Smart Garbage Management

Design Document v2

Team sddec18-08
Client/Advisor: Prof. Goce Trajcevski
Team Members/Roles:
Steven Brown - Hardware Design
RJ Duvall - Web Development
Brendan Finan - Mobile Development
Sam Johnson - Big Data
Colin McAllister - Embedded Systems
Nicholas Pecka - Networking
Team Email: dec1808@iastate.edu
Team Website: sddec18-08.sd.ece.iastate.edu
Revised: 02/04/2018, v1

0. Introductory Material

0.1 List of Definitions

- Garbage: Residential waste.
- Collect/Collection: The act of picking up garbage.
- Resident: A homeowner who has garbage to collect.
- Collector: An employee of a waste management company, tasked with collecting garbage
- Admin: A user in charge of an instance of our software.
- App: The Android mobile application component of the project, including both the Resident and the Collector functionality.

1. Introduction

1.1 Acknowledgement

Goce Trajcevski - Idea creator and IoT mastermind

1.2 Problem and Project Statement

Problem Statement: In 2013, Americans generated approximately 254 million tons of garbage. That quantity has steadily increased over the past half-century, shown in the diagram on the right. However, garbage collection techniques have not significantly changed over the same time period, still relying on static routes and scheduling. This antiquated technique does not account for each individual's unique waste disposal habits, creating inefficiencies where one resident's garbage is picked up too often, while another's garbage is not picked up enough.

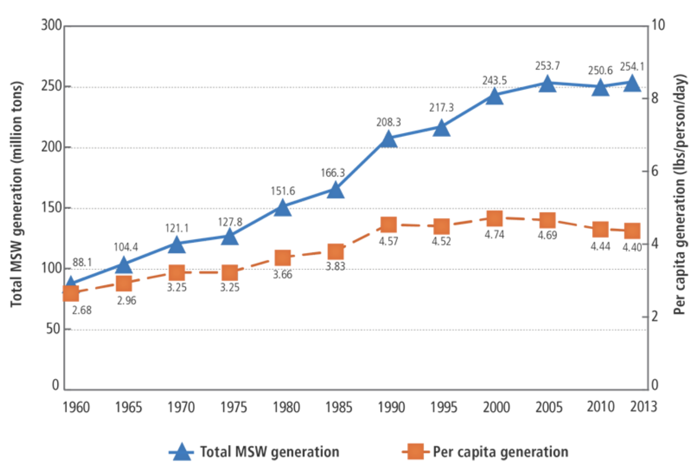

Fig 1. US garbage generation over time.

Solution Approach: Our solution is to add a small sensor package to each resident's garbage can. The sensor package routinely monitors the container's waste level, weight, and location. The sensor package then transmits the compiled information to the cloud. The information received in the cloud is stored in a database. Each night the application will take the garbage bin information to build clusters of nearby bins that need to be picked up and build fuel efficient routes to pick up the clusters. This means that the garbage will be picked up when it needs to be as opposed to being on a fixed schedule. Our application will give users a better garbage collection experience, allow waste management to better allocate their limited resources, and be better for the environment.

1.3 Operational Environment

The environment of our mobile app is a mobile phone. We are planning on supporting Android 4.0 and later. We will also require our app to have access to the internet.

Due to the large range of environments that smart garbage containers will be exposed in, the sensor package installed on garbage containers will need to be water and dust resistant. The sensor package is also expected to perform in extreme temperatures due to the variability of climates in the United States. The IEC standard 60529, commonly known as the IP Code, will be used to standardize the weather rating of the enclosures. Since the product will be outside for potentially long durations, the smart garbage container will be designed with an IP66 rating. The sensor package will also be designed to withstand other environmental threats like vibration, shock, and salt-water.

The initial design for the sensor package uses a solar panel to recharge its battery. Since the garbage will only be guaranteed to be outside for collection days, the sensor package must be able to function using only the amount of power generated in that time.

1.4 Intended Users and Uses

Our platform aims to make waste collection more enjoyable and efficient for both residents and service providers.

Residents

Residents will use one of the platform's mobile applications to communicate with their service provider and gain further insight into their waste behavior. The resident's mobile application will notify them when they should place their garbage bin on the curb for their scheduled collection day. The user could also alert their service provider if their garbage bin needs to be collected as soon as possible. Additionally the mobile application could have the ability to show each resident their individual trends for their trash generation, allowing the customer to gain additional insight into how they throw away trash.

