMS Average

Appendix H: Magnetometer Test Graphs

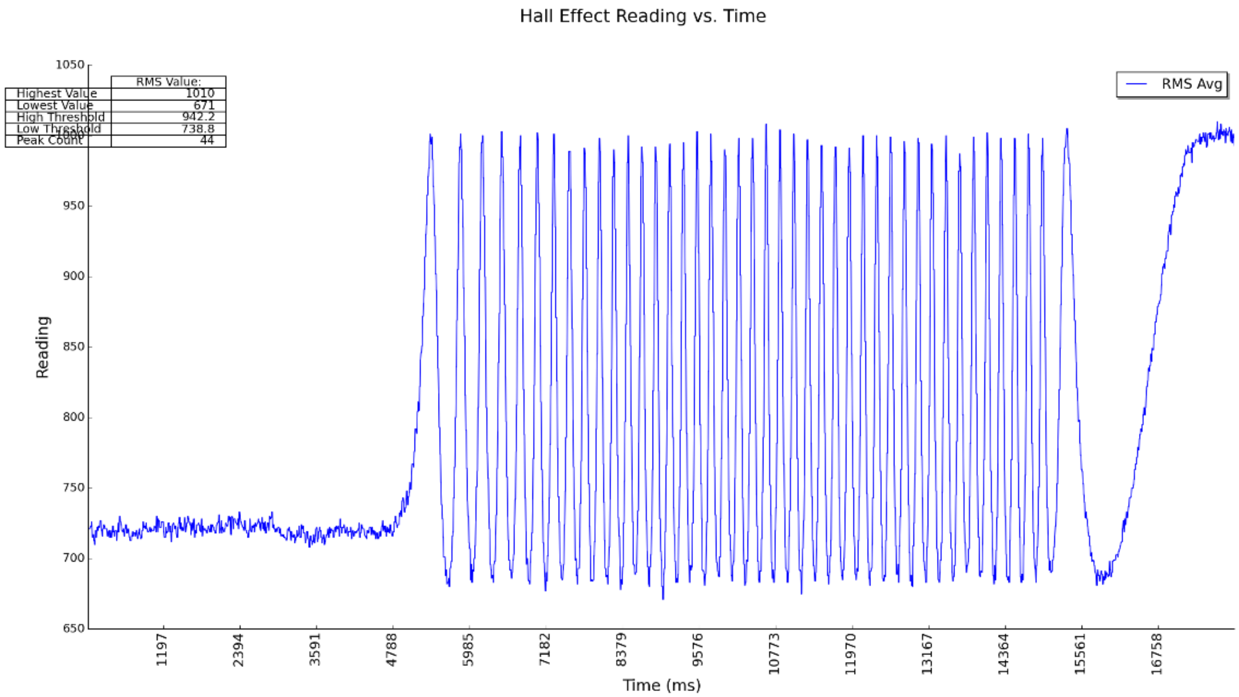


Figure 1: 5ms sample rate, placed directly above meter

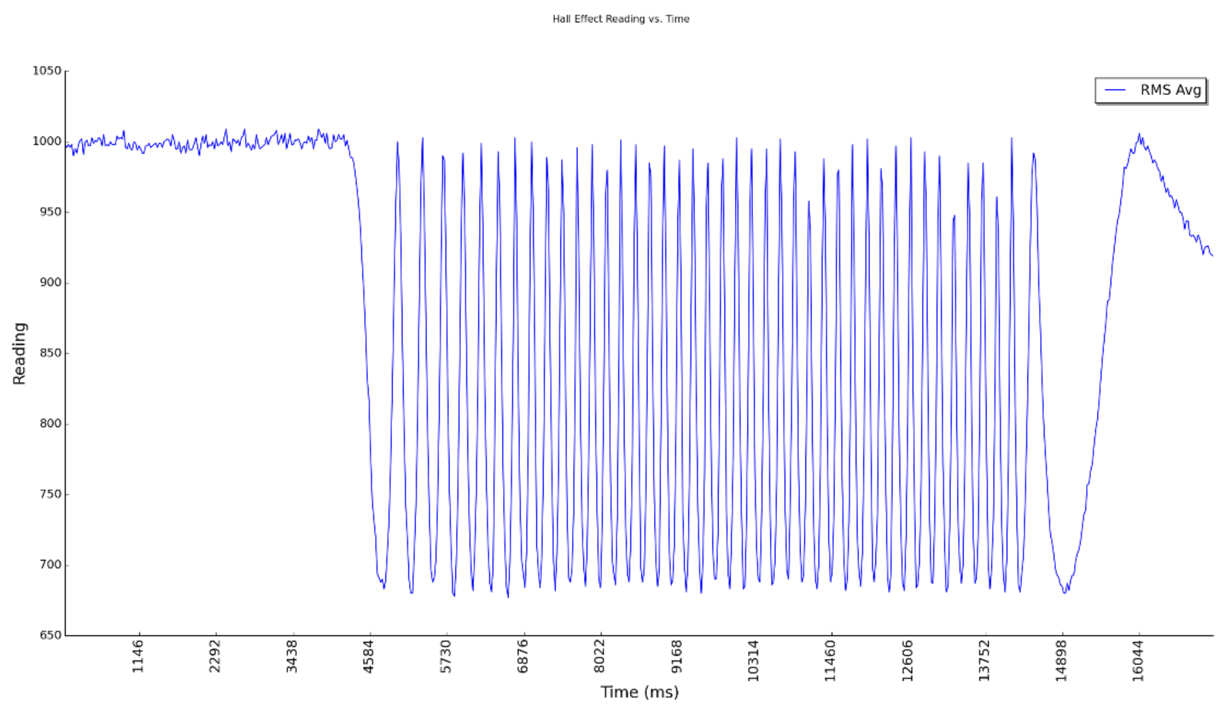


Figure 2: 25ms sample rate, placed directly above meter

