DEVELOPMENT OF A SMART WASTE MANAGEMENT SYSTEM FOR CITY APPLICATION

A Thesis

Presented to

the Faculty of the

Elmer R. Smith College of Business and Technology

Morehead State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Lawrence D. Fraction

April 14, 2023

Accepted by the faculty of the Elmer R. Smith College of Business and Technology, Morehead State University, in partial fulfillment of the requirements for the Master of Science degree.

____

Dr. Anindita Paul
Director of Thesis

Master's Committee: __________________________________, Chair

Dr. Anindita Paul

____

Dr. Cheng Cheng

____

Dr. Ahmad Zargari

____

Date

DEVELOPMENT OF A SMART WASTE MANAGEMENT SYSTEM FOR CITY APPLICATION

Lawrence D. Fraction
Morehead State University, 2023

Director of Thesis: ____
Dr. Anindita Paul

The Internet of Things (IoT) is an emerging technology that has started to revolutionize our daily lives by utilizing the interconnectivity of small devices to allow the transmission of data across interconnected networks. This work utilizes the Internet of Things framework to facilitate public and commercial waste retrieval in cities by using an ultrasonic sensor and a low-cost, low-power, system on a chip microcontroller with integrated dual-mode Bluetooth ESP32 series to check the status of the waste receptacle. A city project needs to be feasible, modular, and able to help the city's inhabitants. The various components for this implementation are analyzed to determine the efficacy of each element. After analyzing the different networking capabilities of the networked modules, the Bluetooth Low Energy (BLE) option is considered the best practical option for the connection of the networked modules. Though it has a short range, Bluetooth Low Energy's most prominent feature is power saving, where energy is the primary constraint. The hardware used in the implementation (ESP32), makes a significant difference, especially regarding

the modularity that can be obtained in the future for this implementation. A city is constantly expanding and changing. So, any implementation or project that is put into use in a town needs to be able to change and grow with the city. The chosen microcontroller makes a difference in the overall operation of the modules in the network. The impacts and repercussions can be quantified and analyzed by looking at the components of this implementation. The best network topology to utilize is a mesh network, allowing for the best modularity of the network implementation. This thesis shows how the Server-Client network option allows for reliable modularity and data transmission to sanitation workers' mobile devices. By looking at several aspects of this proposed smart waste management system (such as the code and hardware), this thesis ascertains whether this can be implemented for cities' use efficiently. This implementation should demonstrate the effectiveness of using the Internet of Things for city purposes and the benefits that can be gained for the environment and the people living in a city using it.

Accepted by: ______________________________, Chair
Dr. Anindita Paul

____
Dr. Cheng Cheng

____
Dr. Ahmad Zargari

Table of Contents

Table of Figures

List of Tables

1 Introduction

Smart technology has become a prevalent subject since the expansion of the Internet of Things (IoT). IoT has found a substantial number of growing uses as the connectivity of devices has become a centralized occurrence in homes. As it becomes easier to obtain smart devices for individual use, the thought of how they can be used not just for commercial use, but also for public use. To understand the implications of IoT in the scope of the size of a city, the factor that need to be understood and analyzed are efficiency, budget, and distance (or city area). This project analyzes a department in a city that blankets the busiest areas of a city, Waste Management. According to a study done by the Environmental Protection Agency (EPA) "The City of Atlanta disposed of approximately 100,000 tons of residential garbage in calendar year 2008" (IFC International, 2009). Waste management is just one of the many departments that make up a city's infrastructure, but the waste management department raises the need for an efficient and affordable, smart waste receptacle. According to a study done by the Lexington Fayette Urban County Government, commercial waste accounted for 16% of their annual trash in 2022 (RRS, 2022). By utilizing the technology available it is possible to efficiently retrieve commercial waste using this implementation.

Table 1: Fleet vehicle Age (MSW Consultants, 2017)

Equipment Type	Number of Pieces	Useful life (Years	Expected Average Age (Years)	Actual Average Age (Years)	Status
Automated Side loader	16	7	3.5	2.4	ok
Rear loader	66	10	5	11.9	severely deficient
Grapple Truck	4	7	3.5	5.5	deficient
Roll-off Truck	3	10	5	5.3	deficient
Street Sweeper	8	7	3.5	6.4	severely deficient
Medium Duty Truck	7	10	5	10.9	severely deficient
Mobile Equipment	7	10	5	7.7	deficient
Pick-up Truck	23	10	5	11	severely deficient
Sedan	11	10	5	10.5	severely deficient
Trailer	4	15	7.5	4	ok
Container	21	10	5	7	deficient
	Total 170			9.3	

Table 1 (MSW Consultants, 2017) shows the fleet information from a solid waste study conducted in Louisville in 2017. It shows that the average expected life of a vehicle is half of the equipment's useful life. It also shows that the actual average age is normally higher than the expected average age. It is observed from the table that the fleet equipment is mostly deficient and in need of repair. The proposed technology will allow for a longer useful life for the fleet vehicles by cutting down on the amount of travel for public and commercial waste removal.

Three main problems are addressed with this project: a.) coverage area, b.) efficiency, and c.) affordability. To handle the wide span of the coverage area for a city, this application uses a

low power, low-cost system on a chip microcontroller with integrated dual mode Bluetooth ESP32 series (ESP32) for the waste receptacle. To manage efficiency, the ESP32 sends a status (Empty/Full), to show whether a receptacle is ready for retrieval or not. This relieves unnecessary travel for the disposal vehicles, which can save gas and time. For the affordability factor, the components are the main factors in ensuring that this implementation does not require overhead or extra personnel to make it a reality.

1.1 Coverage area

To handle the vast area that a waste management department must cover, it was necessary to investigate a way to interconnect the waste receptacles so that the signals could be sent along the network, or at least allow for a node/receptacle to contain the information of at least one other node/receptacle on the network to make it easier to tell the status of multiple nodes on the route. By using an ESP32 microcontroller it becomes a matter of coding the controller to be both a server and a client. Interconnecting the microcontrollers into a network using the Bluetooth Low Energy (BLE) on the ESP32 allows for the creation of Bluetooth servers and clients in the network capable of noticing the characteristics of the other nodes/receptacles on the network.