Collectors

Garbage collectors will use the platform to develop better logistics and planning for their service. Garbage collectors will be able to use our product to generate optimized routes that minimize drive time and maximize efficiency. The app will also help garbage collectors discover trends in their customers behavior. This new data driven approach will allow collectors to allocate resources more efficiently throughout the year by giving them the ability to forecast using historical data. This will enable them to give workers more time off during slow parts of the year by running fewer or less frequent routes. They will also be able to ensure that there will be enough collectors on staff when trash generation is expected to peak.

1.5 Assumptions and Limitations

We assume the following about the usage of our platform:

Assumptions

- We assume that Residents will have access to a cell phone with Android 4.0 or later.
- Residents will be willing to participate in our platform,
- Residents will not want to charge their garbage cans and/or may be unable to provide traditional mains power where the garbage cans are stored and used.
- A waste management company will only collect in a single city
- All residential areas where the product will be used are covered by a cellular network

Limitations

- Our platform is designed for “traditional” residential areas: blocks of houses with a grid of interconnected streets.
- We will only be able to produce good routes: producing the best possible route may be beyond computational resource limitations.
- Sensor package form factor will be designed such that it easily fits on the lid of a garbage can.
- Sensor package recharges on its own without customer interaction.

1.6 Expected End Product and Deliverables

We will be confident by the end of the first semester that our garbage bin sensor device is capable of communicating with Amazon IoT via MQTT messages over an LTE Cat M1 connection. After the first semester we will also have an understanding of the sensors needed to build the garbage bin sensor. These sensors will be responsible for measuring the trash height, weight, and the location of the trash bin. Once we have selected the types of sensors we want to use, we will start selecting parts that could be used for a finalized proof of concept device. This means each component will be capable of withstanding our environmental requirements and use the minimum amount of power possible.

The second semester will be assembling the selected components into a finalized fully functional and ruggedized prototype that could be used to demonstrate our idea to potential clients or investors. This device will be able to satisfy all the requirements established in our design document and project plan, including the ability to be attached to a standard garbage bin, delivering trash and location data to the cloud, and demonstrating the non-invasive charging methods that will keep the device always powered on.

2. Specifications and Analysis

2.1 Proposed Design

The mobile application will consist of different Activities, which can be accessed through the following pattern:

