Household Waste Tracking System
Final Design Report

ECE 499 – Senior Capstone Project

James Triola
Electrical Engineering
Union College Schenectady, NY 12308
Advisor: Water Dixon III, Cherice Traver

March 2016

I. REPORT SUMMARY

The past few years have seen the highest recorded amount of waste produced in America. Despite several options of responsible wasting, landfills are still bearing the brunt of waste deposits. Although recycling is a common practice, it is still not accomplished with maximum efficiency, and composting is even less common in areas where it is not legally mandated. Food scraps and byproducts that could otherwise have been broken down in proper composting methods continue to pile up in landfills where they breakdown into methane gas.

Although there is no easy answer to the complex problem that is consumer waste, the household waste tracking system offers one solution that uses the capitalist system that created the waste issue to incentive our society to rehabilitate our waste practices. By generating large and small data, the system will be able to produce the information needed to create a system of proportional costs for waste. As a result, local governments and waste management companies will be able to provide financial subsidies in the form of tax breaks and bill reductions for responsible waste habits. Additionally, the large scale data accumulated through location tracking will allow a reduction in inefficient waste collection routes, saving companies' money and unnecessary exhaust emissions from the environment.

The household waste tracking system correlates specific quantities of waste to specific customers by weighing the waste in specific bins upon collection. The bins having been embedded with RFID tags will inform the system as to which customer owns the bin. GPS/GPRS services will allow the system to identify the location of each bin retrieval. Finally, all of this information is sent wirelessly through GSM to a web based display that will allow waste management companies, customers, and local governments to utilize the data collected by the system.

TABLE OF CONTENTS

TABLE OF FIGURES

II. INTRODUCTION

The long-term goal of reducing waste has become an ever increasingly conscious goal of our generation, as environmental sustainability and responsibility have become a contemporary societal goals. Despite the increasing prevalence of responsible waste techniques like recycling and composting, waste production has reached an all-time high with each American producing an average of 4.4 pounds of trash per day, contributing to a total national waste production of 254.1 million tons of wastes per year [1].

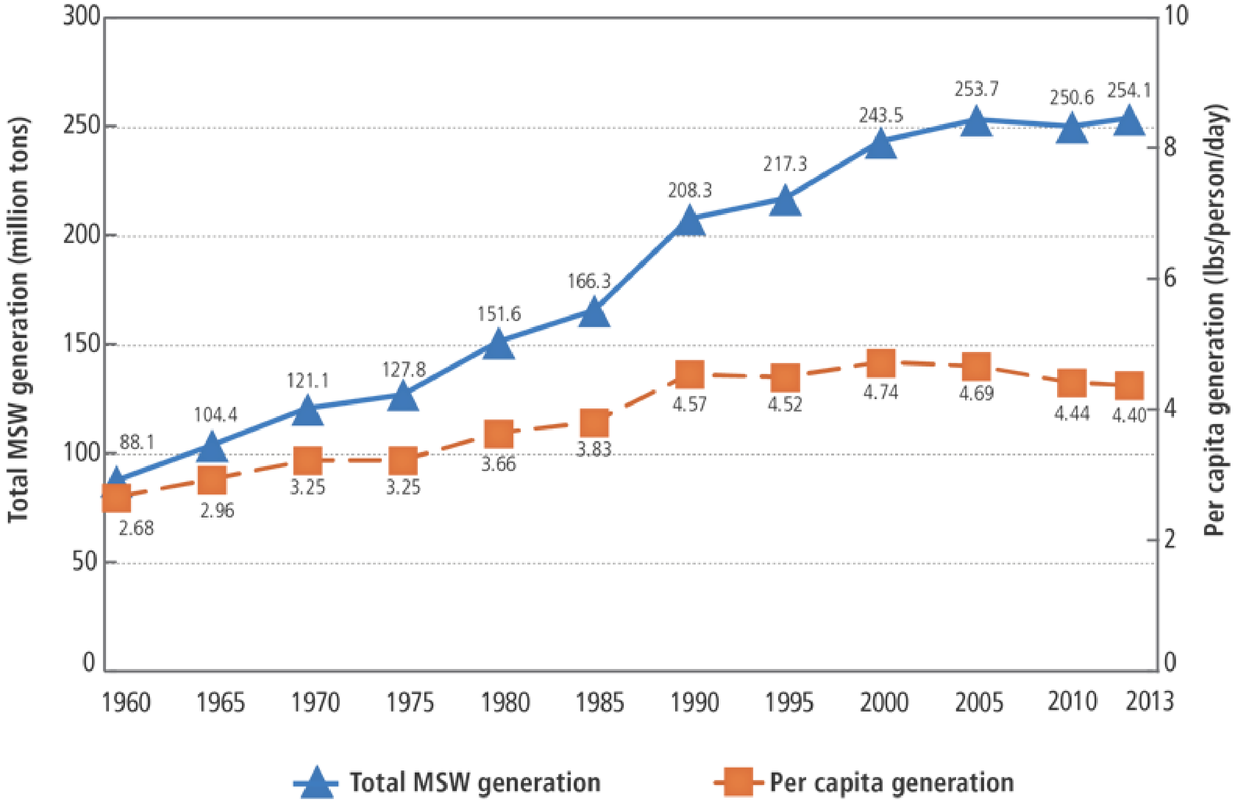


Figure 1 - United States solid waste statistics between 1960 and 2013

http://www3.epa.gov/epawaste/nonhaz/municipal/

Although recycling is a common practice, it is still not accomplished with maximum efficiency, and composting is even less common in areas where it is not legally mandated. Food scraps and byproducts that could otherwise have been broken down in proper composting methods continue to pile up in landfills where they breakdown into methane gas that rises into the atmosphere, where it contributes to the greenhouse effect in an even more dramatic fashion than CO_2. [2] In the plot below, provided by the United States Environmental Protection Agency, we can see that even at all-time high, our country only recycles about a third of the generated waste. [1]

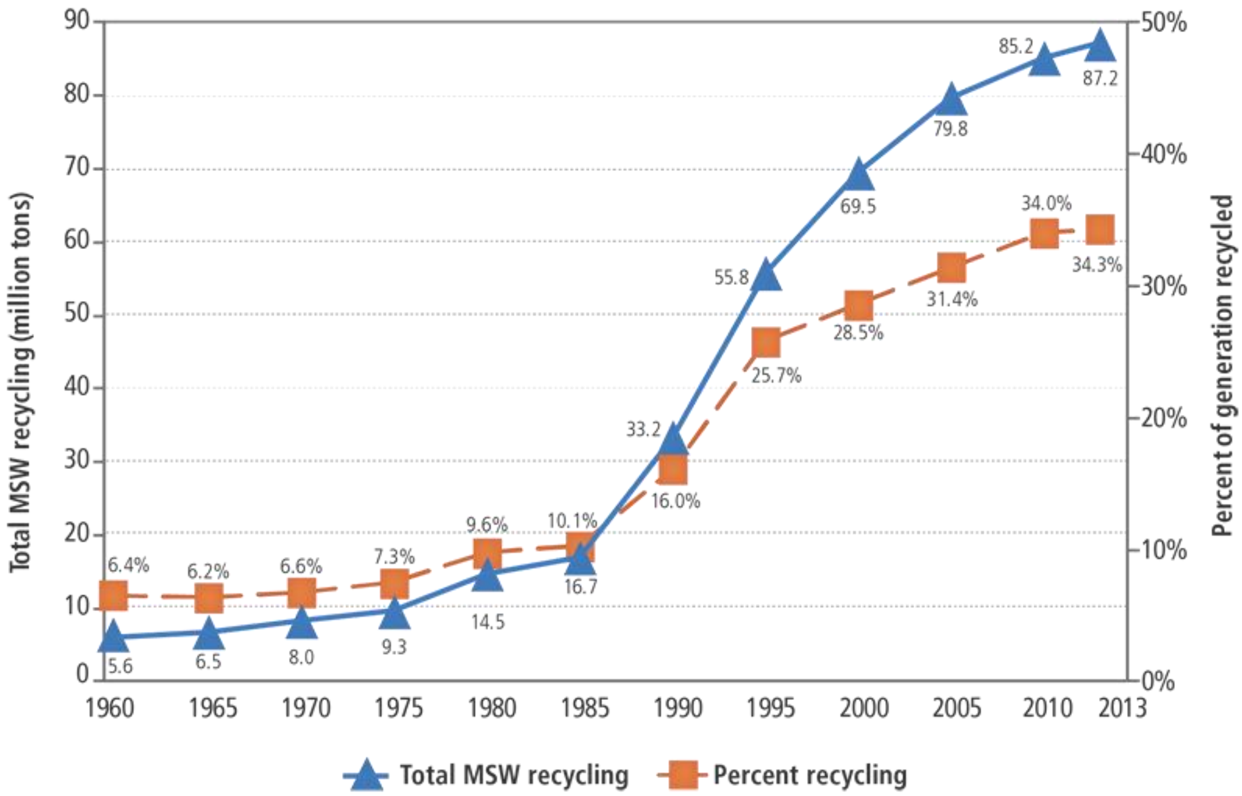


Figure 2 - United States recycling statistics from 1960 to 2013

The household waste tracking system aims to increase the sustainability and personal accountability of our consumer society through the quantification of household waste production. Like so many other facets of our lives, data and the knowledge attained from that data are the keys to reducing the impact of post-consumer waste and moving towards a better future. By implementing a household waste tracking system we will be able to generate individualized and big data that will allow us to incentivize responsible waste, as well as optimize the waste collection process. Using the data generated from individual waste bin collections, local governments and waste management businesses can offer tax cuts, reduced pricing for responsible waste, or other financial incentives and subsidies to stimulate societal motivation. This motivation will cause consumers to not only take serious consideration of waste practices, but also modify greater shopping habits so that market goods become more environmentally responsible as a whole. Large scale location data, when tied with individual waste reception and time data will generate information on waste trends allowing waste management companies to tailor waste collection routes to reflect the habits of their customers. These smart waste collection routes will reduce the number of vehicles, and miles driven to complete the job, ultimately saving the company money and the environment by reducing fuel expenditures.

Although the aim of this project is not that of legislative or financial reform, ideally this system would be used to create a measure by which waste disposal costs are directly correlated to the amount of landfill waste disposed. Federal subsidizing or billing manipulation on the part of the waste management services could easily create a system by which customers become financially responsible for their waste habits. Additionally, this report will not analyze the deeper economic effects of the Household Waste Tracking System. The goal of this design project is ultimately focused on providing a system by which to gather the data that could be used to make these financial political opportunities possible. This report will discuss the background of the project, the design requirements, the history of the project and final implementation, as well as the projects performance, cost and operation.

III. BACKGROUND

Waste collection is standard part of living in the United States. The convenience of the service allows us run our lives efficiently without having to worry about the detail of disposing our post-consumer waste. Unfortunately, with convenience these waste services have led the American people to become blind to the impact of their waste practices. Luckily, in the past few decades, America has been become more aware of the effects of its waste practices and popularized recycling and composting, but these services are still not as widely used as they could be.

Current standard waste management services provide the bare minimum user experience for the average American customer. After signing a contract, the service provider supplies the customer with a number of waste bins that correlate to the approximation of waste that the customer believes they will produce on any given week. Most often this looks like a single land-fill waste bin, a single stream recycling bin that accepts most of the major recyclable materials (glass, plastic, paper, etc.) and, if the customer is lucky, a composting bin. These services may vary, sometimes charging the customer for services other than the standard landfill waste reception, meaning that recycling and composting can even become a financial burden for some customers.

The past few decades have seen the attempt at several systems that would modify this waste monitoring system to optimize the routes, but none have been implemented on a large scale in the United States. Inventors have proposed and patented systems that monitor and categorize the waste a trash truck is accumulating through global positioning services, radiofrequency identification, and video monitoring, setting a precedent for waste tracking through waste bin identification location tracking [3]. This system however did not track the amount of waste in each bin, unlike the household waste tracking system. So it was incapable of creating a system of individual customer awareness, or economic incentives for responsible wasting. There was also been a system proposed that tracks total waste collected throughout a route by summing the total weight of every bin received, which has provided the concept of waste tracking though weight at the retrieval sight [4]. This system could have been used to track big data waste practices but its untimely development in the 1980's meant that the processes and technology required to perform such large scale were unavailable. As well, this system fell short of the capability of recording the waste produced by individual households. The system closest resembling the tracking system purposed in this project is an Apparatus for weight-based charging for waste collection containers, designed by Henry Hornhauer and patented in Germany. This apparatus uses smart cards and weighing device to generate for trash bill correlation. However this device does not use location tracking. [5]

In the new immerging world of the Internet of things, many companies are racing to apply connected, imbedded device technology to municipal waste in the hopes of creating smarter waste collection routes. Smart city companies such as Cisco and Smartbin are using embedded devices to track when bins are full and create optimized single use collection routes.[6-7]

In a world where the household waste tracking device has been implemented, Americans will have become not only aware of their waste habits, but have grown a sense of accountability for waste production. The household waste-tracking device will be able to make substantial effects on the environment, and local economies. Additionally, the sustainability of the system itself will ensure that the system won't end up in a landfill.