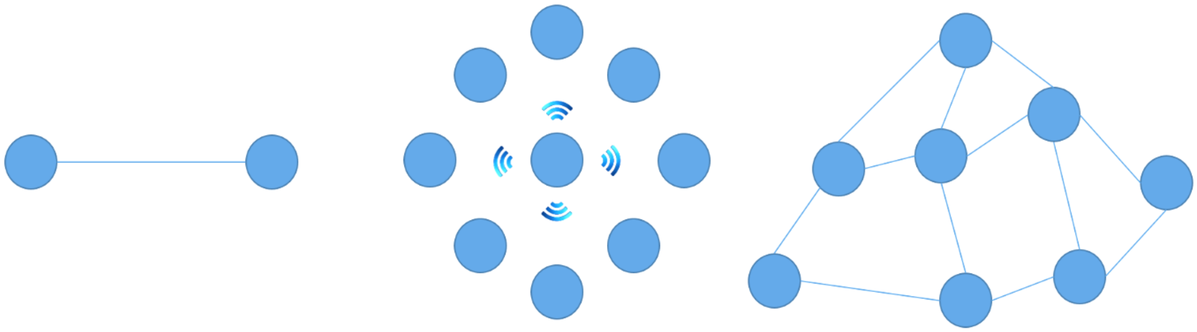


Figure 1: BLE Network Topology (Lesson 2 – BLE profiles, services, characteristics, device roles and network topology, 2023)

Figure 1 shows the distinct types of topologies that could be potentially used to accomplish the necessary goals for this project. The mesh topology is the best option because of its inherent expandability, but the one-to-many connection has almost the same amount of potential and less coding.

Linking the nodes/receptacles also has the advantage of being able to modulate the network. Due to construction and other factors, cities are constantly changing, so modularity becomes a significant factor to obtain flexibility. BLE networks may have a maximum range of approximately 150 meters (RF Wireless World, 2023) while the average city block, according to reference, is between 94.5 - 98.5 meters (Writer, 2020) so the effective range of the BLE nodes is able to cover between city blocks adequately.

1.2 Environmental Safety and Efficiency

To keep a city running efficiently, city administration must continually find ways to improve the overall operation of departments. Knowing how to save money/time from a single operation in a department can save the city thousands of dollars. Minimizing the need for lengthy travel while traversing the city can also lessen the need for large waste retrieval vehicles.