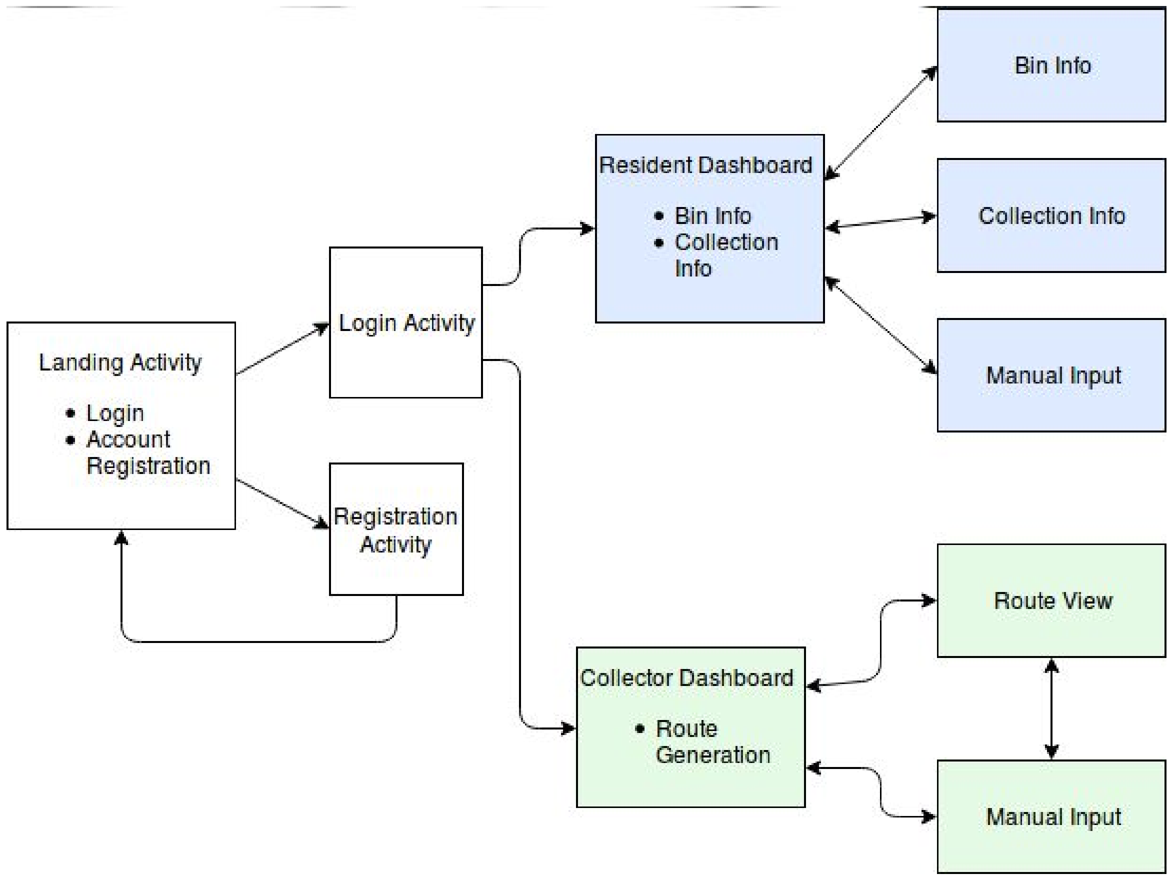

Fig 2: Navigation of the Android Activities.

As of right now we have developed an initial idea for a sensor package prototype. The sensor package will include four individual sensors that detect location, lid movement, trash height, and trash weight. These sensors are connected to the PyCom FiPy, which houses the MCU that controls the device. The PyCom FiPy comes equipped with a 4G LTE-M modem which is capable of transmitting low bandwidth, high range messages using very little power. The PyCom FiPy and sensors will be powered by either an array of supercapacitors or some type of rechargeable battery. The capacitors or battery would be charged by a solar panel. Some type of circuit will be built that manages charging of the battery or capacitor and regulating the voltage to the PyCom FiPy.

The parts to build the initial prototype have been placed on order. Once the parts arrive we will begin testing each of the components individually. For example an infrared proximity sensor and an ultrasonic sensor have been selected as possible sensors for detecting trash height. The sensor that performs best in testing will be incorporated into the proof of concept device. Once all sensors are selected. Software will be written to interface each of the sensors with the FiPy.