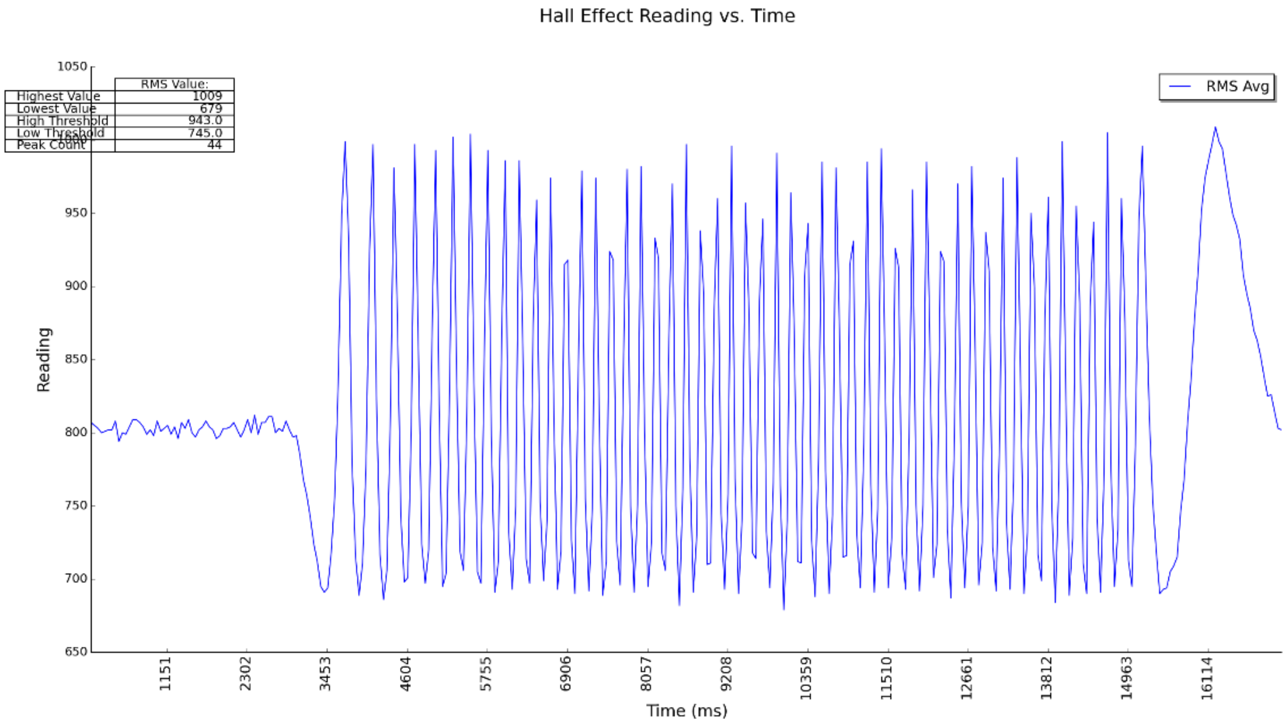


Figure 3: 50ms sample rate, placed directly above meter

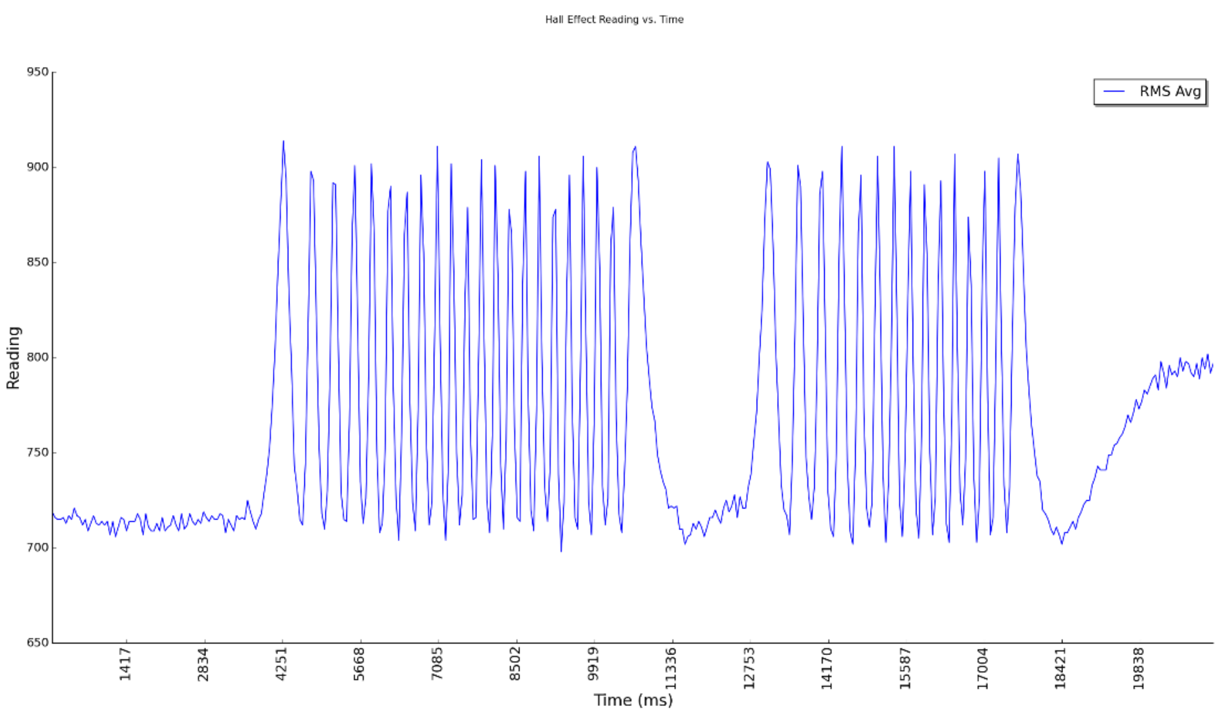


Figure 4: 50ms sample rate, placed on the right side of the meter