The most obvious effect of the household waste tracking system is on the environment. As this system is designed to reduce unnecessary house hold waste, a wide spread implementation of the system would not only aim to reduce the amount of consumer waste that ends up in the landfill, but ultimately aim to increase the amount of recovered materials. In 2010 the EPA found that only 34.1% off all recoverable waste solids were reclaimed through recycling [1], leaving a lot of room for improvement.

If local governments and/or trash companies adopted a system by which responsible wasting (recycling and composting) are subsidized, thus creating an incentive for recycling, consumer waste trends will shift dramatically. By raising consumer waste habit awareness, we will not only choose to correctly sort their recycling, but also seek out products that are environmentally sustainable. This means that the US consumer market would not only create a shift towards a sustainable goods and packaging. In this way the amount of total waste produced by the American people may not reduce, but the waste produced will be much more environmentally sustainable and provide many more recoverable materials. Additionally, with data collected by the household waste tracking system, waste management systems will be able to optimize their routes, cutting down on excess fuel waste, and subsequently burnt fuel emissions. If the tracking system is also applied to compost, America can also reduce large amounts of methane from building up in landfills and releasing in the atmosphere.

In addition to having strong impacts on the environment, the household waste tracking system will also have a strong impact on local and market economies. By subsidizing responsible waste practices, local governments and waste management services can provide cheaper trash services, which could end up saving money for households that are environmentally concious. Accordingly, those who choose not to waste responsibly would end up offsetting the cost of the waste management system by paying higher rates. In addition to shifting the business model of waste management, several third parties would see new business potentials in this system. Similar to the business model of compost companies in Vermont, where law requires composting, enriched topsoil production through compost companies would increase [4]. This would increase the supply of enriched soil, ultimately driving down the cost of enriched soils. This not only makes enriched soil more plentiful and available to American households, but could also decrease the cost of farming in America, leading to cheaper food prices and/or reducing the amount of required U.S. federal agricultural subsidies.

One of the large economic impacts the household waste tracking system will have is directly focused on the waste management service industry itself. As mentioned before, the hold waste tracking system will allow the waste management service to collect data on who and where trash is being collected from. Since the Household waste tracking system collects GPS data on the location of waste retrieval, waste management services will be able to optimize the routes of their collection service. By optimizing their routes, these waste management companies will significantly reduce the amount of fuel they need to purchase for their service.

The household waste tracking system will also have an economic impact on the product packaging industry. The demand for sustainable packaging will increase due to the subsidization of responsible wasting. Accordingly, environmentally responsible packaging production will have to increase to meet the demand as to not drive up cost.

In addition to creating sustainable practices, the household waste tracking system will be a sustainable device in its self. Since the device only requires to be retrofitted to existing waste infrastructure, and the device itself has no moving parts it will likely have a long life, even though it will see a lot of use. The components most likely to fail and need replacing are small and require minimal knowledge to replace.

The largest sustainability concern of the system will be the implementation of tracking tags. The tracking tags of the system will need to be fitted directly on to the waste bins themselves and in the case of failure or damage will need to be disposed of and replaced. Since these tags are composed of small pieces of plastic, large-scale failure of the tags could generate a lot of waste. However, a expanded up later in this report, other forms of “internet of things” based solutions to consumer waste could end up generating massively larger amounts of infrastructural waste due to the sheer amount of devices required to implement the systems.

IV. DESIGN REQUIREMENTS

The household waste tracking system will give waste management services the ability to catalog the net weight of waste in a particular waste bin at the curbside during the typical residential trash truck pick up service. Simultaneously, the system will identify the type of bin, categorizing the type of waste (landfill, recycling, compost, etc.) and then tie that waste with a specific customer and location. This data will then be sent to a database where a company can relay, store, and convey that information as they see fit. Ideally, this system will have as little impact on the current trash collection process as possible, as to alleviate the waste-management service workers and household customer of any additional responsibilities.

The most basic design requirement of the system is simplicity. A system that is capable of generating and collecting this magnitude of data will need to have a very simple implementation, so that waste management service worker will need to acquire little to no more skills in order to use the system. If the system is too complex or cumbersome, it runs the risk of becoming unpopular and ultimately misused. Accordingly, to attain the maximum simplicity, the system will need to be automated. An automated system naturally lends its self to a microcontroller-based design. Additionally, to simplify the design, the system must not modify the existing waste management protocols and service technique. This also means that the system must retrofit to existing waste management infrastructure, i.e. trash trucks. Likewise, the system must be either dependent on the truck as a power supply or power independent.

Figure 3 - Household waste tracking system retro fitted to waste reception vehicle

As the primary function of the household waste tracking system, data accuracy and bin identification fidelity are of the upmost importance, as to minimize the risk of incurring extra costs on the users of the system, and ensure the most utility possible. If an inaccurate waste amount is recorded, the data collected the system will prevent waste-management services and local governments from providing the correct incentives for responsible wasting. Furthermore, if the identification of the bin is inaccurate or unidentifiable the waste-management service could end up losing money and the ultimate purpose of the system is defeated as the data will not be of any use. The household waste tracking system must have a weight tolerance of 84 grams. This tolerance is the average one-cent unit weight of waste management service; calculated using the average amount of waste produced by American households and the pricing average of waste management services (see appendix). Past this accuracy, differences in weight will have less than a cent’s worth of impact on waste management service fees.

Additionally, once data and identification is collected the system must be able to relay or save the data in a way this is useful for the customer, waste-management service, and local governments. This means that data cannot be lost, so it must either be transmitted from the device or stored so that it can be off located at a different time. The information gathered and transmitted by the system will also need to be displayed in a convenient way to provide the most use. Although the system will provide information that will help waste-management services optimize their routes, the data analysis aspects that enable these functions are not a part of the household waste tracking system and will be the responsibility of the management service. The main function of the household waste tracking system is summarized by the figure below.

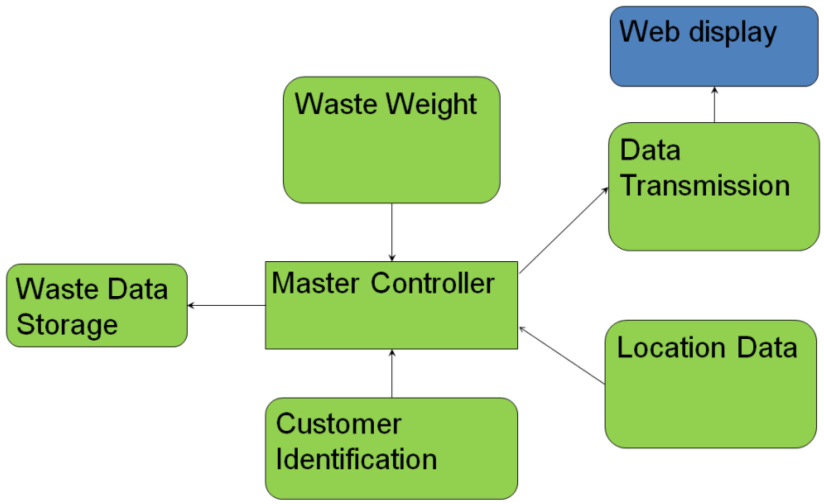


Figure 4 - Final top-level functions block diagram

Due to the logistical and financial restrictions of this project the system will need to be testing on a scaled model rather than a full sized trash truck. Accordingly, the system will need a few small modifications to fit the new project model. This means that the technique for weight measurement will need to be slightly modified form the full-scale implementation. Additionally, the power system of the prototype will need to be independent and most likely battery oriented. Another modification to the system that must be made, due to the scale of the prototype, is user feedback. Since the dumping processes of waste collection will not be automated in this project, the device will need to give feedback to the operator as to when and how to dump the waste in cooperation with data collection.

Instead of implementing a unique web service for the household waste tracking system, this implementation used two preexisting "Internet of things" platforms to perform the functions of the system. To pass information between the device and the web display, the system uses

Dweet.io Dweet is great at storing small amounts of variables (including integers, float values, strings, and locations) for web enabled “things,”, because it only requires a simple Http post to store data. Once the data has been stored on Dweet, another “Internet of things” platform called Freeboard.io automatically gathers the information and displays it in a way that is simple for customers, waste management companies and local governments to use.

V. DESIGN ALTERNATIVES

There are several different ways that household waste tracking could be accomplished. One of the largest schools of thought around household waste data generation is through the "Internet of Things" [6-7]. The "Internet of Things," to provide a loose definition, is the concept of embedding network enabled microcontroller-based devices into physical objects in order to produce large quantities of data. The astounding concept here being that the data generated will have been generated and input all without the aid of humans, and yet will be used to help optimize the lives of humans. The vast amount of information that these devices produce can then be used to automate and optimize our lives. Although the design purposed in this project could still be classified as an "Internet of Things" based solution to the problem, the design differs greatly from "Internet of Things" based solutions produced by many of the major smart city companies today.

Cisco, and Smartbin as well as many other companies have purposed systems that track waste bins through embedded devices within the bin its self. These systems would ideally supply every bin with a microcontroller, a way to detect trash (such as an infrared detector or ultrasonic range finder), and internet connectivity through Wi-Fi. Then, on a day when waste is to be collected, the bins would inform the collection service of its status, and the optimal waste collection route would be generated [6-7]. This method is, by any comparison, the best way to automate waste collection routes, as it provides real-time data for optimization rather than relying on trends and probabilities. However, the downsides of this approach to waste tracking are the ambiguity of the waste loads and the scale of infrastructure required to implement this solution. Since the waste in each bin will be of varying density there is no way to give an accurate representation of how much space each bin will take up on a collection vehicle.

Additionally, that information could not be used to correlate amounts of waste produced with specific waste producers. If every bin requires a device, millions of embedded platforms, sensors, batteries and charging stations will be required to keep the system running. Ensuring internet connectivity for every single bin will be a nightmare, not to mention the maintenance required to keep every bin online. Furthermore, the environmental cost of producing all of the devices would be very heavy.

The household waste tracking system will not be able to communicate the status of an individual waste bin but rather will rely on big data collection to help optimize waste routes. Additionally the household waste tracking system, due to the much more limited number of sensors and passive communication elements will require much less power and a much less complicated system of infrastructure. Accordingly, the minimal amount of infrastructure involved in the household waste tracking system will reduce the number of failed electronics and generation of electronic waste. With the internet of things approach to waste generation, the need for replacing electronic sensors will drive up the total system cost and contribute to the growing problem of electronic waste.

In addition to different conceptual approaches to data collection, there are also several alternative component implementations that could have manifested the household waste tacking system. Several of the largest design decisions that lead to the current prototype were the micro controller, the method of data communication and, and the method of bin identification.

The three main considerations when choosing a microcontroller were ease of use, the ability to handle all of the subsystems specified by the design requirements, and ability to prototype. Ultimately, the two main systems that would fulfill these requirements were the

Arduino Mega, and the Raspberry Pi. In the end either system would have worked just as well for this project, but the Arduino offered easier implementation for the sub-systems, and a wider library of established programming.

Table 1 - Comparison of microcontrollers

Name	Processor	Operating /Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
Arduino Mega 2560 [8]	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Arduino Uno [9]	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
Raspberry Pi[10]	ARM Cortex-A7	5 V	700 Mhz	0	26/2		512000	Adjustable SD	Micro	1

As opposed to the initial design of the system, the Arduino mega will serve as the master microcontroller of the system. As there are four devices that need to interface with the board, and multiple user interaction LEDs needed to connect the system, only the mega would ultimately provide enough connectivity. The mega provides a much larger platform on which to construct the household waste tracker. The Raspberry Pi was also a major contender for the device master but ultimately the prototyping capabilities of the Arduino, along with its simple modular and coding implementation give it an edge. Although the Raspberry Pi provides a greater computing power, an onboard operating system and greater memory, the simplicity of the Arduino and modular additions more closely suit the caliber of the project.

The next big consideration of the system was the method of data communication. Initially this system had been designed to communicate over Wi-Fi and used proximity sensing to initiate data communication. Considering the difficulty of implementing Wi-Fi and proximity sensing, GSM communication paired with back up SD data storage was implemented. GSM offers the advantage of being accessible anywhere within a data network, and when GSM is unavailable; the SD data storage offers a simple way to store information until the network can be accessed again. Additionally a GSM board allows the system to have redundant location tracking for identification verification and waste trend analysis.