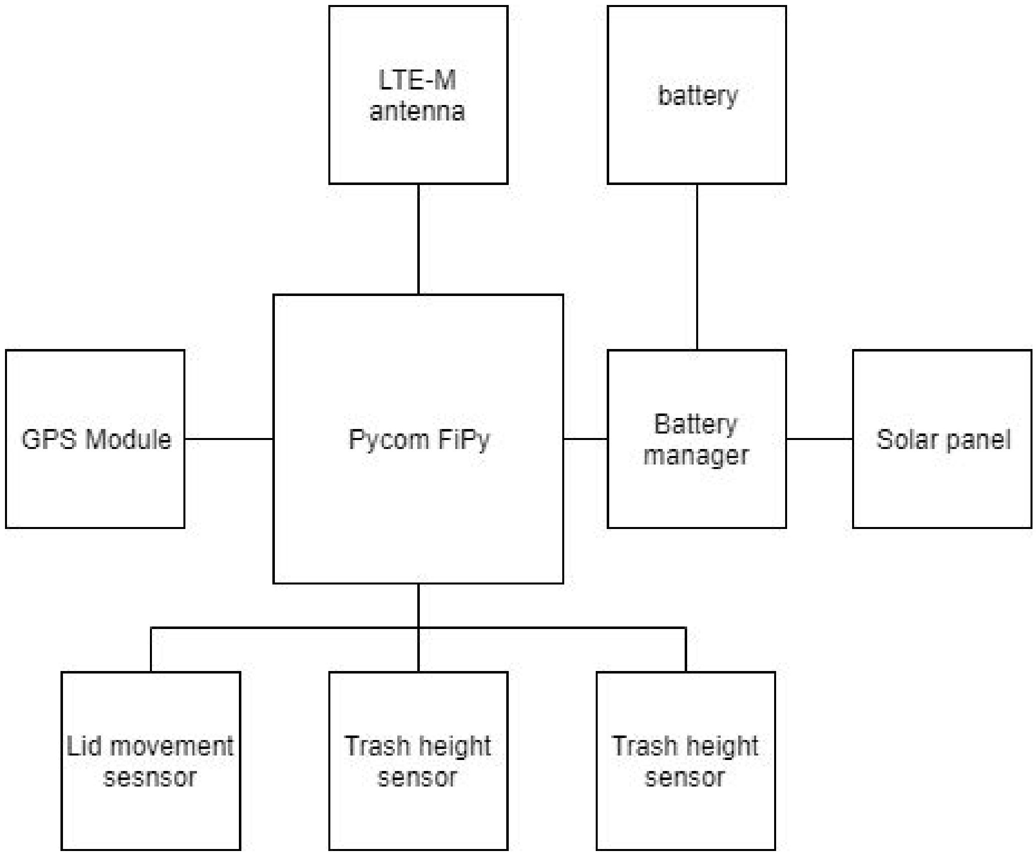

Fig 3: Abstract diagram of hardware components

After the sensors have been interfaced, the software to implement the flow diagram shown below will be written. Software to connect the FiPy with the cloud via MQTT will also be written as well. This software will have the following flow. In order to reduce power consumption the device will sleep until it is woken by the accelerometer detecting the lid being opened. The device will then wait until the lid has been closed for a minute until taking a reading and relaying that information to the cloud infrastructure. If the device does not detect the lid closed after a long time, it will send a "lid open" message to the cloud infrastructure so that the end user may be notified.