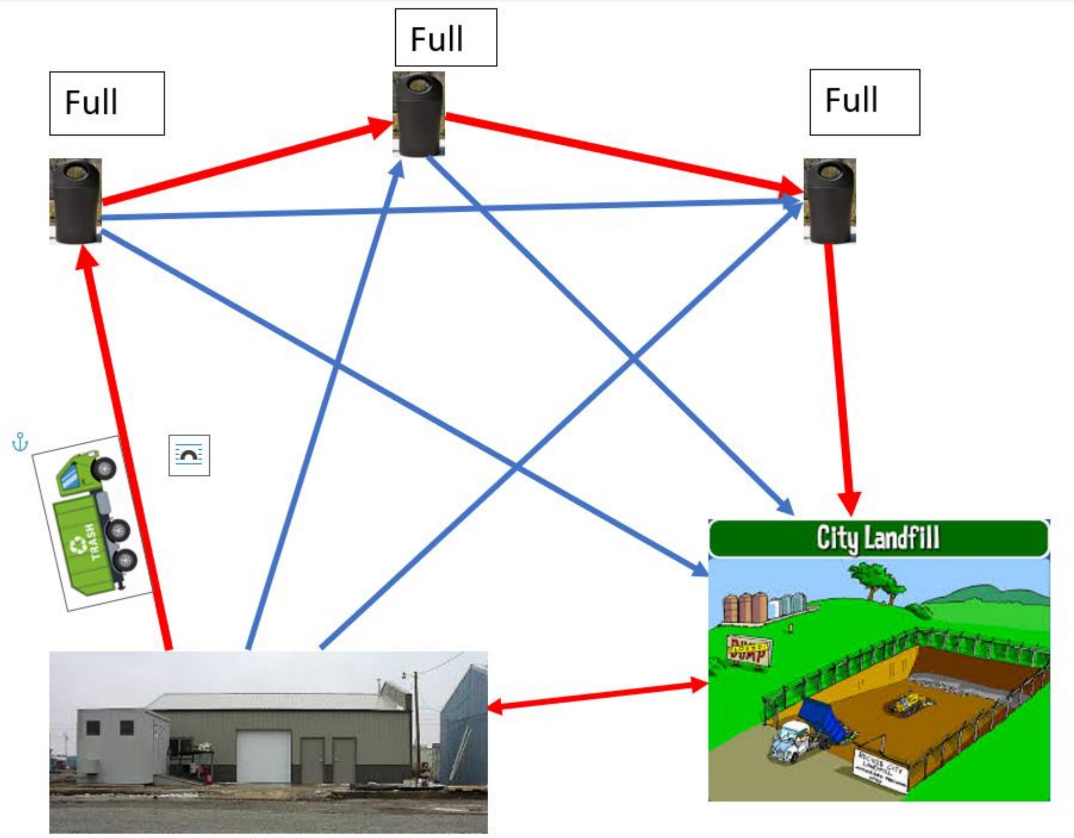


Figure 2:Route Without Optimization

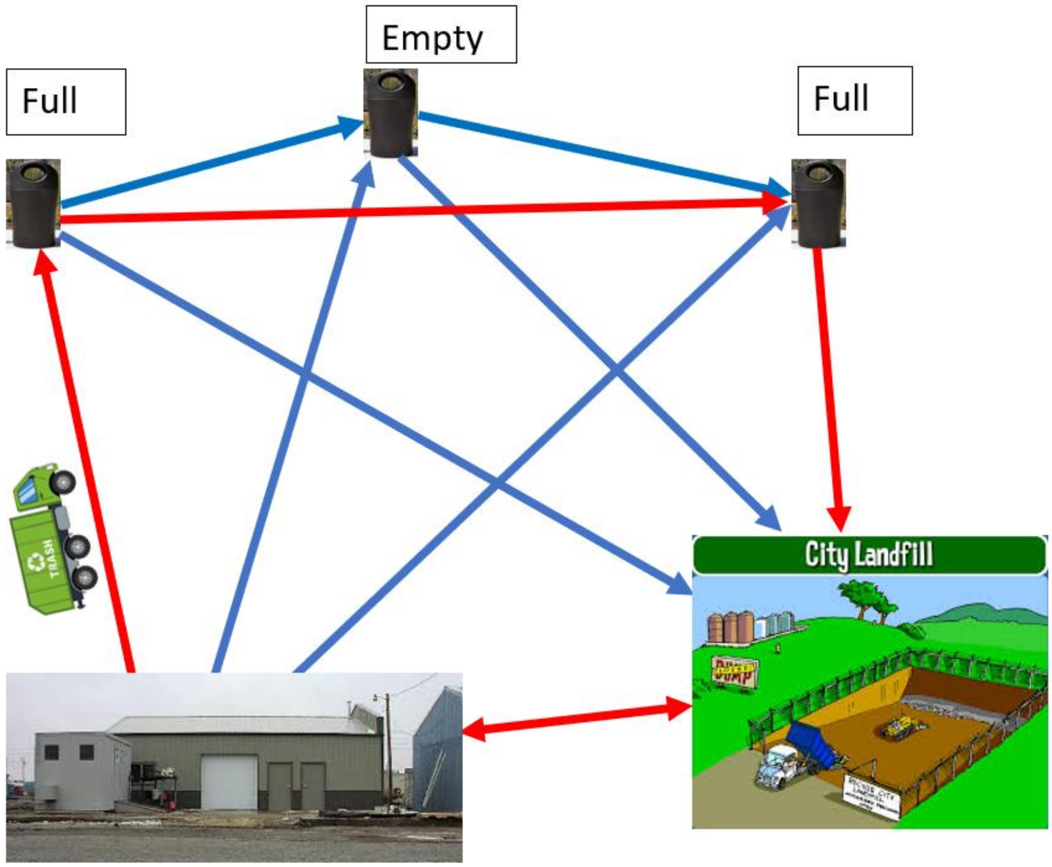


Figure 3: Route With Optimization

Though these vehicles can hold more, they are also bad on gas mileage. To streamline the efficiency factor for this, this implementation will be able to send a signal along the route to notify if a receptacle is Empty/Full. Figures 2 and 3 show how route optimization works. It looks for the best path to reach the end; which is for the waste vehicles to collect and dump the waste and then return to the garage. This will drastically optimize routes and times spent traveling. Another way to streamline efficiency is to cut down on the maintenance that must be performed. “The average age of a car on the road in the United States is around 12 years, which means that many cars require frequent repairs and maintenance” (Abdalslam, 2023).

By cutting the amount of maintenance that needs to happen on the receptacle modules, this application of smart technology becomes more feasible for public/commercial use. The way that this is achieved in this project is with a solar panel. “Solar radiation represents the largest energy flow entering the terrestrial ecosystem” (An Assessment of Solar Energy Conversion Technologies and Research Opportunities, n.d.). The use of renewable energy sources will improve the constant need to change the batteries in the device. That means if there are no technical problems that arise in the device all maintenance can be done using the software.

According to Nature.org, “A carbon footprint is the total amount of greenhouse gases (including carbon dioxide and methane) that are generated by our actions” (The Nature Conservatory, 2023). The different boundaries of a carbon footprint refer to the carbon footprint generated by not only the vehicles, but also the individuals that use the vehicles (Gao, Liu, & Wang, 2014). To be able to adequately judge how much of an environmental impact this project could have, this project will look at the carbon footprint generated by a couple of cities to project an estimated decrease in the carbon footprint due to this project. The carbon footprint for a city is more complex than the footprint for individual households. By looking at the waste management reports compiled by a couple of cities, this report should be able to see the carbon footprint for those respective cities and be able to show what aspects of the carbon footprint will be reduced by this project.

1.3 Affordability

Most state governments require that each municipal county maintain its own solid waste management department while reporting annually on the condition and status of the departments. Kentucky has a revised statute that states, “Each county shall provide a universal collection program…” (KRS 224.43-315, 2023). However, one thing that is not given in the statute is how

the counties will fund the solid waste management programs. While most states offer grants to help with the waste management program, a large amount of the budget for waste management comes from the county/city budget. For big cities like Chicago, implementation across the city is too expensive to consider without considering what the city can gain. The budget for a city is usually allocated over a fiscal year (like most businesses). A major selling point for this implementation is that the materials for this implementation are very affordable, even for a graduate student. The main components of this implementation are inexpensive or singularly created by the researching individual. The major components for the implementation were all bought using the most affordable options available. An itemized budget for the implementation can be found in the results chapter. By creating the implementation using the parts listed below, it will not only be affordable, but will require little maintenance for each waste receptacle.

2 Literature review

The rising solid waste worldwide has created a need for waste management solutions. Over two billion tons of municipal solid waste (MSW) are generated each year worldwide (Hwhenvironmental, 2023). To find a solution to the growing waste management issues, many scholars have taken to utilizing the prominent IoT solutions that have become prevalent in smart homes. Three main components are utilized in each instance: a) a microcontroller, b) one or more sensors, and c) a data transmission/signal relay. Each solution has its own benefits and, at the same time, some drawbacks.

Design of smart Trash bin (Chandrea & Tawami, 2020) discusses the impact of smart trash bins. The implementation of the smart trash bin used in this article contained an Arduino UNO microcontroller, HC-SR04 ultrasonic sensor, Pyroelectric InfraRed sensor (used to detect infrared radiation and heat generation), and a voice module. This implementation uses several sensors to make the implementation more people-friendly, but the implementation itself is geared more towards a home application instead of mass implementation. The implementation has no signal relay because its primary goal is to help change people's mindset about disposing of trash properly.

A Smart Waste Management Solution Geared Towards Citizens (Pardini, et al., 2020) tackles the challenge of creating an efficient and real-time waste management system. The hardware used for the implementation are a load sensor (HX711), ultrasonic sensor (HC-SR04), temperature and humidity sensor (DHT11), a GPS module (model Neo-6M), and a GSM module (SIM900). These sensors all work with solar panels and an Arduino-based microcontroller. This implementation's main selling point is focused on using a cellular network coupled with a custom application designed for smart waste receptacles. The biggest problem with this implementation is the inability to conceal the wiring for implementation.

Smart Bin Implementation for Smart Cities (Sharma, Singha, & Dutta, 2015), researches the potential smart bin applicable in a smart city. By using data analytics and machine learning, this implementation analyzes the algorithm used for route optimization. The hardware for this implementation was a simple microcontroller (PIC-16F73), an ultrasonic sensor (HC-SR04), and a GSM module (SIM-900A). By reviewing a database of the data collected and analyzing the data with a predictive algorithm, the researchers could gather data accurately and show the pattern. The microcontroller for this implementation is lacking in both digital pins and built-in functions. Still, it was able to complete the necessary task with the help of several external devices. However, it does not have all the bells and whistles of the Arduino UNO or ESP-32 modules; it still serves its functional purpose.