Another major design decision of the project was the method of receptacle identification. Of the currently employed to protocol catalogue trashcans there are two primary methods. Radio frequency identification or RFID allows a high frequency reading unit to send a series of inductive pulses at a tare that reads a similar frequency. These pulses cause a loop of wire within the tag to power an integrated circuit chip within the tag to respond with an identifying signal back to the reader. This method of identification has proved useful within a wide variety of industries where cataloguing and tracking vast amounts of items is paramount. The other major method of item tracking is barcoding. This method is a standard throughout the consumer industry as a way of tracking purchases and quickly identifying information about an item. This method of identification relies on a laser beam that is scanned across the surface of a bar code, a series of dark and light lines. As the laser beam hits a light area of the code light is reflected back into the scanner, and when there beam hits a dark line, light is not returned to the scanner. This series of light pulses creates a binary code to the scanner, which creates a number correlating to a specific product. Ultimately, a comparison shown below revealed that RFID would provide the most consistent and accurate results for my system. Furthermore, the waste identification system that is already developed by Arebey, Hannan, Basri, Begum, and Abdullah's, shows that RFID tracking in this sort of application proved to be the best choice [3]. Luckily Arduino, along with several other third party companies, produces an inexpensive RFID shield that fits to the Arduino prototyping board.

Technology	Pros	Cons
RFID [11] [12] [13]	• Easily Embeddable • High accuracy • Reading is based on proximity not line of sight	• More expensive
Barcode	• Easily retrofits to existing Trash Bins • Low Cost	• Requires Visibility

Additionally, although location tracking, via services like GPS, offers the ability to track the point of reception, they be inaccurate up to a specified tolerance and in the event that a waste bin in not placed in a particular spot, can create inaccurate identification. The household waste tracking system, in its current form, does utilize location tracking but only as a means of secondary emergency identification.

VI. PRELIMINARY PURPOSED DESIGN

Although the main objectives of the household waste tracking systems were fundamentally the same as there are in the final design implementation, the ways in which the system functions differed greatly. As initially purposed an Arduino prototyping microcontroller would create the best brain and processing unit for the device. The trade study in the design alternatives section revealed that the prototyping capabilities of the Arduino, along with its simple modular and coding implementation give it an edge over the other popular prototyping microcontroller board, Raspberry Pi. Although the Raspberry Pi provides a greater computing power, an onboard operating system and greater memory, the simplicity of the Arduino and modular additions more closely suit the caliber of the project. Although the final implementation would use an Arduino initially the system was to run on a Uno model, not the Mega. The Uno ultimately has too little memory and the number of ports required for the system would have stretch the connectivity limitations of the Uno.

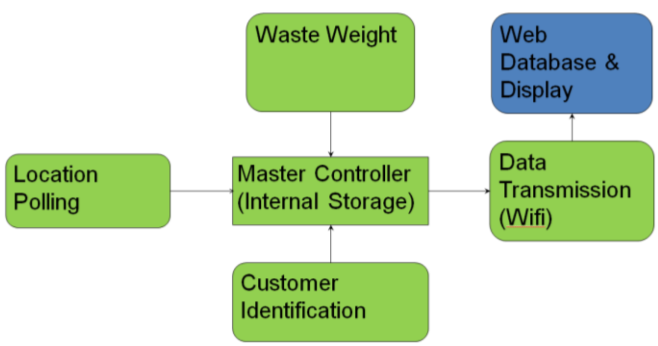


Figure 5 - Initial top-level function block diagram

The next major design decision of the project was the method of receptacle identification. Like the final implementation of the system, the preliminary design of the household waste tracking system used radio frequency identification (RFID) to identify waste bins. This technology proved to be far superior to other forms of identification, as explained in the previous section of this report.

The first iteration of the system utilized an electronic load cell just as the final implementation. Load cells are an efficient and accurate method of converting physical weights into an electronic signal, so they are a natural choice for measuring the amount of waste inside a bin. Each customers waste can be measured by simply taking the net weight of a waste bin before and after collecting the waste. Unlike the final implementation, the original system proposal suggested that the load cell should just be plugged directly into the analog input pins of the Arduino. Ultimately this was a bad choice, as the load cell produces a differential voltage that only varies by a few millivolts, which translates into a very low accuracy measurement when read purely by the Arduino. The final implementation would ultimately require an additional A/D converter & amplifier to produce useable data.

Data transmission on the preliminary design of the system required Wi-Fi and storage. As the waste collection vehicle returned to the processing plant, data would be uploaded to servers using Wi-Fi and the system would stop attempting to record the weight of bins. Waste collection data would otherwise be stored on the device while out on a collection route.

As is the case with the final implementation, data collection was to require very little user interaction. Data collection in the preliminary system was to be fully automated by the device proximity to facility Wi-Fi. Identification systems, and the waste dumping process, would have been automated, so the only actual human interaction with the system would have occured after data has been offloaded from the device. The original implementation of the system required the creation of a web-based database that allows the customer to view individual bin data over multiple collection periods. The customer, waste management company and local governments could then use that database to view trends and statistics about the waste collection process. Additionally the system prototype would have been enclosed in a ruggedized encasing to shelter the delectate electronic components from the harsh conditions faced in the waste collection process.

In the preliminary design phase of the system a testing apparatus was not purposed, but later on in the design process it would become apparent that such an apparatus would be necessary.

VII. FINAL IMPLEMENTATION

i. System Overview

The design and construction process of the Household Waste Tracking System ultimately changed the final implementation of the system significantly. Between the first and final designs of the project many things changed but the primary tasks of the system remained the same.

The major differences in the overall design of the system include changes to the micro-controller, the data storage system, the data transmission method and the web interfacing of the project. In the final implementation of the system, waste weight measurement is still evaluated using a load cell, although the system now utilizes an off-board analog to digital converter and amplifier board to return a highly accurate value. The system still uses radio frequency identification (RFID) to identify tags that are embedded inside trash bins.

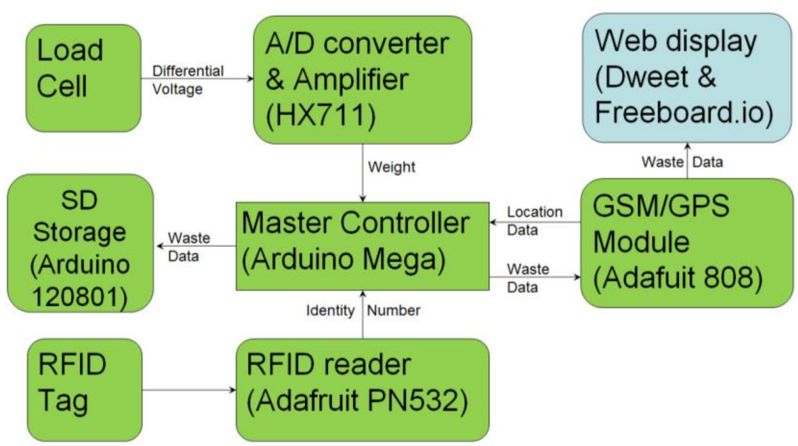


Figure 6 - Top level devices and connectivity block diagram

The waste bin measurement, like the original implementation still occurs through a load cell, but in the final implementation an analog to digital converter and amplifier are used to convert the value from the load cell to a usable value. Since the differential voltage provided by the load cell is very small, in the mere millivolt range, the analog to digital converter onboard the Arduino would provide a very small range of resolution to read the weight, a feature that would end up providing inaccuracies that would cost waste management companies and their customers countless dollars.

Since a primary goal of the final system was to provide location data along with specific customer waste information, the system required a GPS location. With the addition of a GPS/GSM module the system was able to acquire the location of each individual waste reception. In the event that this identification will fail, the household waste cataloging system will record the GPS location of the trash retrieval sight and use that as a reference in determining the identity of the trash and owner of the trash bin, based on historical retrievals of that location. If a bin is habitually unable to be read, the trash company may either service the waste-bin or notify the owner of the bin that they need to replace their waste bin. Although GPS is highly reliable since it's based on highly reliable satellites, occasionally the system may have an internal error making the module inaccessible, or in the event the GPS antenna is covered, the system will still be able to acquire the location of a waste reception through the GSM network, using GPRS technology. Although GPRS locations are not as accurate as GPS location, they will still give waste management companies insight into the amount of waste received in a specific area which can be used for big data.

When weight, identification and location data have all been acquired, the system then stores the data on a Secure Digital (SD) card. This creates a hard copy of all data collected along a route. In the event that the system is unable to wirelessly transmit the data back to a server, this hard copy will allow provide redundancy and reliability for customer information.

Once all customer waste information has been collected, the system wirelessly transmits the information through GSM data to a web based display. The household waste tracking system accomplishes this by using a GPS/GSM break out board. Using a simple Http post the system transmits individual user information to a storage server upon individual waste bin retrieval.

ii. System Implementation – Hardware

As stated before the system master is an Arduino Mega [8]. This Mega uses synchronous and asynchronous serial communication to connect to the load cell amplifier, RFID reader, SD read/write board, and GSM/GPS breakout board. Additionally the system communicates with the operator through a series of LEDs and for trouble shooting purposes USB serial communication. Additionally, as the system is intended to run without the aid of a computer, a 9V battery supplies power to the Arduino, this supplies power to all of the auxiliary devices except the GPS/GSM module.

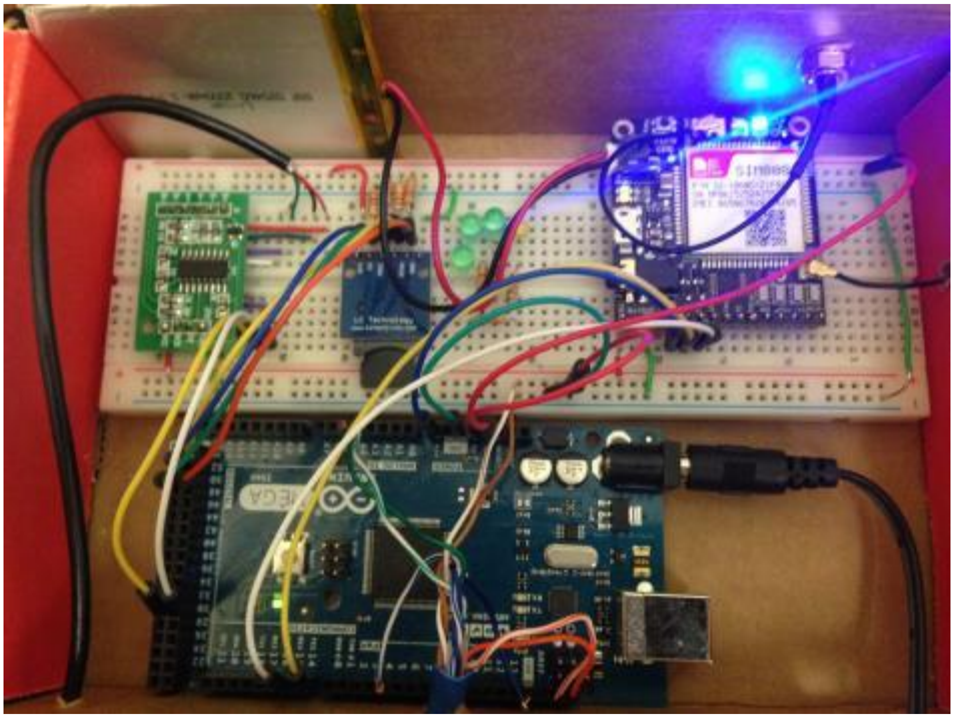


Figure 7 - Constructed device inside enclosure

The RFID board chosen for the system is an Adafruit PN532 RFID/NFC board [11]. This board was chosen because it easily interfaces with Arduino. With minimal wiring, and a simple library for operation, this board was a natural choice. The RFID board connects to the system through Inter-Integrated Circuit (I^2C) communication. Additionally the Mega prompts a card read on the board through an interrupt request (IRQ) which requires an additional digital pin to implement. The RFID board draws its power directly from the Arduino Mega board through the 5 volt power supply and ground. The board communicates with embedded RFID tags through inductance signals at a frequency of 13.56MHz.

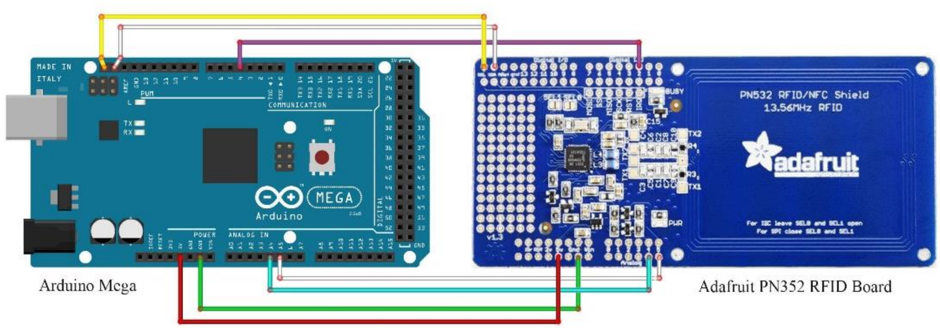


Figure 8 - RFID reader/write board wiring diagram

The load cell itself is constructed out of a metallic disk with a Wheatstone bridge of strain gauge resistors mounted to the disk. As the metallic disk experiences forces, it deforms ever so slightly stretching or compressing the strain gauges that are attached to the metallic disk, skewing their resistances. This skewing creates a differential voltage proportional to the amount of weight placed on the load cell. The load cell used in the final implementation of the system is a Sparkfun 50 Kg disk load cell [14]. This load cell was chosen because the average house hold will produce around 35 Kg with an additional 11 Kg for a standard 35 gallon waste bin. This load cell would need to be modified depending on the waste company, their trash truck receptacle are style, and the customers that they cater to.