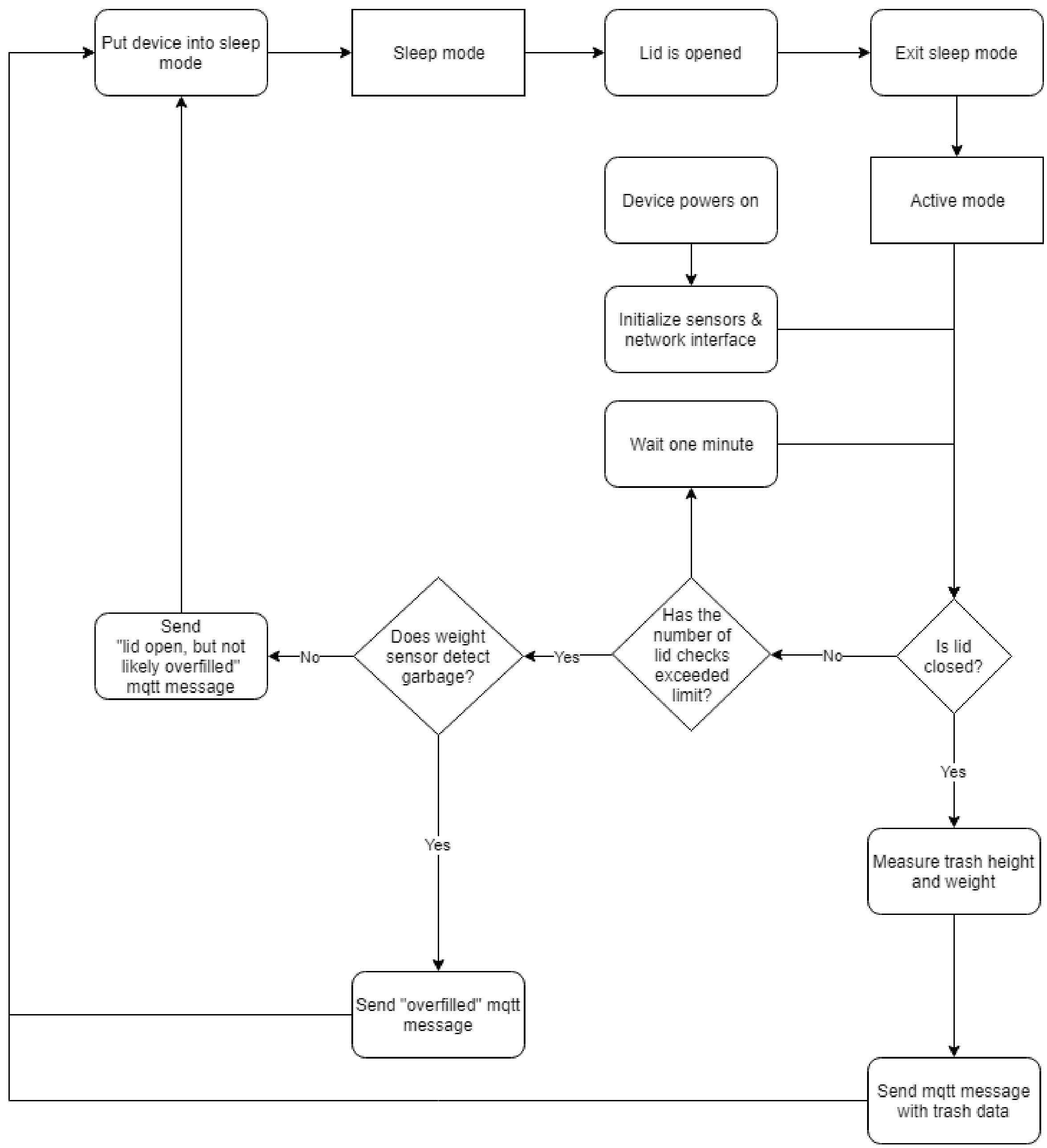

Figure 4: Sensor package flow

2.2 Design Analysis

So far our initial testing of the components for the garbage bin sensors has shown that ultrasonic sensors are much more feasible for trash height measurement compared to infrared proximity sensors. Infrared proximity sensors output an analog voltage that is non-linear as distance is increased or decreased. This means that a complex equation must be used to calculate the distance based of the analog output. Also infrared proximity sensors have a narrow field of vision and are sensitive to the materials their light bounces off of. One possible option would be to place an array of sensors along the side of the garbage bin, however this would not be easy to install on garbage bins currently in use. Testing showed that ultrasonic sensors can measure distance accurately over a wide field of view. Ultrasonic sensors are also indifferent to the optical properties of the garbage inside the container.

As shown in the image below, load cell testing has shown that the output is very noisy. This will make it difficult for the ADC to read an accurate value for the current weight of the garbage can. This leads the hardware team to conclude that we should test building a low-pass filter to reduce the noise output from the sensor. This load cell is specified as 1.5mV/V at max applied load. A simple differential op-amp circuit will be created to assist in amplifying the signal to typical 0-3.3 ADC input levels.