Smart Dustbin for Waste Management (Gayanthika, Maduranga, Silva, Wikramarathne, & Ranasinghe, 2019) discusses the use of smart dustbins to improve living conditions in Sri Lanka. This implementation utilizes an Arduino Mega 2560 microcontroller, Ultrasonic sensor (HC-SR04), load cell, RFID reader, and GSM module. The design and implementation are developed to allow user data to be pulled from a server, but this idea is not efficient for a city use because of the tourist flow that comes through a city. The full functionality of the system needs about 500mA to 700mA (Gayanthika, Maduranga, Silva, Wikramarathne, & Ranasinghe, 2019). One of the primary goals of this implementation was to categorize waste , which is only necessary if you are looking for patterns to dispose of waste.

The Design and Implementation of the Smart Trash Can based on the Internet of Things (Wang, Xu, Zhang, Zhang, & Su, 2020) investigates the development of an intelligent trash can primarily for use in individual homes. The implementation utilizes MSP430F149 as the main control chip and uses the human body sensor module, infrared ranging module, motor, ultrasonic

module, and GSM module as peripheral components. For open-air applications some of the components used in the development of this implementation would be unnecessary, but when used in an individual home it becomes necessary. The application for waste detection in a public setting would not be necessary unless you were trying to locate public patterns or trends. A major flaw of this implementation was the visible wiring that could be damaged by liquid waste.

The Eco-Smart Can V2.0 (Nanto, 2019) looks to solve the problem of efficiency in the waste management system by being able to tell whether a receptacle is full or empty to cut down on wasted time by waste management personnel. The hardware utilized for this implementation includes ESP32-WROOM-32, HC-SR04 ultrasonic sensor and uses the built-in Wi-Fi of the ESP32 for data transmission. The ESP32 is the ideal micro-controller because it contains most of the necessary elements for the device without having to have too many peripheral modules. The biggest issue with this implementation is the exposed wiring; even with a custom-made chassis, the wiring from the microcontroller to the ultrasonic sensor is still exposed.

After looking at the relevant research is apparent that the previous implementations have flaws, which this design and implementation will remedy. The main thing that will be accomplished with this implementation is the data transmission network.

3 Methodology

To achieve the goals of affordability, environmental safety, efficiency and coverage, there are two parts that form the actual modules for the waste receptacle implementation. The hardware, which is the physical components that help to make up the primary functionality of the implementation. Secondly, the software which is the coding and program behind the hardware including the network construction for the multiple connected ESP32 devices.

3.1 Hardware

The hardware used in this implementation are listed in Table 2. Each piece is selected to meet the goals of the project and create a low power and sustainable implementation. After weighing all the potential options, the equipment listed is the most affordable option for the limited budget of this project. Each entry listed in the table has a corresponding data sheet in the appendix.

The ESP32 used for the implementation contains all the necessary I/O ports and networking capabilities embedded in the microcontroller. The necessary power required to power the ESP32 is approximately 3.3V. Though it is possible to add more sensors, in the interest of the limited time frame, a single sensor to measure the height of the waste in the receptacle is used. The HC-SR04 ultrasonic sensor is a common and inexpensive peripheral device that uses sonar waves to measure the distance from the sensor. To help with the sustainability of the implementation, a solar cell is used in conjunction with a rechargeable 3.7V battery to power the entire implementation. To keep the battery from overheating/overcharging, an intermediary charging module (TP4056/TC4056C) is used here.

Table 2: Hardware List

Hardware	Description
ESP32	KeeYees ESP32S ESP32 Development Board 2.4 GHz Dual Core WLAN Wi-Fi + Bluetooth 2-in-1 Microcontroller ESP-WROOM-32 Chip for Arduino (38 pin narrow) (Appendix A)
Ultrasonic Sensor (HC-SR04)	Sonar distance sensor offering a range of up to 4m (Appendix B)
Solar Panel	Fielect 5V 1.2W Polycrystalline Mini Solar Panel Module DIY for Light Toys Charger 110x69mm 1Pcs
Rechargeable 3.7V lithium-ion battery	3.7V Batteries, Pre-Charged
Charging Module	TP4056 / TC4056A Lithium Battery Charger and Protection Module (Appendix D)

3.2 Complete Circuit and Code Description

To implement the intended design a battery charging circuit is designed to power the entire unit and a microcontroller-based sensor network is developed for the efficient data transfer within the cities' network. The dc power supply utilizes an alternative power source to cut down the frequent regular maintenance by using a renewable energy source which curbs the need for the replacement of batteries.

The charging circuit consists of three main parts, the solar panel, the battery and the TP4056 charging module. The TP4056 serves as an intermediary to control the electrical flow to the battery and send power to the main module. Though the TP4056 is a small component the purpose that it serves is to maintain the overall health of the battery by preventing overcharge and overheating. The TP4056 has many features some of which include protecting the battery from inrush currents using a soft start protection feature and over discharge protection which prevents

the battery from being discharged below 2.4V (Richards, 2020). These features make this component necessary.

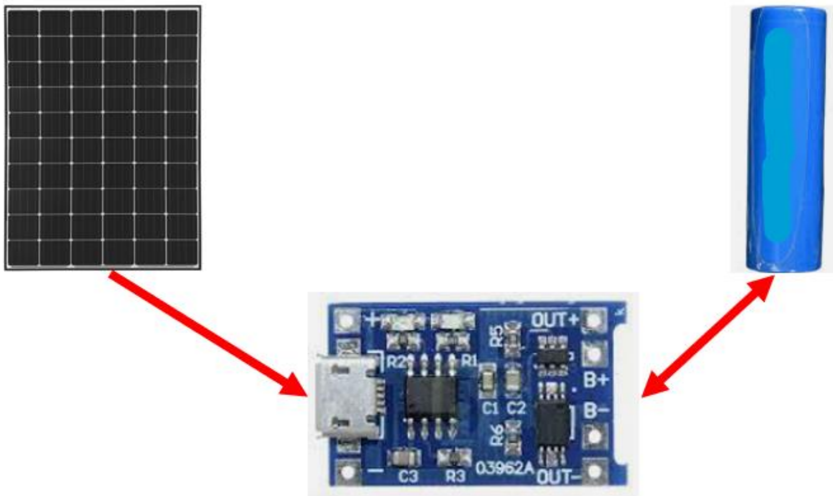


Figure 4: DC Power Supply circuit diagram

Figure 4 displays the basic diagram for the flow of power through the charging circuit. The solar panel charges the 3.7V battery through the TP4056 charging module which is responsible for both pulling energy from the battery and charging the battery utilizing the input from the solar panel. The main layout for the TP4056 is described in the datasheet in Appendix D.