The load cell amplifier and analog to digital converter used in this project was a HX711 model [15]. This is a very standard model of A/D converter used for load cells. The HX711 uses a 24 bit converter to turn the differential voltage from the load cell into a useable value, and then communicates with the Arduino Mega through synchronous digital communication. Since the HX711 does not use SPI or I^2C the implementation of this board only requires the use of two non-specific digital I/O pins. The board comes with a preassembled master header file, so there is little more required to set the board up than to calibrate and tare the initial weight value.

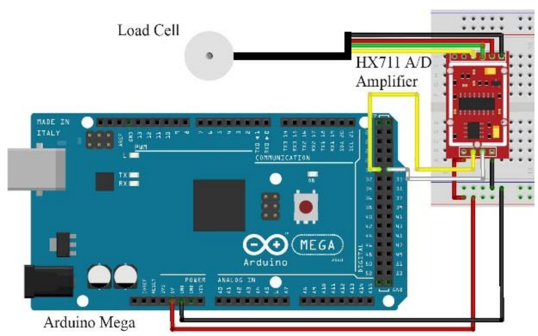


Figure 9 - Load cell & A/D amplifier wiring diagram

One of the simpler systems to implement in the Household Waste Tracking System was the SD read/write card [16]. As Arduino already has a built in SD library, the only real challenge was finding the correct pins to use on the board. The SD card communication uses a Serial Peripheral Interface (SPI) Bus interface so the pins for master-in-slave-out, master-out-slave-in, and system-clock were already predetermined. The only pin left to assign was the chip select, which was assigned to be near the other SPI pins. During construction it was found that the Micro SD board required pull up and pull up resistors (not shown in the figure below) on the communication lines, clock line, and the chip select line.

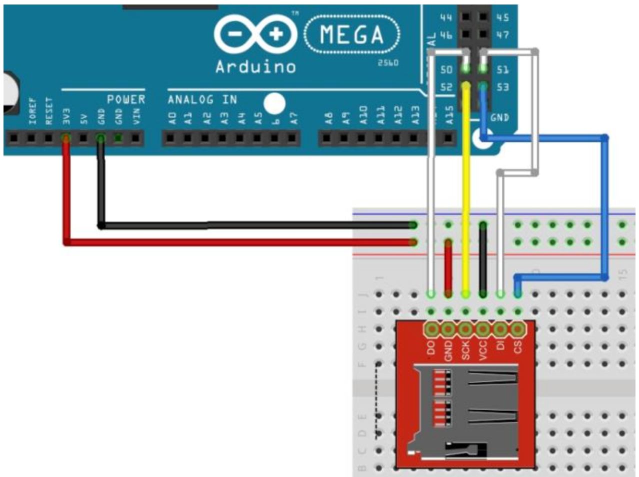


Figure 10 - Micro SD card board wiring diagram (pull-up and pull-down resistors)

One of the largest pieces of the system implementation was the GSM/GPS board [17]. Initially, the system was to be constructed using a Sparkfun MG2639 GSM/GPS board. Ultimately, this board proved to be under documented and ultimately unusable. So, the system was implemented using an Adafruit FONA 808 GSM/GPS board. This system communicates with the Arduino, using the hardware serial UART interface. With the very complete header files and library provided by Adafruit, implementing this system was infinitely easier than the Sparkfun MG2639. Using a prepaid T-mobile cellular plan and Simcard, the system was able to gather GPRS location information and send information using the GSM network and the Http post commands that were documented in the header file. Additionally this board easily gathered location information from GPS satellites. Since the board draws so much energy during use, the system required an additional 2000 mAh, 3.7 to 7.4 V lithium ion battery [18]. In addition, the board did not have internal antennas, so passive antennas were purchased and attached for both the GPS and the GSM connectivity.

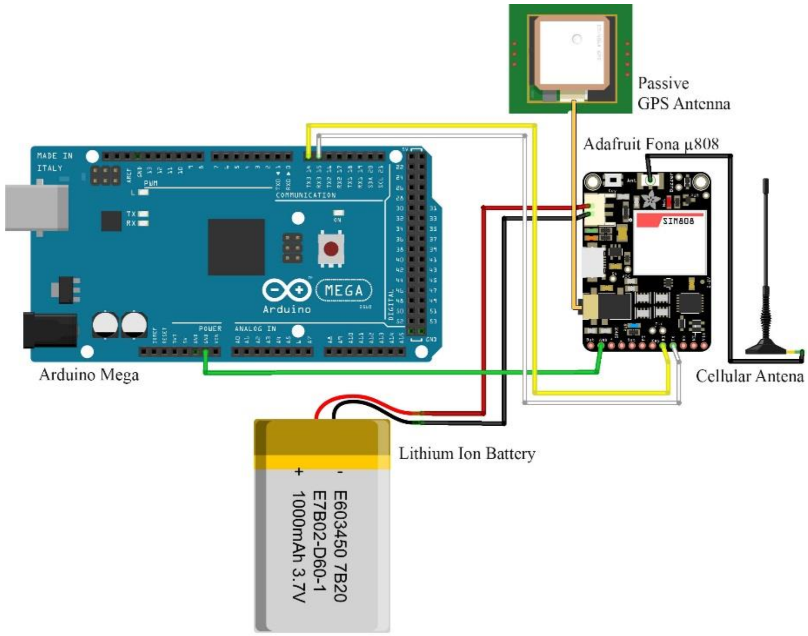
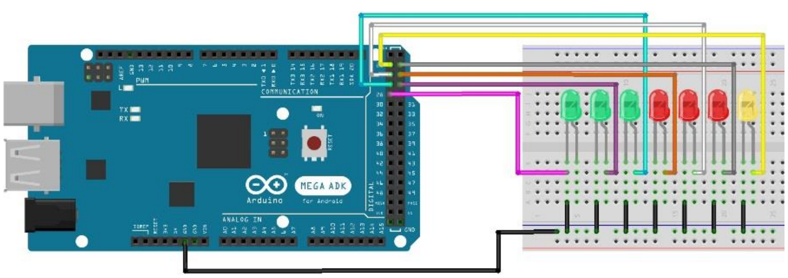


Figure 11 - Adafruit Fona 808, GSM board Battery, GPS & Cellular Antennas wiring diagram

In addition to the main devices of the system, there is one method of user interaction that is a part of the hardware. As the system progresses through a routine waste reception, there is a series of seven LEDs that notify the operator of the status of the system. Not shown in the figures below are the resistors required to lower the current flowing through the resistors.



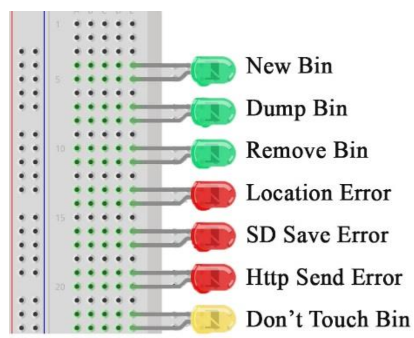


Figure 12 - LED wiring Diagram (LEDs not shown) **Figure 13 – LED designation diagram**

iii. System Implementation – Software

Since the system has a set number of functions that must occur in a synchronous order every time, the master program for the Household waste tracking system was constructed using a simple looping algorithm.

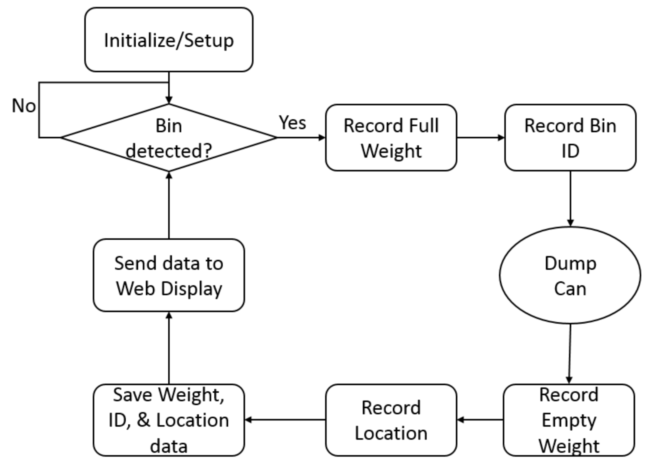


Figure 14 - Overall procedural flow diagram

After initializing and setting up all of the devices and constants in for the program the system starts polling for a waste bin and indicates to the user via LED that the system is ready to measure a new bin. The system then waits for either a significant amount of weight to be placed on the load cell or an RFID tag to be detected. In the event that significant weight has been placed on the load cell but and ID was not detected the system will note that an ID was not found, record the weight of the bin for later use in determining the net weight of the waste, and proceed to the dumping phase. An addition ID check was implemented in the code after the weighing proceedures to check if a previously unavailable ID becomes available after a dumping procedure. In the event that an ID is detected before weight is placed on the load cell, as often is the case with the system, the ID is recorded and then the system waits for a bin to be placed on the load cell. Once a bin is detected, the system records the full weight of the bin, and then proceeds to prompt the operator to dump the waste out of the bin. This procedure is summarized by the figure below.

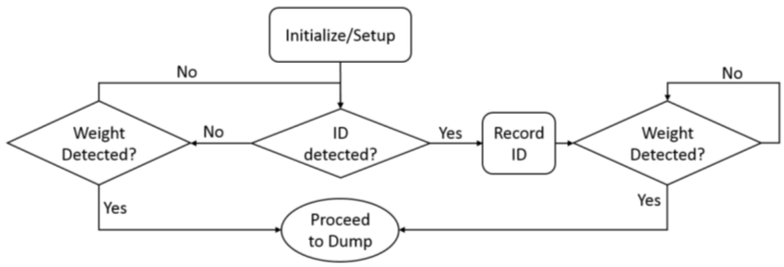


Figure 15 - Bin polling method flow diagram

After the user dumps the bin, the system measures the empty weight of the bin and subtracts that value from the previously established full weight, then prompts the operator to remove the bin via LED. Next, the system prompts the GSM/GPS module to start the GPS connection, and records the location coordinates. If a connection to the GPS cannot be established the GSM/GPS module is prompted to begin a GPRS location search. If neither GPS nor GPRS locations can be established, the system notes that location could not be found and notifies the operator via LED. After location, weight, and ID information have been established all of the data is stored to a micro SD card in a .csv file. If, for some reason, the SD card cannot be accessed, the system notifies the operator through another LED. Finally, the system prompts the GSM/GPS module to begin a GSM connection. All of the data is then formatted into a string to be sent to Dweet.io [19]. The string is then attached to a URL and sent to Dweet via an Http “post” protocol. If the system is unable to complete the Http post the user is notified with an LED. A post from the device is accomplished simply by requesting the URL. There is no formal information that is actually pushed after the URL connection is established. As shown below in the figure, all of the data sent that correlates with a specific bin is embedded in the URL that is called by the http.

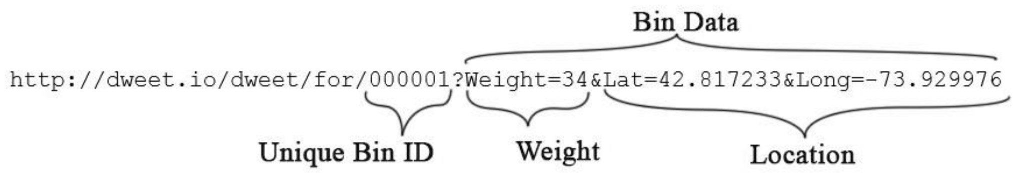


Figure 16 - Dweet.io information post through URL

The household waste tracking system, largely due to the Arduino micro controller, was coded primarily in the Arduino coding language. The Arduino coding language is an abstracted form of C++, so all header files were also completed in C++ (see appendices for full program). One of the largest advantages to using the Arduino platform is the modularity of the devices it interfaces with. Almost every device that was used in this project came with some amount of prewritten code. With the use of preexisting libraries a master program was stitched together to accomplish all of the functions of the system. The RFID reader, load cell amplifier, and GSM/GPS board all came not only with header files that greatly simplified the implementation of these devices, but all also contained example sketches that highlighted many of the key features of these devices. From the basic sketch examples and the predetermined functions in the header files, a master program was stitched together to accomplish all of the tasks of the Household Waste Tracking System.

iv. System Implementation – Web Display

After the GSM board makes a post to dweet.io, the information embedded in that post is retrieved by the service Freeboard.io [20]. Using the ID information from the embedded RFID tag as a unique identifier, Freeboard is able to establish who the bin belongs to and the type of waste the bin is. The data from that bin is then displayed in a historical context and in a numerical value that tells the customer how much waste was retrieved on the last collection. Additionally, the location information of the last retrieval is displayed on a google map.