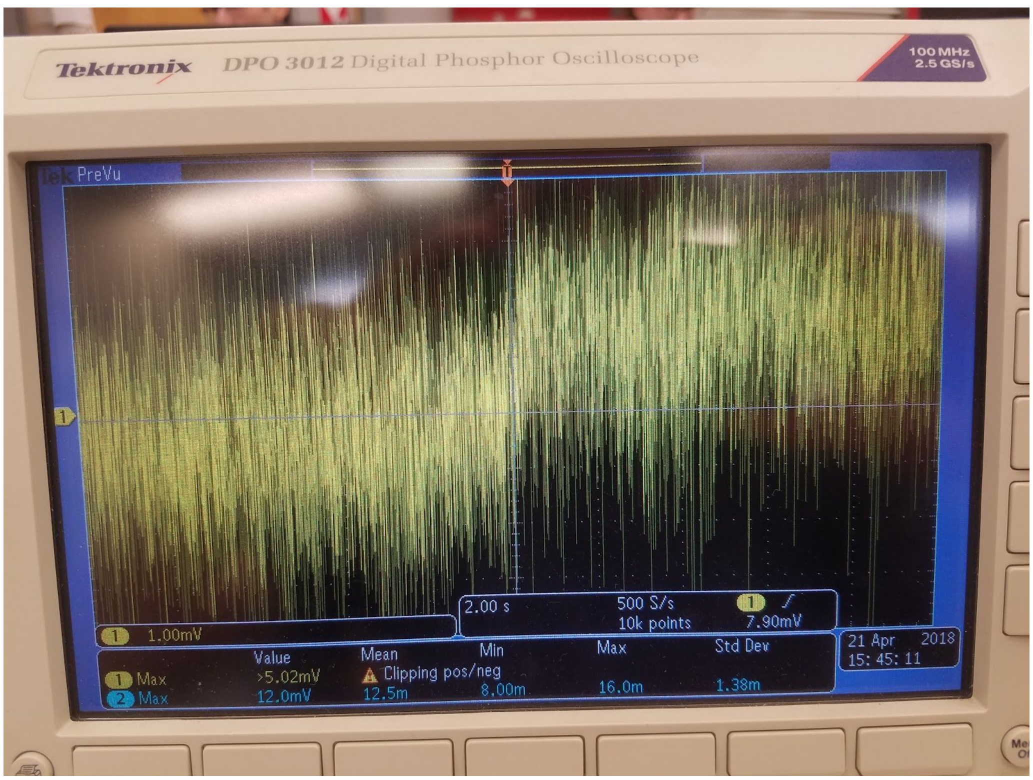

Figure 5: Load cell output with no amplification and 10v excitation voltage. Force was applied to the load cell at t=0 in the center of the screen.

Testing the communication between the microcontroller and the cloud showed that our current LTE-M modem development board, the PiCom FiPy. has issues communicating with the AT&T network. However we found that secure MQTT messages could be sent from the device to the AWS IoT gateway via WiFi networks. Our results suggest that we need to find an LTE-M development board that is capable of using AT&T's network so we can continue building our prototype. Since the only LTE-M development boards on the market use MicroPython, most of the code written to test the Pycom module can be used on other development boards.

3. Testing and Implementation

Our project is a full stack application that integrates hardware parts with a mobile app. As such, we will need to test both the hardware and software as well as the integration.

Hardware:

The hardware will consist of a device mounted to the lid of a standard residential style garbage can that will be able to detect the distance to the bottom of the can by some sensing means that will be determined after testing a few candidate sensors. A weight sensor may also be attached to the bottom of the can such that the weight of the trash can be used in decision making. These sensors will be connected to a microprocessor with radio peripherals that will be able to transmit the state of the weight and height of garbage to our server side infrastructure.

Our hardware testing plan is as follows:

The sensor elements will be individually tested using calibrated laboratory equipment available in our labs to very their function, calibration and suitability. This tentatively includes tests for:

- Distance sensing using a LDPE and paper targets every 15 centimeters from 0 to 2 meters and measuring the sensor output using a multimeter.
- Weight sensing using known weights from every 5kg from 0 to 100kg and measuring the sensor output using a multimeter
- Switch testing using an oscilloscope and a power supply to verify digital input use suitability and switch bounce characteristics

The energy harvesting and storage system will be tested using a dummy resistive load that will simulate approximate load of our actual system after testing of the sensor system yields a current consumption information for all device states. This resistive testing will generate storage capacity rating and approximate usable lifetime figures. The test will pass if it can retain more energy than required by our worst use case.

The remaining piece of the hardware to be tested is the microcontroller which in this case has numerous radio interfaces. LoRa, LTE-M, SigFox and WiFi. Not all of the capabilities of the radios may be tested individually due to lack of receivers to test them with and with the large expense in both time and equipment it would require to test them in a controlled manner. However these radios will be covered by integration testing and WiFi in particular will be exercised extensively during development.

We currently do not have any intention to do accelerated lifetime testing or combined environmental testing due to the proof of concept nature of this project. However we would advise that prior to a product being commercialized that these types of tests be performed.

Software:

For software we will be required to test in four major areas: The algorithm for clustering garbage bins, the route creation algorithm, connectivity between the mobile app and the server, and general usability. The clustering algorithm will be tested by manually creating datasets that range from very easy to hard to cluster and comparing the performance of our algorithm with the optimal clustering. For route creation, we will randomly generate clusters of garbage bins that need to be picked up and compare the routes generated to already existing garbage routes and other algorithms used to generate shortest paths. Testing connectivity will be done by simulating use cases on the mobile app and assuring that the users will be able to do them. General usability will be tested by having people with varying levels of technical literacy attempt to use some function on the app. We will also periodically check user satisfaction with a quick survey screen.

3.1 Interface Specifications

Interfacing between the hardware component and the software system will be accomplished through MQTT messages on Amazon Web Services' WebSocket Protocol.