The complete circuit is comprised of the charging circuit, ESP32 microcontroller and the Ultrasonic sensor. The charging circuit of Figure 4 sends power from the TP4056 to the ESP32 microcontroller and the ultrasonic sensor (HC-SR04). The ESP32 is then programmed to send the status of the waste receptacle over BLE to the connected device(s). For this implementation, the device that receives the status is the mobile application. A cell phone is used here to show not only the status, but also the amount full.

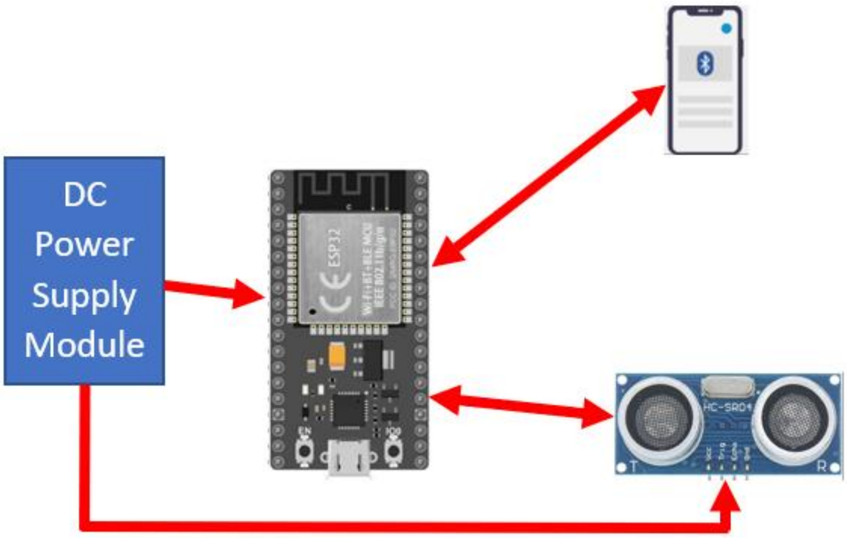


Figure 5: Complete circuit diagram

The basic layout shown in Figure 5 depicts the basic flow of signals for both power and information. The flow for the signals between the ESP32 and the ultrasonic sensor is two-way. The information that is gathered by the ultrasonic sensor is sent back to the ESP32. ESP32 sends the current status of the bin to the mobile devices based on the received sensor data.

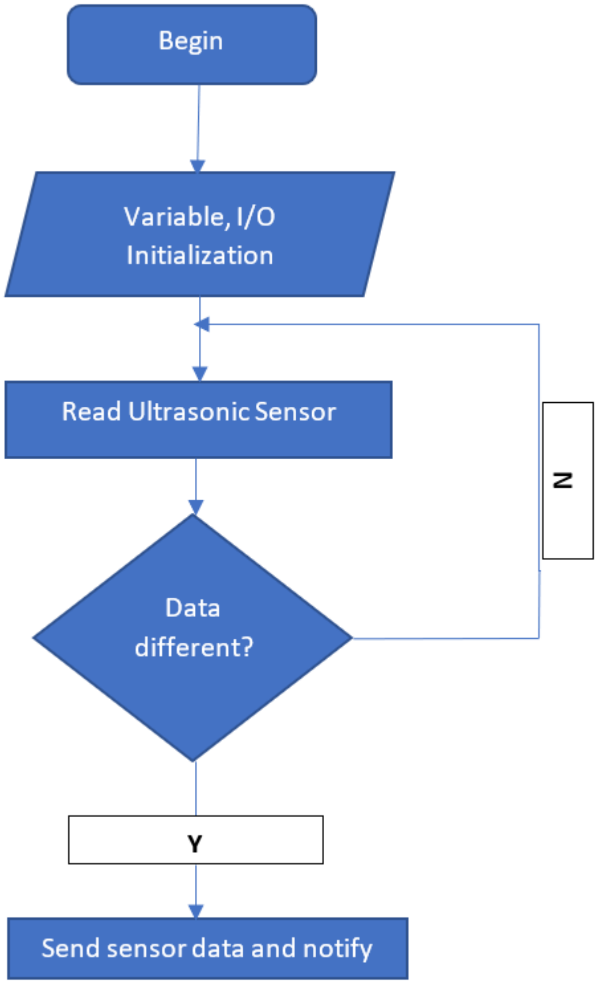


Figure 6: Basic Code Flowchart

A code was developed to properly execute the operation of the circuitry. The code is used for the data processing and data relay to the relevant device(s) using Bluetooth low energy (BLE). The implementation model uses a server client model which allows for the interconnectivity of nodes and creates a star topology. The core of the code follows the flowchart in Figure 6. After the initialization of the serial and I/O ports, the signal generated by the sensor is transmitted to the ESP32. Once the ESP32 receives the data it is then transmitted via Bluetooth low energy to the connected device(s). By creating a specific characteristic for the waste receptacle status, the code simply updates the value of the status characteristic. This automatically updates sanitation

worker's mobile devices. The data flow in real time, showing the change in data for the waste receptacles.

The pseudocode in Figure 7 shows the basic strategy for updating the characteristic and displaying the characteristic in the serial monitor. The characteristic is given the NOTIFY property which allows for notifications about the update of the status to be sent to the connected device(s). The full code for the implementation can be found in Appendix E.

```
BLE characteristic status

if distance < 5
print FULL
update status variable
if distance > 32
print EMPTY
update status variable
else
print HALF FULL
update status variable
```

Figure 7: Sensor Pseudocode

4 Results

The overall circuitry for the implementation in figure 8, shows that the main power comes from the 3.7 V battery which can keep the implementation running for approximately 24 hours without the solar panel recharging. With the help of the solar panel, recharging the battery, the implementations battery life more than doubles. This can be seen to extend the overall life of the battery as well so that the maintenance requirements for the implementation's battery are minimalized. The basic flow of the power from the TP4056 can be seen in figure 8, which shows that all power for the main implementation flows through the TP4056, but the ESP32 pushes the power to the ultrasonic sensor. Figure 9 shows the experimental setup created from the diagram in Figure 8. More photos of the experimental design can be found in Appendix F.