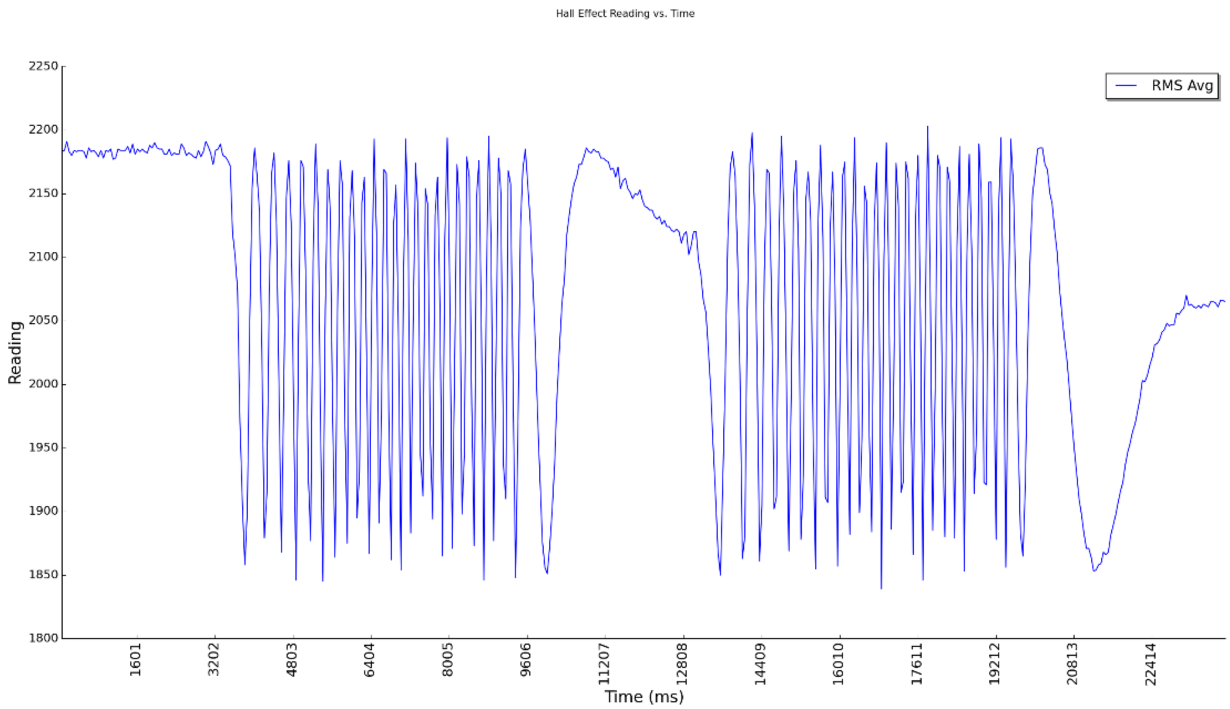


Figure 5: 50ms sample rate, placed top-side down on left of meter

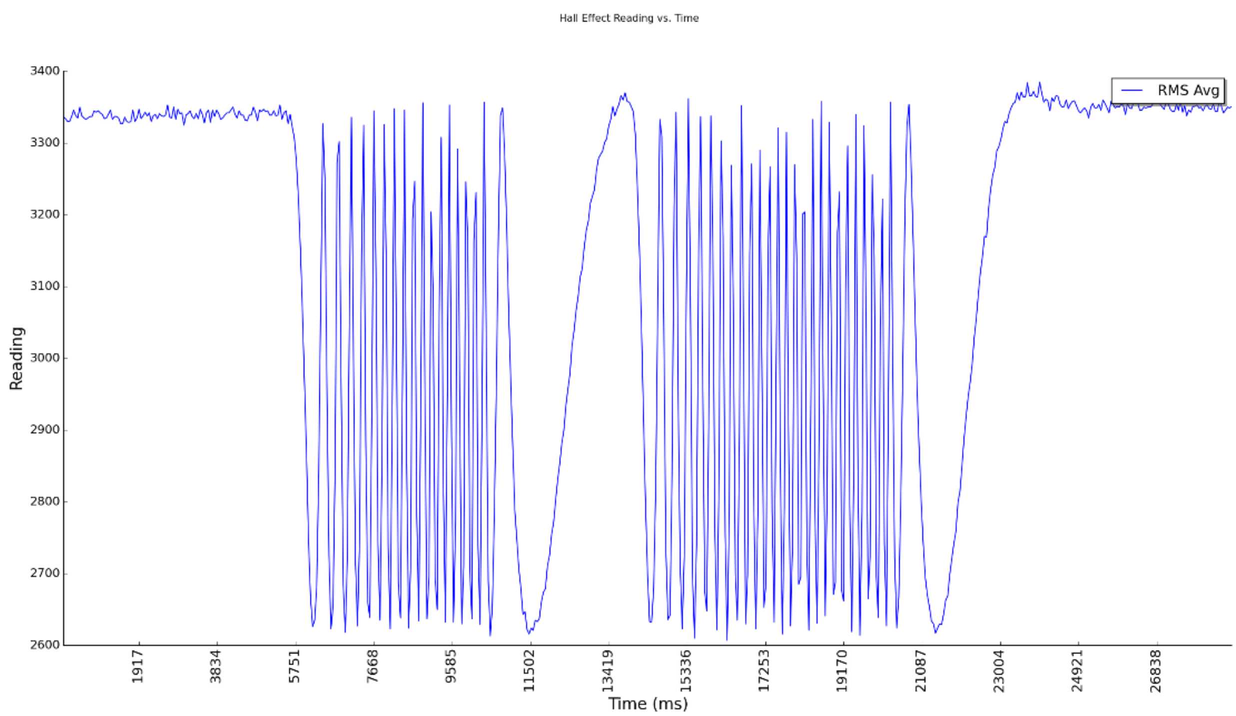
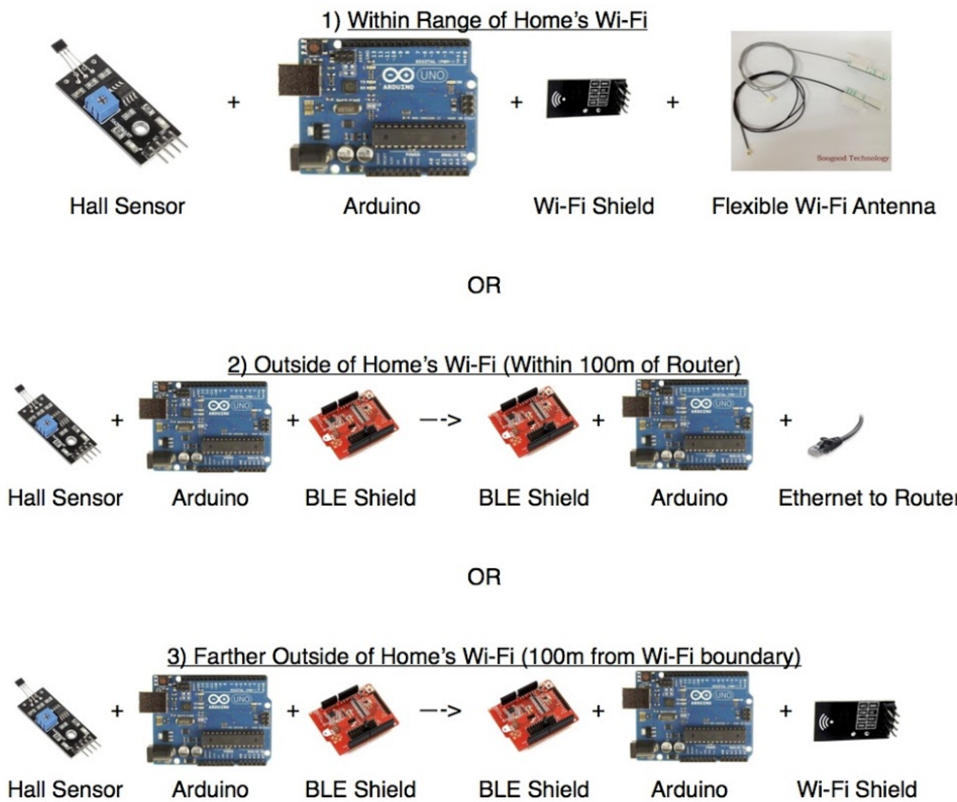


Figure 6: 50ms sample rate, placed top-side (facing meter) on bottom of meter

Appendix I: Microcontroller Comparisons

1. Design Development of Base Unit with Wifi/Ethernet and Master BLE Combinations



2. Design Development of Peripheral Unit (Microcontroller + Bluetooth LE)

1)

Arduino Red Bear Lab BLE Mini (master) Ethernet Shield

2)

Wifi/Ethernet built in Arduino Red Bear Lab BLE Mini (master)

3)

BLUNO BLE built in Arduino (master) Ethernet Shield