We decided on MQTT (specifically MQTT-SN which is just MQTT but for sensor networks) as it is designed to function well with low power usage which is a main concern in our project. With low power usage in mind, MQTT supports an offline procedure which will save power by buffering messages while in sleep mode and then deliver them upon wake. Since we also only need short bursts of communication between our network of trash cans and database MQTT works great as one of their main designs is to interact well over low bandwidth and short messages.

Our group decided on using Amazon Web Services (AWS) for our server and AWS has a protocol that supports MQTT, which is AWS's WebSocket protocol. Specifically the message broker supports the use of MQTT to publish and subscribe over Signature Version 4 Signing Process (SigV4) authentication.

http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf -Link for reference on MQTT-SN

3.2 Hardware and Software

Garbage bin sensor prototype

- Ultrasonic range finder
    - Determines the height level of trash in bin
- 200 Kg load cell
    - Measures weight of container, approximating the weight of trash inside
- Accelerometer with “wake on shake” feature
    - Detects lid movement to wake the garbage bin sensor
    - Determines if the lid is opened or closed
- GPS Modem
    - Returns current location of garbage bin for bin tracking
- LTE Cat M1 Modem
    - Connects the microprocessor to a cellular network
    - Allows the sensor to transmit sensor values to the cloud
- Microprocessor
    - Interfaces with each of the sensors and compiles their readings into a packet that is sent through the cellular connection to the Amazon IoT gateway
- Battery
    - Stores energy to power the garbage bin sensor
    - Must be large enough to keep device powered between recharges
- Battery manager
    - Manages the charging of the battery
    - Regulates battery input and output voltage
- Charing device
    - Collects energy via solar or inductive input for charging of battery

Software

- Android Version 4.0 or later
    - Framework for the mobile app
- Amazon Web Services
    - Amazon EC2 web server
    - Amazon RDS database

3.3 Process

Hardware components were tested using multimeter to clarify they interacted correctly given the right voltages. The sensors were tested using the oscilloscope and checking the waveform was producing an accurate reading based on how the sensor was suppose to respond to outside sources (ex. IR sensor producing waveform on how far away an object was from it).

MQTT will be tested once the prototype is completed. Once the completion of prototype we will test the code written to have the hardware and software interact with one another. The plan is to have the MQTT communicate with the AWS broker over LTE-M.