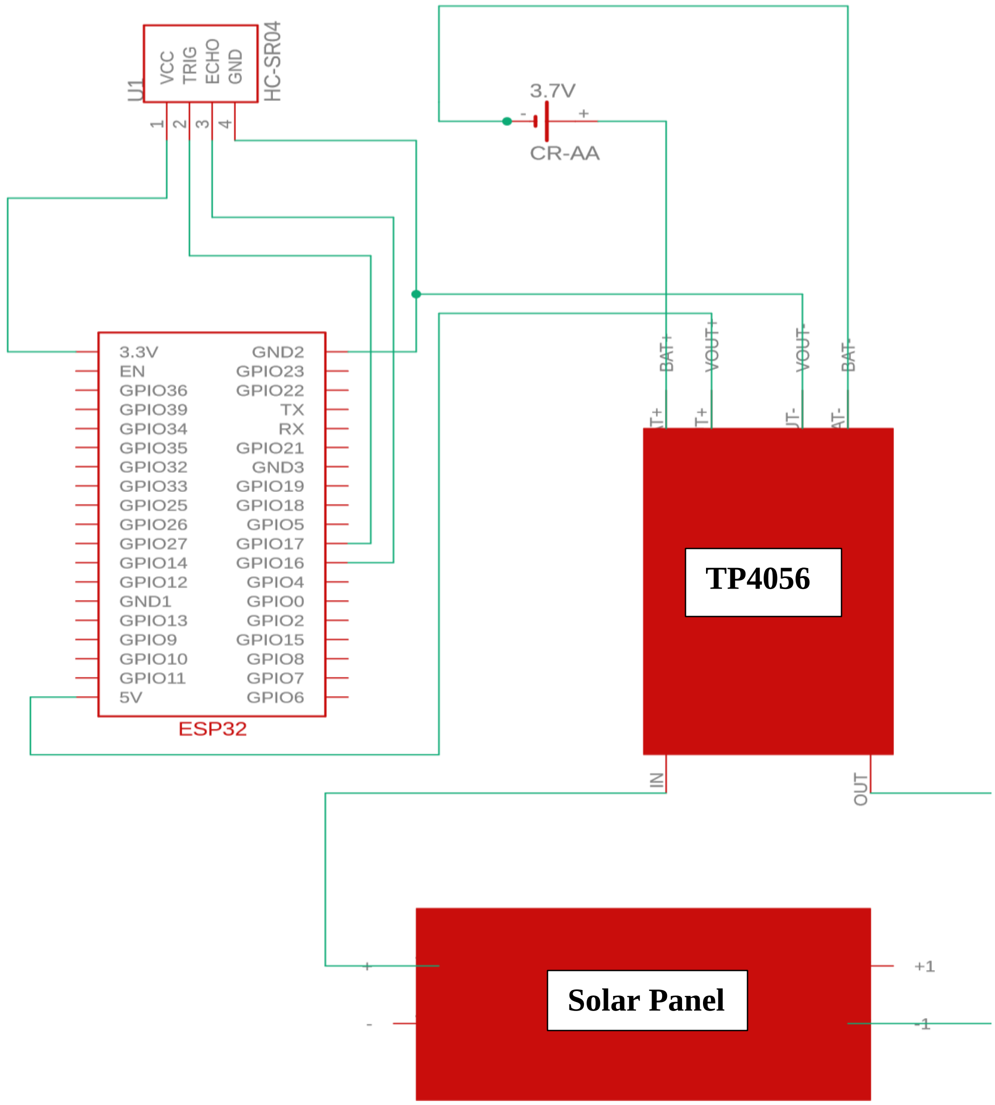


Figure 8: Circuit Diagram

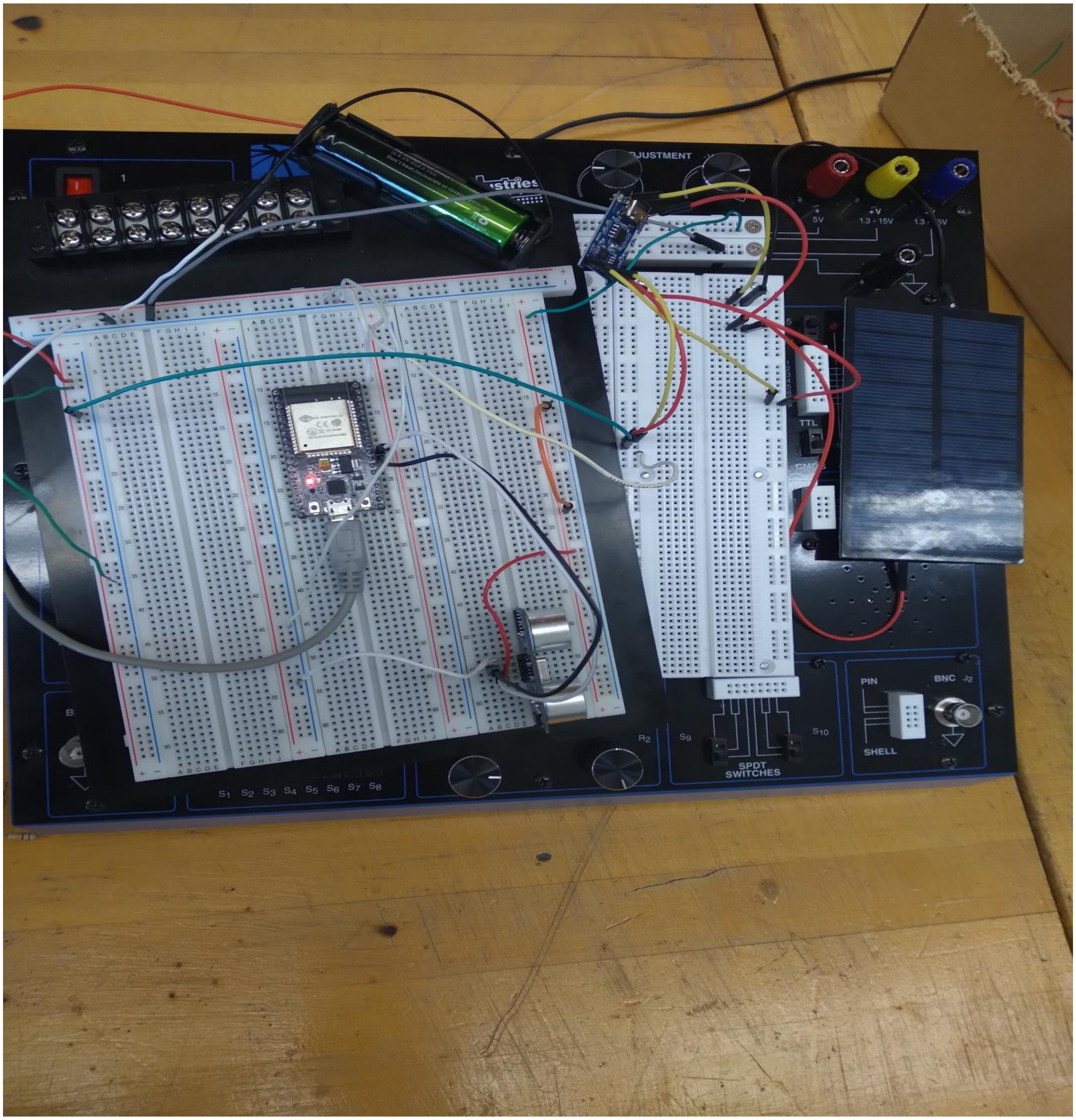


Figure 9: Experimental Setup

The code does all the basic functions (reading/writing data to the BLE and power relay) necessary for the implementation to function without any bugs. The overall functionality as outlined in the code has three status updates that are relayed via BLE to the connected devices: a) Empty, b) Full, and c) Half Full. The various updates are obtained by continually checking the

distance with the ultrasonic sensor and the implementation status is updated in real time on the connected devices. Figure 10 shows the three status options for the modules along with the output for the statuses in the sanitation workers' device.

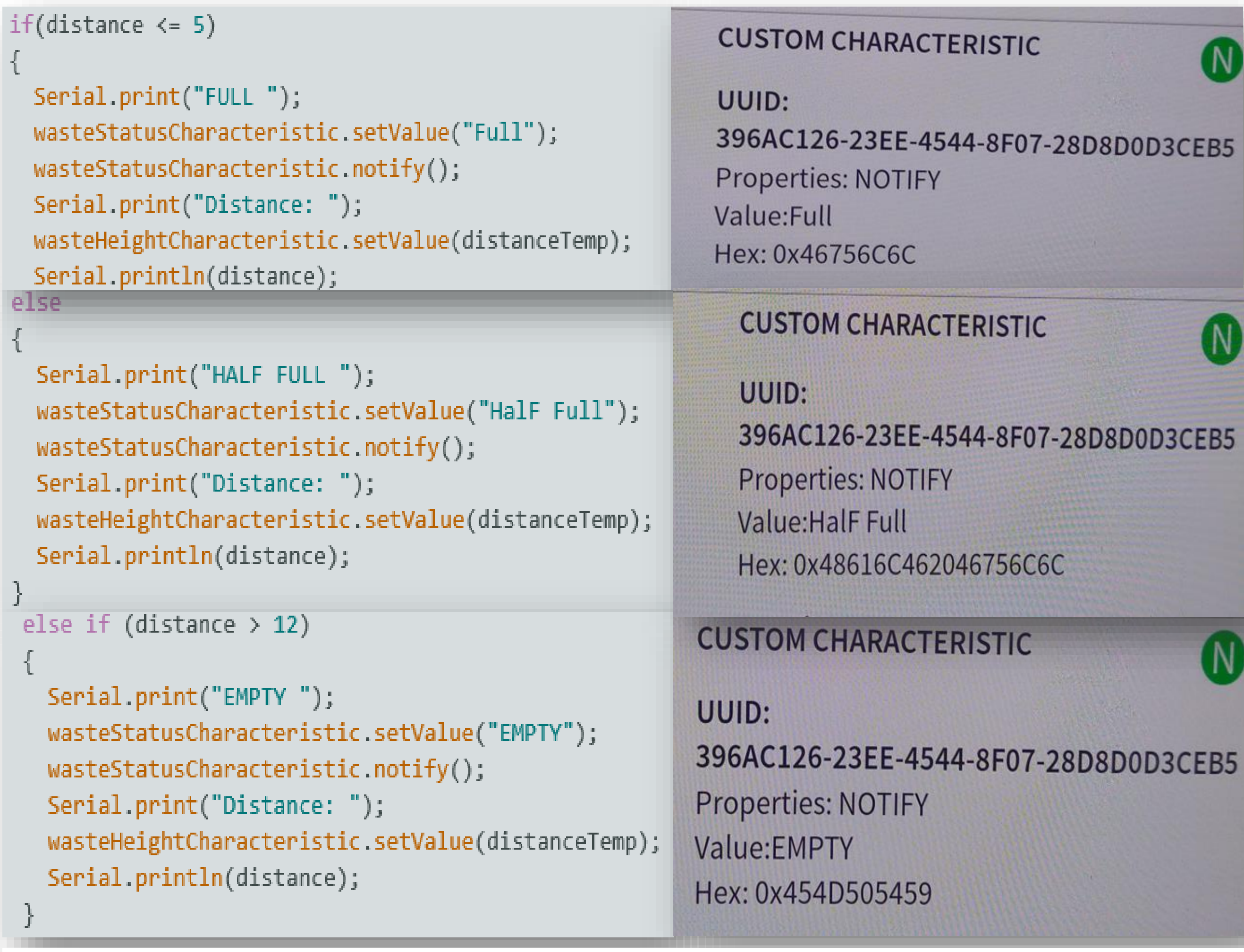


Figure 10: Relevant code and associated output

Figure 10 shows that the code does all the intended functions while maintaining the low energy standard necessary. Thus, the result of the implementation shows that the implementation is fully functional and has very little drawback other than the short range.

Table 3 : Pricing Table

Component Name	**Description**	**Price**
ESP-32	Development Board Dual Core WLAN Wi-Fi + Bluetooth 2-in-1 Microcontroller	$7 (rounded up)
Ultrasonic Sensor (HC-SR04)	Sonar distance sensor with a range of up to 4m	$8 (5 pk approx. $1.60)
Solar Panel	5V 1.2W solar cell	$8 (approx. rounded up)
3.7V battery	Lithium ion 3.7V, pre-Charged	$9 (approx. rounded up)
Charging module	TP4056 Lithium-ion charger	$2 (approx. rounded up)
Arduino IDE	Programming IDE	FREE
Total:		$27.60

The overall pricing is reflected in table 3 which shows that the overall budget for this implementation stays in a reasonable price range. The overall estimated budget for one implementation module is approximately $27.60. The original goal was to find an affordable method. This goal has been achieved, but if you factor in the creation of more than fifty modules

the price begins to reach thousands of dollars, but this only covers the modules themselves. The cost for rollout in a city will depend on the city and the amount of modules needed for the project.