Figure 17 - Freeboard.io web display showing waste information for two customers as well as unidentified customers

v. System Implementation – Test Rig

In addition to the device and web display of the system, this project required the construction of a mechanical apparatus to simulate the waste collection processes using a variety of 80/20 aluminum parts. This testing rig cradles waste bins similarly to that of a waste collection vehicle. When initially placed in the apparatus, waste bins rest all of their weight on the load cell which has been placed directly beneath the cradle. After the household waste tracking system records the identity and initial weight of the waste bin, the operator uses the mechanical test rig to dump the waste bin and return the waste bin to the load cell, where the empty waste bin weight can be used to calculate the net weight of the waste that was in the bin.

Figure 18 - Test rig with waste bin

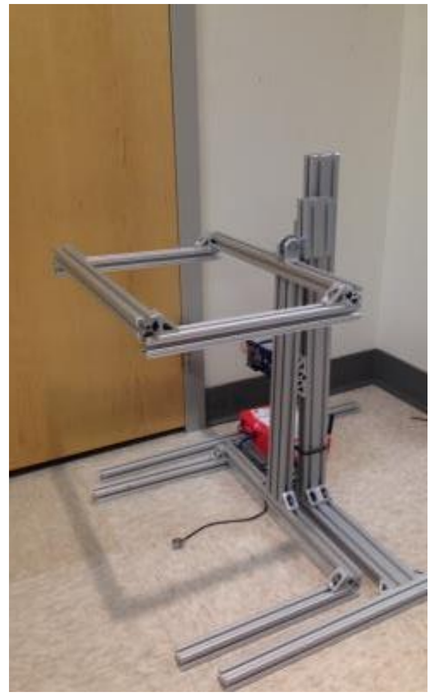


Figure 19 - Empty Test rig with waste tracker

VIII. PERFORMANCE ESTIMATES AND RESULTS

i. LOAD CELL

The load cell is a crucial element of the system. The entire accuracy of the system is contingent on the ability of device to quantify the amount of waste in any given bin. Since the load cell is the device that transforms physical weight quantities into an electrical signal it is paramount that the device is able to do so repeatedly and accurately. To test the accuracy of the load cell, a predetermined weight, in this case a 0.5 kg weight was placed on the cell to determine a calibration factor. Then a series of weights were placed on the scale and their measured weights were noted and compared to their known weights.

Table 2 - Load cell testing

Known Weight	Measured Weight
100g	99g
200g	199g
500g	498g

As shown in the table above, the system is able to read the accuracy of a weight within 2g of the known weight which is greater than an order of magnitude greater accuracy than required for the system design requirement of acquiring accuracy within the 84g tolerance.

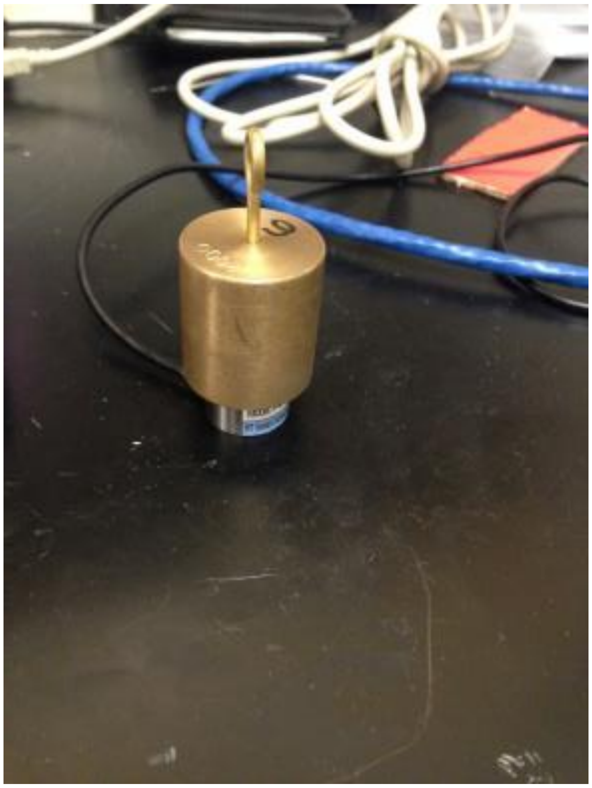


Figure 20 - Load cell testing with 200g weight

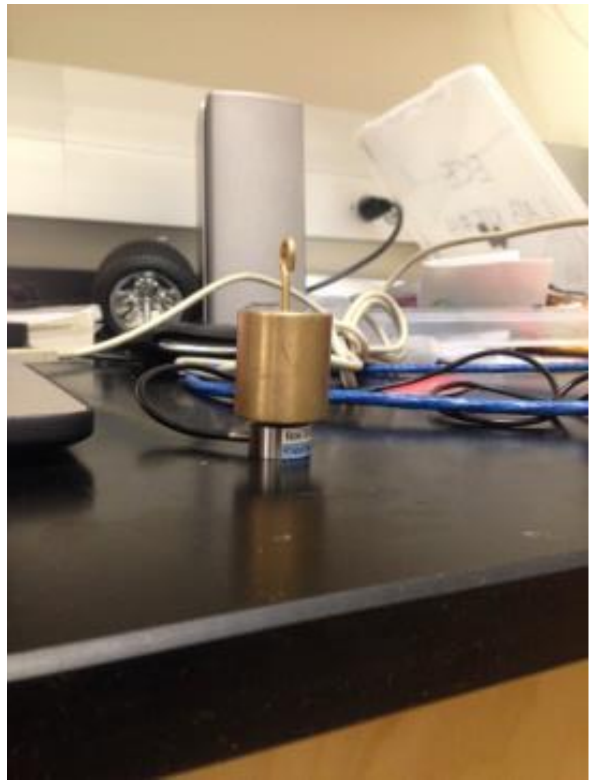


Figure 21 - Profile view of load cell testing

Additional testing was planned to determine the temperature dependence of the load cell, but due to the time restrictions of the project, this testing was ultimately left unfinished. According to the data sheet the load cell should have certain temperature dependence so commercial applications of the system should provide temperature compensation to the programming of the system.

ii. RFID

Testing showed that the reader could identify tags at a maximum distance around 9 cm but the distance of the read was heavily reliant on the orientation of the card relative to the reader (i.e. parallel, perpendicular, angled), as well as displacement from the reader.

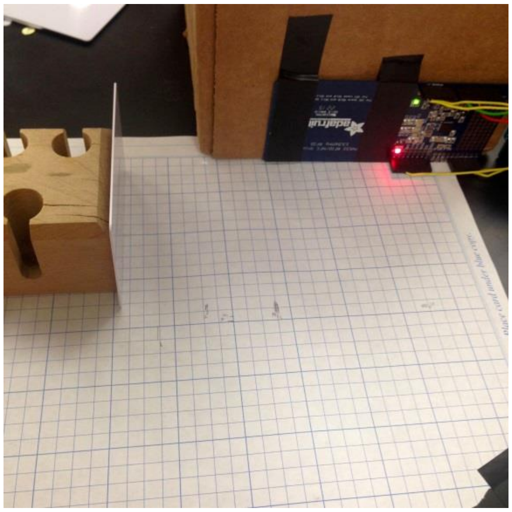


Figure 22 - RFID testing using a perpendicular tag orientation

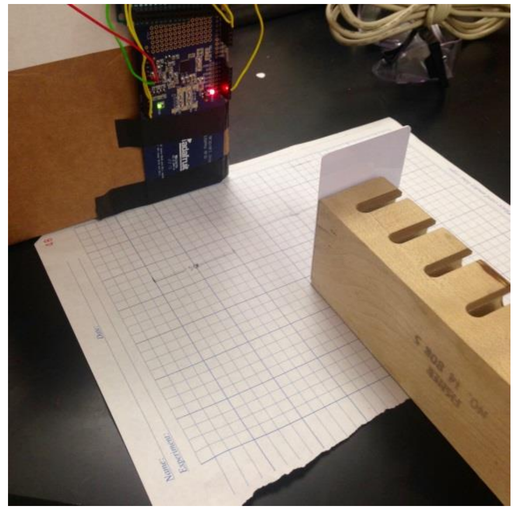


Figure 23 - RFID testing using a parallel tag orientation

The experimental set up consisted of a marked field with the RFID reader at a set location affixed to an upright prop, and a tag that was attached to a mobile upright prop. The tag was then moved toward the reader along a testing path according to the test, until the reader signaled that it read the card. The distance from the reader was then marked and each orientation of the experiment was repeated five times.

In front	
Trial	Distance (cm)
1	9.1
2	9.2
3	9.1
4	9.2
5	9.5
Average	9.22

In front (5cm displacement left)	
Trial	Distance (cm)
1	7.25
2	7.4
3	7.5
4	7.3
5	7.4
Average	7.37

In front (5cm displacement right)	
Trial	Distance (cm)
1	4.8
2	4.6
3	4.6
4	4.7
5	4.7
Average	4.68

Above (5cm displacement right)	
Trial	Distance (cm)
1	8.1
2	8.1
3	8.1
4	7.9
5	7.9
Average	8.02

Onside (perpendicular)	
Trial	Distance (cm)
1	3.6
2	3.65
3	3.7
4	3.8
5	3.8
Average	3.71

These findings show that the RFID reader must be as close as possible to the waste bin. Likely the reader will need to placed so that it can even come into contact with the waste bin, so a protective element may be necessary.

iii. GPS

In addition to providing an auxiliary method by which to establish a customers' identity, location services are also the backbone of the large data analytics portion of the household waste tracking system. Due to the limited resources of this project, precise scientific analysis of the GPS components used to construct the household waste tracking system was infeasible. However, using google maps to verify coordinates, a system that has been known to have its own inaccuracies, the system was able produce locations that were within 15 meters of the reference location, as shown below in the figure.

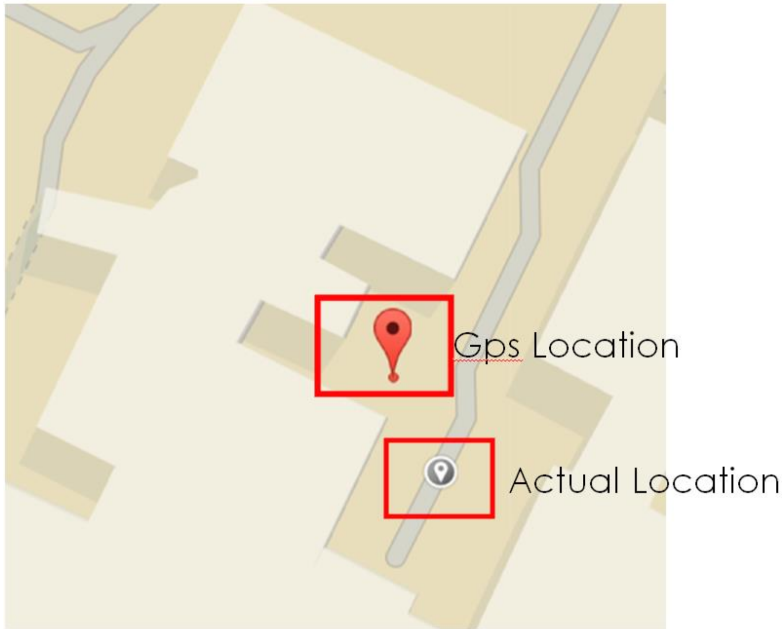


Figure 24 - Location test revealing a location 12.27 meters from actual location

The US government verifies that GPS locations are overwhelmingly found to be accurate to 3.5 meters, and that in a worst case scenario, a GPS location with provide accuracy within 7.8 meters with 95% accuracy. [5]

iv. GPRS

In the event of GPS failure, the system is to rely on the location provided through by GSM towers. This location is provided by calculating the distance to the closest accessible cell towers. Unfortunately, even when the system is able to communicate with multiple cell towers, giving the system the most reliable location, results are often inaccurate. The GPRS was tested in the same fashion as the GPS module, by comparing the system given location to the pinpointed location on google maps. Although this method is less that spectacularly scientific, it does give insight into the relative accuracy of the location data. Shown below is a figure of a common GPRS result.