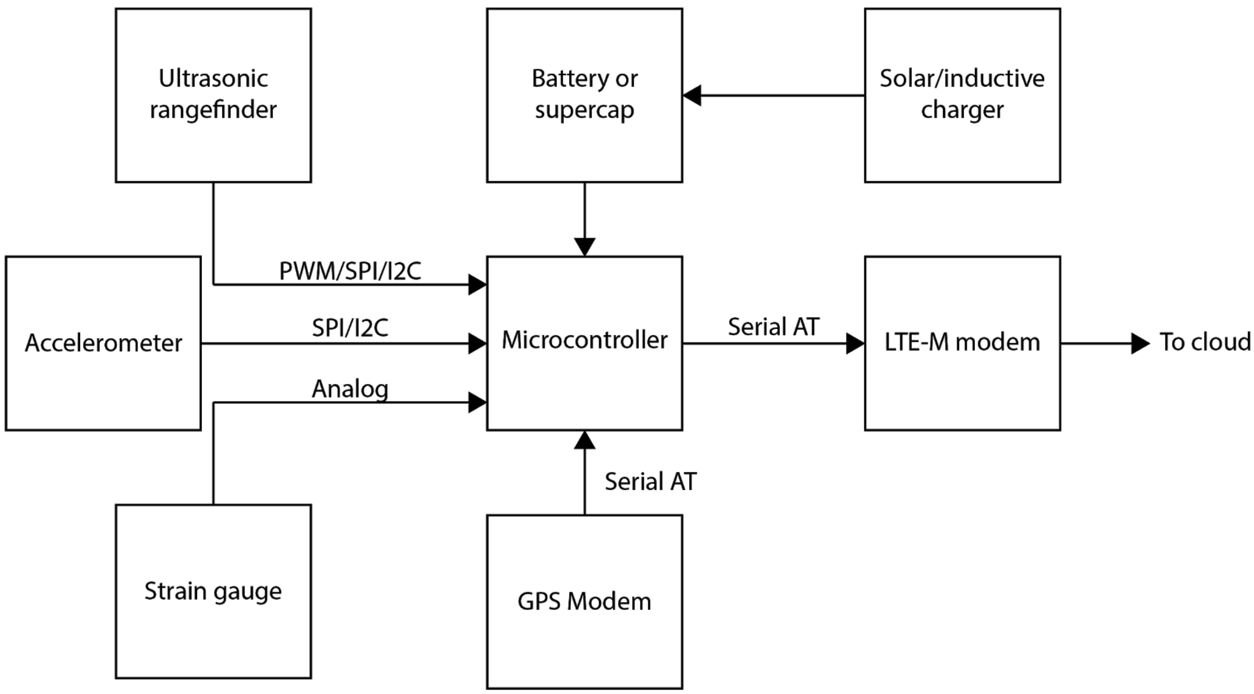

Figure 6: Hardware block diagram

3.4 Results

Implementation Issues and Challenges

Results will be based on the future outcomes of our testing plan.

Testing the garbage bin sensor components gave us an idea of what components we are interested in incorporating into our final prototype. The next challenge will be finding the components that satisfy our environmental requirements and also satisfy any power requirements. We also found that The Pycom FiPy module does not work with the the only LTE-M provider that we have access to. During the semester we saw that other LTE-M development options have entered the market. Thus over the summer and in the beginning of Cpr E 492 we plan to order and evaluate the other LTE-M modem development boards to find an alternative to the Pycom FiPy that will work with AT&T and satisfy all of the specified requirements to be placed on the finalized prototype.

Our current clustering approach has been successful at creating viable clusters that can be turned into routes. In order for a cluster to be viable it must have a path that contains all nodes and be within a range of volumes. For up to 150 node graphs, the clustering code has been able to successfully create clusters such that at least half of the clusters are viable. We believe this to be firm evidence that our clustering solution will be viable when implemented in open street maps.

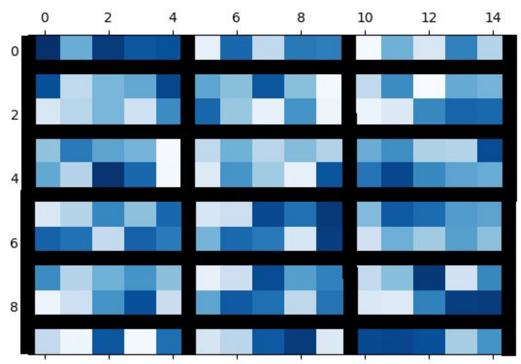

Figure 7: Left 150 Node Graph

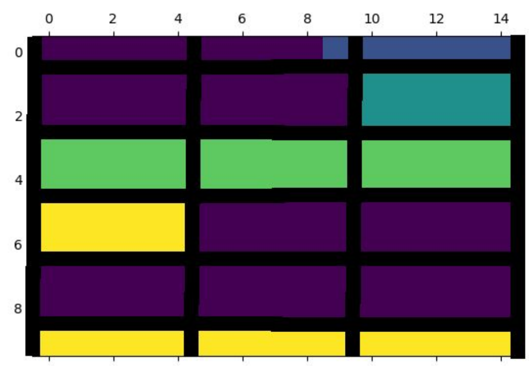

Figure 8: Clusters created from 150 Node Graph

4 Closing Material

4.1 Conclusion

Our project aims to increase efficiency in garbage collection. So far, we have created the designs for our bin augmentation, software systems, and mobile app. We plan to continue improving our

Progress has been made on the garbage bin sensor, but much more work is required to have our completed deliverable by the end of 492. So far we have evaluated what sensors and technologies we want to use in our design, but we have not started building the finalized prototype. The main focus for the hardware team is to immediately start work on building the prototype in the beginning of the fall semester in order to have enough time to have a prototype that has been revised several times to work out any bugs or issues that appear. This means that a finalized parts list will be needed by the beginning of the next semester and circuit design should start immediately along with software development.

4.2 References

[PyCom documents]

AWS IoT services

Environmental Protection Agency, “Municipal Solid Waste,” Environmental Protection Agency, 2016. [Online]. Available: http://www.etsi.org. [Accessed Feb. 12, 2007].

4.3 Appendices

[Bill of Materials]