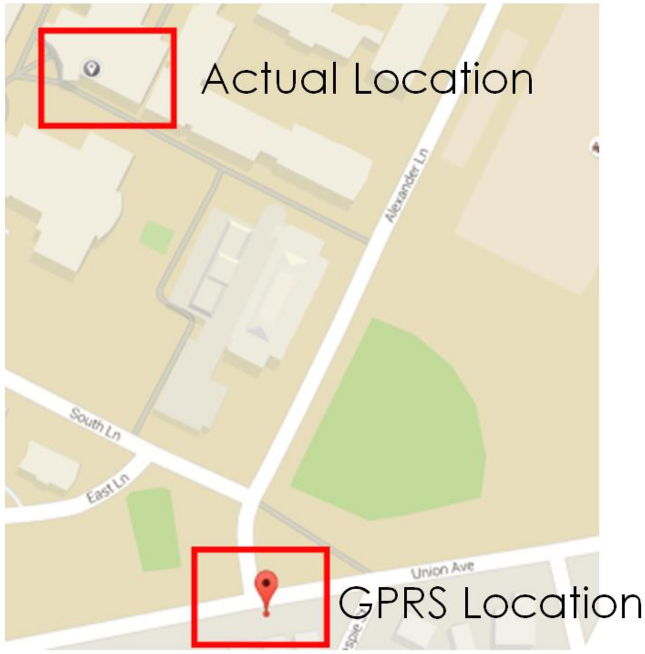


Figure 25 - GPRS location test results in comparison to actual location 204 meters away

Although the data is not accurate enough to use to verify the identity of a bin collection, this result could still be relevant in assessing waste trends.

v. SD Data Saving

SD storage is the key to keeping data secure even when GSM transmission fails. In the beginning of implementing SD data storage device there arose some concern over the viability of the method, as the break out board seemed to cut out accessibility every so often. With some experimentation it was found that the system simply needed some pull-up resistors and pull-down resistors to ensure that the board would function correctly all of the time.

vi. Cellular Data

Data transmission off the device during collection is completely dependent on the system's ability to make http request through the GSM network, so the feature must be very reliable in the household waste tracking system. Although occasional errors occur when attempting to use the GSM features on the board, these error have been overwhelmingly due to the AT style communication between the Arduino and the chip on the GSM board. The household waste tracking system has attempted to irradiate these errors by flushing the serial communication after every communication between the Arduino and GSM board, but occasionally the error occurs. Simply resetting the Arduino using the “reset” button fixes this problem. When GSM is available the system has no problem sending information through http requests, but in the event that the system does not have access to such a network the data will still be saved to the SD memory.

vii. Web Display

Although the web display of the household waste tracking system accomplishes its goal of receiving and displaying waste data from the device, it would be infeasible to use the current system in a commercial setting. Due to the structure of Dweet.io, data can only be saved on their servers for a day using the free version of the service, and only a month on the subscription based version of their platform. Furthermore, Freeboard.io is limited to only displaying data that it receives during the span that the window is open. These applications work wonderfully as a proof of concept for the project but would not allow a waste management company or government to analyze the data over a long period of time, and would be difficult to use to correlate waste data to service fees. Ultimately if the system were to be implemented in a large scale commercial fashion, dedicated servers and databases would need to be established to display and store the data generated by the system.

IX. PRODUCTION SCHEDULE

The production cycle for the Household Waste Tracking system was originally scheduled to span between September (fall term) 2015 and March (week 9 of winter Term) 2016. To this end, the project schedule was broken down into several major tasks: Research, Parts Acquisition, Construction, Coding, Testing, Debugging, Presentation Preparation, and Reporting.

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION
Research	1	5	1	4
Parts Aquistion	5	5	5	18
Construction	7	12	8	16
Coding	10	10	12	12
Testing	8	15	8	16
Debugging	10	12	12	12
Presentation Prep	23	2	24	1
Report Writing	20	6	24	3

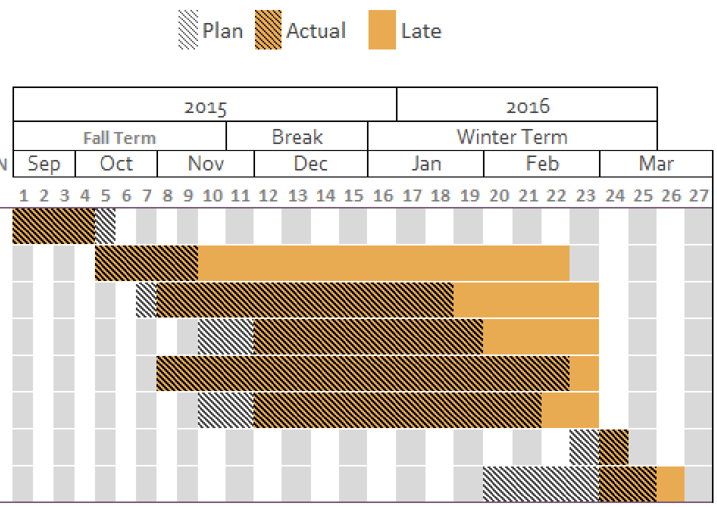


Figure 26 - Purposed and actual implementation schedule of the system

As originally purposed, the production schedule for the system became increasingly ambitious. Major setbacks including damaged and inoperable parts held the system back from completing many of the major tasks on the schedule. Ultimately, adhering to the proposed schedule would have served the project greatly but the system was still fully completed and functional by week 8 of winter term.

X. COST ANALYSIS

As accounted for in the table below, the system cost approximately $280 to develop plus an additional $160 allocated to develop the mechanical testing apparatus for the system. These expenses are broken down into the following two tables.

Table 3 - Test apparatus cost breakdown

Part	Individual Price	Quant.	Total
72 in. 1x1 Extrusion	$18.51	2	$37.02
24 in. 1x1 Extrusion	$7.47	1	$7.47
68.5 in 2x2in Extrusion	$28.67	1	$28.67
Pivot Joint	$5.25	1	$5.25
2 Hole Gusset	$1.75	14	$24.50
4 Hole Gusset	$5.65	2	$11.30
Linear Bearing Housing	$9.99	1	$9.99
Linear Bearing Pads	$1.60	3	$4.80
Aluminum Sheet	$2.03	2	$4.06
Screw & Nut set (25 pcs.)	$12.50	2	$25.00
		Total	$158.06

Although the cost of the mechanical test apparatus represents a significant amount of the project expenditures, in a full scale prototype these expenses would not exist as the system could be adapted to a real waste collection vehicle.

Table 4 - Total system cost breakdown

Item	Manufacter	Part #	Quantity Needed	Price Per Unit	Total Estimated Price
Micro Controller	Arduino	Mega	1	$45.95	$45.95
GPS/GSM sheild	Adafruit	Fona µ808	1	$49.95	$49.95
GPS Antena	Adafruit	Passive GPS Antenna	1	$3.95	$3.95
GSM Antena	SparkFun	Quad-band Cellular Duck Antenna SMA	1	$7.95	$7.95
GSM Battery Pack	SparkFun	Polymer Lithium Ion Battery	1	$12.95	$12.95
Battery Pack Charger	SparkFun	parkFun LiPo Charger Basic - Mini-USB	1	$7.95	$7.95
50 Kg Load Cell	HT SENSOR TECHNOLOGY CO	TAS606	1	$56.95	$56.95
Load Cell amp	Keyes	HX711	1	$4.81	$4.81
Micro Controller Power Supply	Adafruit	9V battery holder with switch	1	$3.95	$3.95
9V Battery	Duracell	9V Battery	1	$1.25	$1.25
SD read/Write boardx	Arduino	120801	1	$1.97	$1.97
RFID Reader	Adafruit	PN532	1	$39.95	$39.95
RFID Chips	MiFare	Classic Card	4	$2.50	$10.00
Arduino Headers	SparkFun	Female Header Pack	2	$1.50	$3.00
SD Card	Excelvan	8GB TF Card with Adapter Class 6	1	$0.49	$0.49
Cell Contract/SIM card	T-mobile	1 Month 3 gB plan	1	$30.00	$30.00
Resistors	General		11	$0.15	$1.65
LEDs	General		7	$0.15	$1.05
Test Rig	80/20 Parts	Aluminum Parts	1	158.06	$158.06
				Total Estimate:	$441.83
				device:	$283.77

XI. OPERATION MANUAL

The household waste tracking system is designed to be as easy to use as possible, as it is meant to be retrofitted to preexisting waste retrieval vehicles. On a vehicle that is already meant to automate waste retrieval, the waste tracking device would be integrated into the dumping process, so no user operation would even be necessary. However, in the current implementation, the device requires users to dump the waste in a fairly particular manner. This manual will focus on operating the device, using the test rig, which was constructed to simulate the dumping process of a waste collection routine. Additionally, this manual will inform an operator how to maintain the web display of the system.

i. Waste Retrieval

The routine for operating the household waste removal is repetitive and simple. The process status is indicated to the operator through the seven LEDs that are mounted on top of the device enclosure.

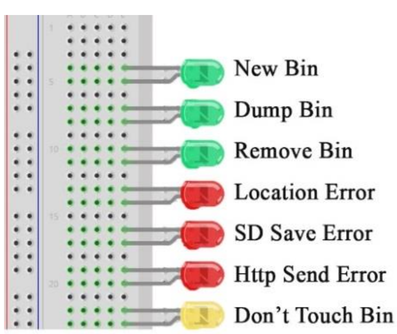


Figure 27 - LED light designation

Figure 28 - Implementation of user notification LEDs

After turning on the device, by flipping the switch on the 9V battery supply, the operator must wait approximately 30 seconds for the system to initialize. When the system has been initialized, the first green LED will illuminate, indicating to the user that the system is ready to start the waste collection of a new bin. The operator then places a waste bin in the cradle section of the dumping rig with the side of the bin containing the RFID tag aligned with the RFID reader. In a commercial setting the side of a waste bin would have visual markers in addition to the compulsory waste bin direction that most waste management services already require, indicating the side that contains the RFID tag such that aligning the tag and the reader would be trivial.

After a full bin is placed in the receptacle, the system automatically reads the RFID tag and records the identity of the waste bin. At the same time, the yellow LED illuminates indicating to the user that they should not touch the waste bin or the dumping rig at this time. While the yellow LED is on, the system is waiting for the load cell to stabilizing and recording the “Full” weight of the waste bin. When the system has established and recorded a “full” weight both the first green LED and the yellow LED will turn off.

Figure 29 - Full bin on rig, establishing full weight

At this time the second LED illuminates informing the user to dump the waste from the bin. The user then lifts the cradle of the rig and dumps it over the back side of the center column. The device waits until it notices a weight placed on the load cell to proceed. Once the operator is finished dumping the waste, they return the empty waste bin back to the original orientation, still in the cradle. The yellow light then re-illuminates, indicating once again that the user should not touch the bin or dumping rig.

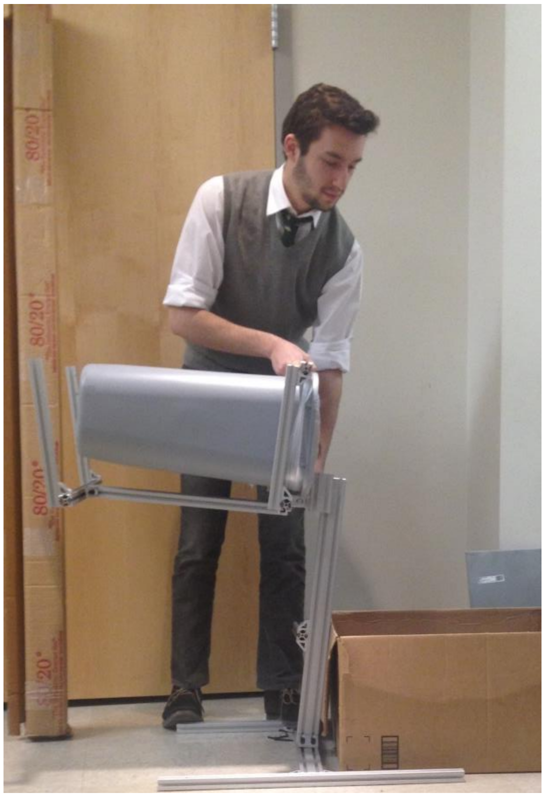


Figure 30 - Testing apparatus amid simulated dumping procedure

Once the yellow LED turns off again, the final green LED illuminates indicating that the user should remove the waste bin to restart the process.

When the final green LED illuminates, the system simultaneously attempts to acquire location data, save the weight, ID, and location data to SD memory, and transmit all of the data to the web server. If any of these processes fail, a corresponding red LED will illuminate. If the first red LED illuminates, the system was unable to establish a GPS location. The location may have been established through GPRS tracking, but the system will not indicate whether or not this has occurred. The second LED illuminates in the event that the system was unable to save the data to the SD memory, but this process is highly stable and memory failure is unlikely. The final red LED illuminates in the event that the system is unable to transmit the data to the web server. This last form of failure is usually due to a communication error inherent to the cellular module. Should this third LED illuminate, the operator should open the enclosure of the device and press the reset button on the Arduino, which will flush the communication of the cellular module and prepare the system to continue collecting data. Resetting the Arduino will not reset the SD memory, so the operator should not be worried about losing preexisting data.

ii. Web Display

When a new customer is added to the system, the web server must be updated to display the data of that customer. The RFID number that was programmed into the tag of the new customers bin will serve as the customer's ID linking them to that bin. On the freeboard.io website, the operator must add a new data source with that ID number, labeling the data source according to the customer's name and the type of waste of that bin (landfill, recycling, etc.). Then the operator must create a new pane with the widgets corresponding to their display preferences. In the current implementation, gauge widgets were used to indicate the weight of the latest retrieval, spark lines to display historical collections, and a Google map to indicate the location of the latest retrieval. When allocating data sources on widgets the weight of a bin will be identified with the value "Weight", and the location will be identified using "Lat" and "Long" for latitude and longitude coordinates. The operator will not need to update any part of Dweet.io as the system will automatically format information for use on Dweet, and Dweet will automatically create new “things” as new bins are collected for the first time.

XII. DISCUSSION, CONCLUSIONS, AND RECOMMENDATIONS

Although there is no easy answer to the complex problem that is consumer waste, the household waste tracking system offers one solution that uses the capitalist system that created the waste issue to incentive our society to rehabilitate our waste practices. By generating large and small data, the system will be able to produce the information needed to create a system of proportional costs for waste. As a result, local governments and waste management companies will be able to provide financial subsidies in the form of tax breaks and bill reductions for responsible waste habits. Additionally, the large scale data accumulated through location tracking will allow a reduction in inefficient waste collection routes, saving companies’ money and unnecessary exhaust emissions from the environment.

The household waste tracking system correlates specific quantities of waste to specific customers by weighing the waste in specific bins upon collection. The bins having been embedded with RFID tags will inform the system as to which customer owns the bin. GPS/GPRS services will allow the system to identify the location of each bin retrieval. Finally, all of this information is sent wirelessly through GSM to a web based display that will allow waste management companies, customers, and local governments to utilize the data collected by the system.

This system is able to read the weight of a bin down to the precision of 2 grams, giving waste management companies more than enough information to correlate a bill to the customer waste habits. Furthermore the RFID capabilities allow the system to accurately identify a bin within a proximity of 9 cm, which, based on the positioning of the reader and tags, which could easily be achieved in a repetitive collection routine. Utilizing GPS technology the system tracks bin retrieval with better than 3.5 meter horizontal accuracy [5] (http://www.gps.gov/systems/gps/performance/accuracy/)., a statistic confirmed by the testing of the system. In the event of GPS failure, the system is able to track retrieval through a slightly less accurate method, GPRS. System testing proved that GPRS accuracy was close 200 meters, which is enough accuracy for most collection route analysis. After location, weight, and identification information is collected, it is stored on an SD card and sent to an online web service for display.

With the current implementation of the system, the web service can display all of the packages that have been retrieved in the past day. For a commercial implementation of the system, a custom database and server would need to be established for long-term data collection and storage.

In the future, the testing rig used to evaluate the accuracy and procedural stability of the system would need to be replaced with an automated waste vehicle arm. This process would need the supervision and advice of both electrical and mechanical engineers, but would ultimately allow the system to be adopted as a standard of the waste management system.

XIII. REFERENCES

[1]"Advancing Sustainable Materials Management: Facts and Figures | Sustainable Materials Management | US EPA", *Epa.gov*, 2016. [Online]. Available: http://www.epa.gov/waste/nonhaz/municipal/. [Accessed: 18- Mar- 2016].

[2] Environmental Protection Agency, "RECYCLING", 2016.

[3]M. Arebey, M. Hannan, H. Basri, R. Begum and H. Abdullah, 'Integrated technologies for solid waste bin monitoring system', *Environmental Monitoring and Assessment*, vol. 177, no. 1-4, pp. 399-408, 2010.

[4] Pontech Gesellschaft Fur Technologische Entwicklungen Mbh, 'Process and device for recording the weight of refuse material as the material is emptied into a refuse 16 collecting vehicle', US 4645018 A, 1987.

[5]H. Hornhauer, "Apparatus for weight-based charging for waste collection containers", DE19536318 A1, 2016.

[6] Internetofeverything.cisco.com, 'Cisco Internet of Everything', 2015. [Online]. Available: http://internetofeverything.cisco.com/vas-public-sector-infographic/. [Accessed: 25- Nov- 2015].

[7]Akbari, A.; Mirshahi, S.; Hashemipour, M., "Comparison of RFID system and barcode reader for manufacturing processes," in *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on* , vol., no., pp.502-506, 3-6 May 2015

[8]"Arduino - ArduinoBoardMega2560", Arduino.cc, 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardMega2560. [Accessed: 18- Mar- 2016].

[9]"Arduino - ArduinoBoardUno", Arduino.cc, 2016. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno. [Accessed: 18- Mar- 2016].

[10]"Raspberry Pi 2 Model B", Raspberry Pi, 2016. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/. [Accessed: 18- Mar- 2016]

[11]"Adafruit PN532 NFC/RFID Controller Shield for Arduino: Adafruit Industries, Unique & fun DIY electronics and kits", Adafruit.com, 2016. [Online]. Available: https://www.adafruit.com/products/789. [Accessed: 18- Mar- 2016]..

[12]R. Want, 'An Introduction to RFID Technology', IEEE Pervasive Comput., vol. 5, no. 1, pp. 25-33, 2006.

[13]R. Want, "Enabling ubiquitous sensing with RFID," in Computer, vol. 37, no. 4, pp. 84-86, April 2004.doi: 10.1109/MC.2004.1297315

[14]"Load Cell - 50kg, Disc (TAS606) - SEN-13331 - SparkFun Electronics", Sparkfun.com, 2016. [Online]. Available: https://www.sparkfun.com/products/13331. [Accessed: 18- Mar- 2016].

[15]"SparkFun Load Cell Amplifier - HX711 - SEN-13230 - SparkFun Electronics", Sparkfun.com, 2016. [Online]. Available: https://www.sparkfun.com/products/13230. [Accessed: 18- Mar- 2016].

[16] A. Board, "Arduino 120801 Micro SD Card Reading and Writing Board", http://www.gearbest.com, 2016. [Online]. Available: http://www.gearbest.com/development-boards/pp_63013.html. [Accessed: 18- Mar- 2016].

[17]"Adafruit FONA 808 - Mini Cellular GSM + GPS Breakout ID: 2542 - $49.95 : Adafruit Industries, Unique & fun DIY electronics and kits", Adafruit.com, 2016. [Online]. Available: https://www.adafruit.com/products/2542. [Accessed: 18- Mar- 2016].

[18] "Polymer Lithium Ion Battery - 2000mAh - PRT-08483 - SparkFun Electronics", Sparkfun.com, 2016. [Online]. Available: https://www.sparkfun.com/products/8483. [Accessed: 18- Mar- 2016].

[19] Bug Labs, Inc, "dweet.io - Share your thing- like it ain't no thang.", Dweet.io, 2016. [Online]. Available: http://dweet.io/. [Accessed: 18- Mar- 2016].

[20] [14] Bug Labs, Inc, "freeboard - Dashboards For the Internet Of Things", Freeboard.io, 2016. [Online]. Available: http://freeboard.io/. [Accessed: 18- Mar- 2016].

XIV. ACKNOWLEDGEMENTS

I would like to thank many people for their help on this project. First off I would like to acknowledge Walter Dixon and Cherice Traver for all of their wonderful guidance throughout the project. I would like to thank Brad Bruno for his advice in designing the mechanical facets of the project . I would also like to thank Andrew Laugharn and John Sheehan for their assistance cutting aluminum and making sure I didn't lose any fingers in the process. I would also like to thank Sparkfun and Adafruit for their wonderfun libraries that greatly simplified my life.

XV. APPENDICES

Arduino Program Code:

```
/*
 The circuit:
  * SD card attached to SPI bus as follows:

SPI   Mega
SS     53
MOSI   51
MISO   50
SCK    52
*/

//realocate these vars!!!!!
String clongitude;
String clatitude;
int lata;
int latb;
int latc;
int lona;
int lonb;
int lonc;




//SDheaders
#include <SPI.h>
#include <SD.h>

//Load Cell Headers
#include "HX711.h"
#define DOUT  30
#define CLK  31
HX711 scale(DOUT, CLK);
```

```
float calibration_factor = -1000; //-750 worked for my 440lb max scale setup
int weighta;
int weightb;

//RFID headers
#include <Wire.h>//I2Cheader
#include <Adafruit_NFCShield_I2C.h>
#define IRQ   (4)
#define RESET (5)  // Not connected by default on the NFC Shield
Adafruit_NFCShield_I2C nfc(IRQ, RESET);
unsigned long ID = 0;

//Can variables  init
typedef struct _can {
  char weight [32];
  unsigned long id;
  char longitude [32];
  char latitude [32];
} can;
can mycan;
void SaveSD(can& ccan);
int savecount = 0;
#define WEIGHT_THRESH (1.5)


//CEll&GPS init
#include "Adafruit_FONA.h"
#define FONA_RST 2
HardwareSerial *fonaSerial = &Serial3;
Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
float canlatitude, canlongitude, canaltitude;
char replybuffer[255];
bool dweet(can& ccan);
uint8_t readline(char *buff, uint8_t maxbuff, uint16_t timeout = 0);
uint8_t type;
```

```
//LED inits
//int LED = 13;
//int LEDfail = 12;
int Newbin = 28;
int Dumpbin = 27;
int Removebin =26;
int LocFail = 25;
int sdFail = 24;
int GSMFail = 23;
int wait =  22;
int cnt = 0;
boolean gprsenabled = false;
boolean fonaavailable = false;
void setup() {

  // put your setup code here, to run once:
  // Open serial communications and wait for port to open:
//  pinMode(LED, OUTPUT);
//  pinMode(LEDfail, OUTPUT);
  Serial.begin(115200);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
  //LED setup---------------------------------------------------------------------------------------------------
  pinMode( Newbin, OUTPUT);
  pinMode( Dumpbin, OUTPUT);
  pinMode( Removebin, OUTPUT);
  pinMode( LocFail, OUTPUT);
  pinMode( sdFail, OUTPUT);
  pinMode( GSMFail, OUTPUT);
  pinMode( wait, OUTPUT);
```

```
  //GPS&GSM setup -----------------------------------------------------------------------
-------------------------------------------------------------------------------------------
  fonaSerial->begin(4800);
  if (! fona.begin(*fonaSerial)) {
    Serial.println(F("Couldn't find FONA"));
  }
  else{
      fonaavailable = true;
        Serial.println(F("FONA is OK"));
      // Print module IMEI number.
    char imei[15] = {0}; // MUST use a 16 character buffer for IMEI!
    uint8_t imeiLen = fona.getIMEI(imei);
    if (imeiLen > 0) {
      Serial.print("Module IMEI: "); Serial.println(imei);
    }
//    gprsbegin();
      gprsend();

  }

  //RFID setup----------------------------------------------------------------------------
-------------------------------------------------------------
  nfc.begin();
  uint32_t versiondata = nfc.getFirmwareVersion();
  if (!versiondata) {
    Serial.print("Didn't find PN53x board");
  }
  else{
    Serial.print("Found chip PN5"); Serial.println((versiondata>>24) & 0xFF, HEX);
    Serial.print("Firmware ver. "); Serial.print((versiondata>>16) & 0xFF, DEC);
    Serial.print('.'); Serial.println((versiondata>>8) & 0xFF, DEC);
  }

  // configure board to read RFID tags---------------------------------------------------
------------------------------------------------------------------------
  nfc.SAMConfig();
```

```
  Serial.println("Waiting for an ISO14443A Card ...");

 //HX711 calibration---------------------------------------------------------------------------
-------------------------------------------------------------------
  scale.set_scale(calibration_factor);
  scale.tare();  //Reset the scale to 0
  //serialTrigger();
}

void loop() {//MAIN  LOOP ---------------/-/-/-/-/-/-/-/-------------------------/-/-/-/-/-/-
/-------------------------------------------------------/-/-/-/-/-/-//------------
  digitalWrite(Newbin, HIGH);
  memset(&mycan,0,sizeof(mycan));
  ID = 0;
  // open the file.
  mycan.id=RFIDsearch();
  Serial.println("my can ID:");
  Serial.println(mycan.id);
  float ccani=loadread();//initial current can weight
  float ccand=0;//dumped current can weight
  Serial.print("Current Weight:");
  Serial.println(ccani);
  Serial.println("lbs");

  if (mycan.id != 0 || ccani >= WEIGHT_THRESH){
    if (mycan.id != 0){
      while(ccani <= WEIGHT_THRESH){
          Serial.println("CURRENT ID#");
          Serial.println(mycan.id);
        digitalWrite(wait, HIGH);
        ccani=loadread();
      }
    }
    digitalWrite(Newbin, LOW);
    digitalWrite(GSMFail, LOW);
```

```
    digitalWrite(sdFail, LOW);
    digitalWrite(LocFail, LOW);
    cleanloc();
    digitalWrite(wait, HIGH);
    delay(1000);//weight stablization

    ccani=loadread();// Full Can Weight
    float tempweight = loadread();
    digitalWrite(wait, LOW);
    while(tempweight >= WEIGHT_THRESH){
     // int time= millis();
     digitalWrite(Dumpbin, HIGH);
      Serial.println("Dump Can");
      tempweight=loadread();
      delay(500);
      //indicator= ~indicator
//        if(time - millis()>= 30000){
//          Serial.println("Can dump error");
//          break;
//        }
    }
    while(tempweight <= WEIGHT_THRESH){// Empty can weight
      //int timer= millis();
      Serial.println("Relpace Can");
      tempweight=loadread();
      delay(500);
      //indicator= ~indicator
//        if(timer- millis()>= 30000){
//          Serial.println("Can dump error");
//          break;
//        }
    }
    delay(1000);

    float ccanw=ccani - loadread();
```

```
digitalWrite(Dumpbin, LOW);
digitalWrite(Removebin, HIGH);
Serial.println("Remove Can");
Serial.println("Net Can Weight:");
Serial.println(ccanw);
Serial.println("");
//light indication
String weights= String(ccanw);
weighta=(int)ccanw;
weightb=(int)(100*ccanw-100*weighta);
sprintf(mycan.weight,"%d.%d", weighta,weightb);
Serial.println(mycan.weight);

if(mycan.id == 0){
  mycan.id=RFIDsearch();
}

ID = mycan.id;
Serial.println("CURRENT ID#");
Serial.println(mycan.id);
if(locsearch()){
  Serial.println("Location found");
  sprintf(mycan.longitude,"%d.%d%d",lona,lonb,lonc);
  sprintf(mycan.latitude,"%d.%d%d",lata,latb,latc);
  //mycan.latitude=clatitude.toFloat();
  delay(1000);
}
else{
   Serial.println("Location Not found!!!!");
   digitalWrite(LocFail, HIGH);
}
SaveSD(mycan);
if(dweet(mycan) == true){
  Serial.println("GSM Sent!!");
}
```

```
    else{
      Serial.println("GSM NOT Sent!!");
      digitalWrite(GSMFail, HIGH);
    }
  }
  float tempweight = loadread();
  while(tempweight >= WEIGHT_THRESH){
     // int time= millis();
      Serial.println("Remove Can");
      tempweight=loadread();
      delay(500);
      //indicator= ~indicator
//        if(time - millis()>= 30000){
//          Serial.println("Can dump error");
//          break;
//        }
    }
  digitalWrite(Removebin, LOW);
}


float loadread () { // Read Load cell-----------/-/-/-/-/-/-/-/-----------------------/-/-/-/-/-/------------------------/-/-/-/-/-/------------------------/-/-/-/-/-/---------------------------
  float val = scale.get_units();
  Serial.print("read Weight:");
  Serial.println(val);
  return -val;


}



void SaveSD(can& ccan) { // save can info to SD card---------------------------------------------------------------------------------------------------------------------
//parseGPS to make file name (one per day/route)
//Save data object to .txt file
```

```
  //set up sd
  Serial.print("Initializing SD card...");
  if (SD.begin(53)){//53 is the enable pin of the miso/mosi SD pin group
    File dataFile = SD.open("datalog.csv", FILE_WRITE);
    if (dataFile){
      dataFile.print(ccan.id);
      dataFile.print(",");
      dataFile.print(ccan.weight);
      dataFile.print(",");
      dataFile.print(ccan.latitude);
      dataFile.print(",");
      dataFile.print(ccan.longitude);
      dataFile.print("\n");
      dataFile.close();
      // print to the serial port too:
      Serial.println("Data Saved to SD Card");
      //digitalWrite(LED, HIGH);
   }
   else {
      Serial.println("Can't open datafile");
      digitalWrite(sdFail, HIGH);
   }
   SD.end();
 }
 else{
    Serial.println("initialization failed!");
    digitalWrite(sdFail, HIGH);
    return;
 }
}

bool dweet(can& ccan){// GSM send ------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------
  bool sent= false;
  gprsbegin();
```

```
  flushSerial();
    if(gprsenabled){
      Serial.println("Send to:");
    Serial.println(ccan.id);


//    serialTrigger();


    String thing = String(ccan.id);
     if(ccan.id == 0){
    thing = "000001";
    }
    char url [64];
    sprintf(url,"http://dweet.io/dweet/for/%s",thing.c_str());
    Serial.println(url);


    char postdata [64];
sprintf(postdata,"Weight=%s&Lat=%d.%d%d&Long=%d.%d%d",ccan.weight,lata,latb,latc,lona,lonb,lon
c);
    Serial.println(postdata);
    uint16_t postdatalen;
    postdatalen=strlen(postdata);
    uint16_t statuscode;
    uint16_t datalen;

  //  char *url,
  //              FONAFlashStringPtr contenttype,
  //              const uint8_t *postdata, uint16_t postdatalen,
  //              uint16_t *status, uint16_t *datalen
    if(cnt<=4){
    fona.HTTP_POST_start(url,
                  F("application/x-www-form-urlencoded"),
                  (uint8_t*)postdata, postdatalen,
```

```
                  &statuscode, &datalen);
     while (postdatalen > 0) {
          while (fona.available()) {
            char c = fona.read();

            #if defined(__AVR_ATmega328P__) || defined(__AVR_ATmega168__)
                        loop_until_bit_is_set(UCSR0A, UDRE0); /* Wait until data register
empty. */
                        UDR0 = c;
            #else
                        Serial.write(c);
            #endif

            postdatalen--;
            if (! postdatalen) break;
          }
        }
    Serial.println(statuscode);
    if(statuscode>=200 && statuscode<=300){
        sent = true;
    }
    Serial.println("sent attempted");
    fona.HTTP_POST_end();
    gprsend();
    }
  }
  flushSerial();
  return sent;
}

void serialTrigger() { //Serial window trigger used for debugging-----------------------------
---------------------------------------------------------------------------
  Serial.println("Send some serial to start");

  while (!Serial.available()){
//    //digitalWrite(LED, HIGH);   // turn the LED on (HIGH is the voltage level)
```

```
//    delay(1000);              // wait for a second
//    //digitalWrite(LED, LOW);    // turn the LED off by making the voltage LOW
//    delay(1000);
  }
  while (Serial.available()){
    Serial.read();
  }
//  digitalWrite(LEDfail, LOW);
}



unsigned long RFIDsearch(){ //Search for RFID TAG info--------------------------------------
----------------------------------------------------------------------------
  typedef union _id {
    uint8_t uid[4];  // Buffer to store the returned UID
    unsigned long id;
    }
  uuid;

  uuid uid;

  uint8_t uidLength; // Length of the UID (4 or 7 bytes depending on ISO14443A card type)


  // Wait for an ISO14443A type cards (Mifare, etc.).  When one is found
  // 'uid' will be populated with the UID, and uidLength will indicate
  // if the uid is 4 bytes (Mifare Classic) or 7 bytes (Mifare Ultralight)

  uint8_t success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid.uid, &uidLength,1000);
  if (success) {
    // Display some basic information about the card
    Serial.println("Found an ISO14443A card");
    Serial.print("  UID Length: ");Serial.print(uidLength, DEC);Serial.println(" bytes");
    Serial.print("  UID Value: ");
```

```
    nfc.PrintHex(uid.uid, uidLength);
    Serial.println("");
  }
  else {
    uid.id = 0;
  }
  return(uid.id);
}

void gprsbegin(){// Begin GPRS Module-----------------------------------------------------
---------------------------------------- ----------------------------------
if(gprsenabled == false){
    if(!fona.enableGPRS(true)){
      Serial.println("GPRS not enabled");
      delay(1000);
    }
    else{
      Serial.println("GPRS enabled");
      gprsenabled = true;
    }
  }
}


void gprsend(){//End GPRS Module----------------------------------------------------------
-----------------------------------------------------------------------------


    if (!fona.enableGPRS(false)){
     Serial.println(F("Failed to turn off"));
     delay(1000);
    }
    else{
      Serial.println("GPRS Disabled");
      gprsenabled = false;
    }
```

```
}



char gettime(){// Get Time from GSM Network-----------------------------------------------------
--------------------------------------------------------------------------------------------
  gprsbegin();
  flushSerial();
  if (!fona.enableNTPTimeSync(true, F("pool.ntp.org"))){
    Serial.println(F("Failed to enable Clock"));
  }
  else{
    char buffer[23];
    fona.getTime(buffer, 23);
    Serial.print(F("Time = ")); Serial.println(buffer);
  }
  gprsend();
  flushSerial();
}



void flushSerial(){//Flushes both the fona and the usb serial--------------------------------
--------------------------------------------------------------------------------------------
  while (Serial.available())
    Serial.read();
  while (fona.available()) {
    Serial.write(fona.read());
  }
}


bool locsearch(){// Hierarchy of Location search----------------------------------------------
--------------------------------------------------------------------------------------------
  bool locfound = false;
  if(gpsloc() == false){//if Gps not available search for gprs
    Serial.println("Unable to find GPS Location");
    if(gprsloc() == false){//if Gprs not available break downcrying and go home also return
loc not found
      Serial.println("Unable to find GPRS Location");
```

```
    }
    else{
      Serial.println("Found useable GPRS Location");
      locfound = true;
    }
  }
  else{
    locfound = true;
    Serial.println("Found useable GPS Location");
  }
  flushSerial();
  return locfound;
}


bool gpsloc(){// finding the GPS location-------------------------------------------------------
---------------------------------------------------
  bool gpsuseable = false;
  if (!fona.enableGPS(true))//Turn ON Gps
      Serial.println(F("Failed to turn on"));
  else{
    int8_t stat;
      // check GPS fix
      stat = fona.GPSstatus();
      if (stat <= 1){
        Serial.println(F("GPS unsuable"));//If GPS status is less than 1 it is can not find a
fix
      }
      else{
        char gpsdata[120];//Gps buff
        fona.getGPS(0, gpsdata, 120);//retreive gps info
        Serial.println(F("Reply in format:
mode,fixstatus,utctime(yyyymmddHHMMSS),latitude,longitude,altitude,speed,course,fixmode,reserv
ed1,HDOP,PDOP,VDOP,reserved2,view_satellites,used_satellites,reserved3,C/N0max,HPA,VPA"));
        Serial.println(gpsdata);
        bool gpsuseable = true;
```

```
        String replystring = String(gpsdata);

        int latbegin = 0;
        for(int i=0;i<=3;i++){
          latbegin=replystring.indexOf(',', latbegin + 1);
        }
        latbegin=latbegin+1;
        int latend=replystring.indexOf(',', latbegin);
        int longend=replystring.indexOf(',', latbegin + 1);

        clatitude=replystring.substring(latbegin,latend);
        clongitude=replystring.substring(latbegin + 1,longend);

        Serial.println("Parsed:");
        Serial.println("Lat:");
        Serial.println(clatitude);
        Serial.println("Lat:");
        Serial.println(clongitude);

        locparse();
        flushSerial();

      }
      if (!fona.enableGPS(false))//Turn off Gps
          Serial.println(F("Failed to turn off"));

  }
  return gpsuseable;
}


bool gprsloc(){//aquiring location via gprs---------------------------------------------------
----------------------------------------
    bool gprslocuseable = false;
    gprsbegin();
```

```
if(gprsenabled == false){
  return gprslocuseable;
}
Serial.println("gprsloc called");
uint16_t returncode;
if (!fona.getGSMLoc(&returncode, replybuffer, 250))
  Serial.println(F("GPRS locFailed!"));
if (returncode == 0) {
      Serial.println("Raw GPRS Data:");
      Serial.println(replybuffer);

      String replystring = String(replybuffer);

      Serial.println("Longitute:");
      int longbegin = replystring.indexOf(',' + 1);
      int longend =replystring.indexOf(',',longbegin +1);
      clongitude = replystring.substring(longbegin,longend);
      Serial.println(clongitude);

      Serial.println("Latitude:");
      int latbegin = longend + 1;
      int latend =replystring.indexOf(',',longend +1);
      clatitude=replystring.substring(latbegin,latend);
      Serial.println(clatitude);

      locparse();

      bool gprslocuseable = true;

      gprsend();
      flushSerial();
}
else {
      Serial.print(F("Fail code #")); Serial.println(returncode);
    }
```

```
    return gprslocuseable;

}


void locparse() {//Chop up Location Data--------------------------------------------------------
------------------------------------------------------

      lata = clatitude.substring(0,clatitude.indexOf('.')).toInt();

      //Serial.println(lata);

      latb = clatitude.substring(clatitude.indexOf('.') + 1,clatitude.indexOf('.') +
4).toInt();

      //Serial.println(latb);

      latc = clatitude.substring(clatitude.indexOf('.') + 4).toInt();

      //Serial.println(latc);

      lona = clongitude.substring(0,clongitude.indexOf('.')).toInt();

      //Serial.println(lona);

      lonb = clongitude.substring(clongitude.indexOf('.') + 1,clongitude.indexOf('.') +
4).toInt();

      //Serial.println(lonb);

      lonc = clongitude.substring(clongitude.indexOf('.') + 4).toInt();

}

void cleanloc() {

  clongitude=String(0);

  clatitude=String(0);

  lata=0;

  latb=0;

  latc=0;

  lona=0;

  lonb=0;

  lonc=0;

}
```