The components are easy enough to obtain, but the resources required for a feasible utilization of this implementation are personnel and mobile devices. Some cities already have mobile devices issued to several city departments, like Lexington has an electronics recycling program that was projected to cost $230,000 (Kentucky Department for Environmental Protection, Division of Waste Management, 2018-2022). In most cases cities have an allotment for the provision of mobile devices to municipal employees; Louisville for instance has a cell phone program that in 2017 had about 900 phones which cost the city about $528,000 annually (Ryan, 2017). These mobile phones are allotted to multiple departments in the city including Public Works (which houses the waste management department). The recent COVID crisis has caused many municipal departments to switch to a more remote option, and because this implementation utilizes BLE instead of mobile networks the devices necessary for application only need to be Bluetooth capable. Roughly six-in-ten U.S. workers say their jobs can mainly be done from home and (59%) are working from home all or most of the time (KIM PARKER, 2022). A Bluetooth capable mobile device can be purchased at a reasonable price such as the Blu® View 2 which can be purchased for $29.99 (excluding tax) from simple mobile (Simple Mobile, 2023). Though the mobile devices are necessary for the functionality of the implementation, only the sanitation workers that will be out on a route will need them which will curb some of the losses accrued from misuse or loss of the mobile devices; the devices would simply be returned once the sanitation workers return from their route.

The personnel that will manage the functionality of the implementation will be run from the municipal office of Information Technology (OIT), which manages the technology needs for

the city. The major functionality of the devices is inherent within the code, so that there is little need for routine maintenance. The creation of the implementation can be outsourced to experiential learning programs such as the Craft Academy at Morehead State University. By allowing programs like Craft Academy to create the devices, the students will gain experience and knowledge while working on a real-world application. A partnership with these types of programs could allow for a mutually beneficial arrangement that could bring publicity to the program and also allow the city to create the needed amount of device while keeping the cost of creation negotiable.

5 Conclusions

The results show that the implementation does all the functions necessary to be considered successful. The objective of the implementation was to show the status of the receptacle from a distance to make waste removal more efficient. The solution to the problem was to create a way for the waste receptacle to communicate its status (Full, Half Full, Empty) with the sanitation workers ahead of time; thus, limiting unnecessary travel. If an implementation like this were to be introduced into a city the useful life of sanitation vehicles could be improved, which could eliminate several concerns especially as far as the required maintenance is concerned.

The previous work shows that though many implementations have been attempted/created there is still a lot of improvement needed before it is applicable for city use. The main problems that were encountered with the creation of this application stem from the network model creation and coding. The network model that would have been preferrable was the BLE Mesh because of its many features, such as low power and modularity. The BLE Mesh model at present can only be created using the Espressif IDE because of the many header files that have to be included in the project. After trying to create the network model in several other IDEs, to no avail, it was realized that the only program that could sufficiently run the code for the BLE Mesh was the Espressif IDE. The Espressif IDE requires learning several things that are not in basic code which requires a great deal of time. In the future, a BLE Mesh will be created. If this project could incorporate a fully functional BLE Mesh, and a custom chassis for the implementation, it would then be relatively complete.

This application shows great promise in that it will be able to help streamline the efficiency of public and commercial waste removal while being more affordable than most city plans to help reduce waste removal time. The main goals of the implementation were to find an affordable,

modular, and efficient method for public and commercial waste removal. The implementation can be created costing no more than thirty dollars, which has achieved the goal of an affordable implementation. The network model allows for the modularity of the project and the signal relay allows the project to make public and commercial waste collection more efficient.

In conclusion, this work has shown that IoT can be used in many beneficial ways, not just in civilian applications, but also for commercial or public use, though there are many things to consider when building a project on such a large scale. This project shows that IoT can help to eliminate many big issues, while maintaining a low budget which is best for public projects.

6 Future Work

In the future, the best course of action would be to upgrade the code and create a final chassis for the implementation. The software design for this implementation would use a BLE Mesh topology to use each node in the network as a receiver of information and a sender. The Mesh topography can also be achieved using Wi-Fi, but BLE is a better option, due to the low power consumption. The mesh topology allows for modulation and expansion of the network, which is ideal for cityscapes such as Louisville or Atlanta. By utilizing the existing libraries and directly creating some scripts personally, the software implementation will become feasible.

The primary logic involved in the software comes from the BLE mesh networking protocols. The uses for this network protocol are immense and the various features of a BLE Mesh allow for the creation of a managed flooding network for scenarios like sensors (Espressif, 2023); which is exactly what this implementation calls for. There are two parts to logic: the network code and the sensor application code. The network code works on the nodes and the transmission layers while the application code works on the sensor data and that gets transmitted through the network layer.

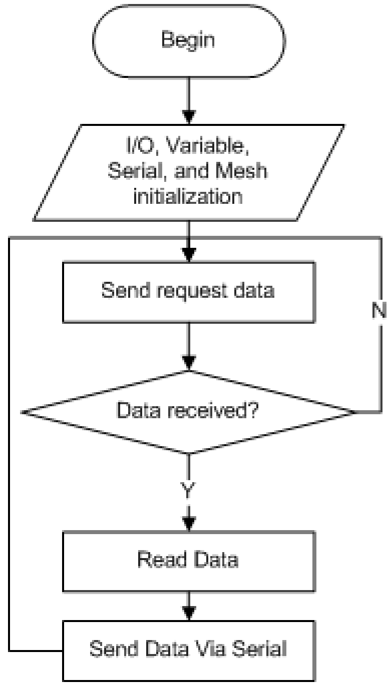


Figure 11: BLE Mesh Flowchart (Wicaksono & Rahmatya, 2021)

In Figure 11 the basic flow for the implementation of the BLE Mesh is outlined. The code begins by initializing all the serial and mesh aspects of the BLE mesh. It will create the basic server and friendship protocols needed to send and receive messages. After initializing these protocols, if there is no error then the next step will be to send out friendship requests to other nodes. BLE mesh nodes are described as being provisioned or un-provisioned, the server will provision the un-provisioned nodes into the network. For the purposes of this implementation, all the nodes are servers which are capable of provisioning nodes into the mesh. Once a node receives the request from the provisioner, the node will choose whether to connect to the mesh network. Once the un-provisioned node connects to the network, the status in the network will be updated, and the relevant data will be sent to all the connected nodes on the network.

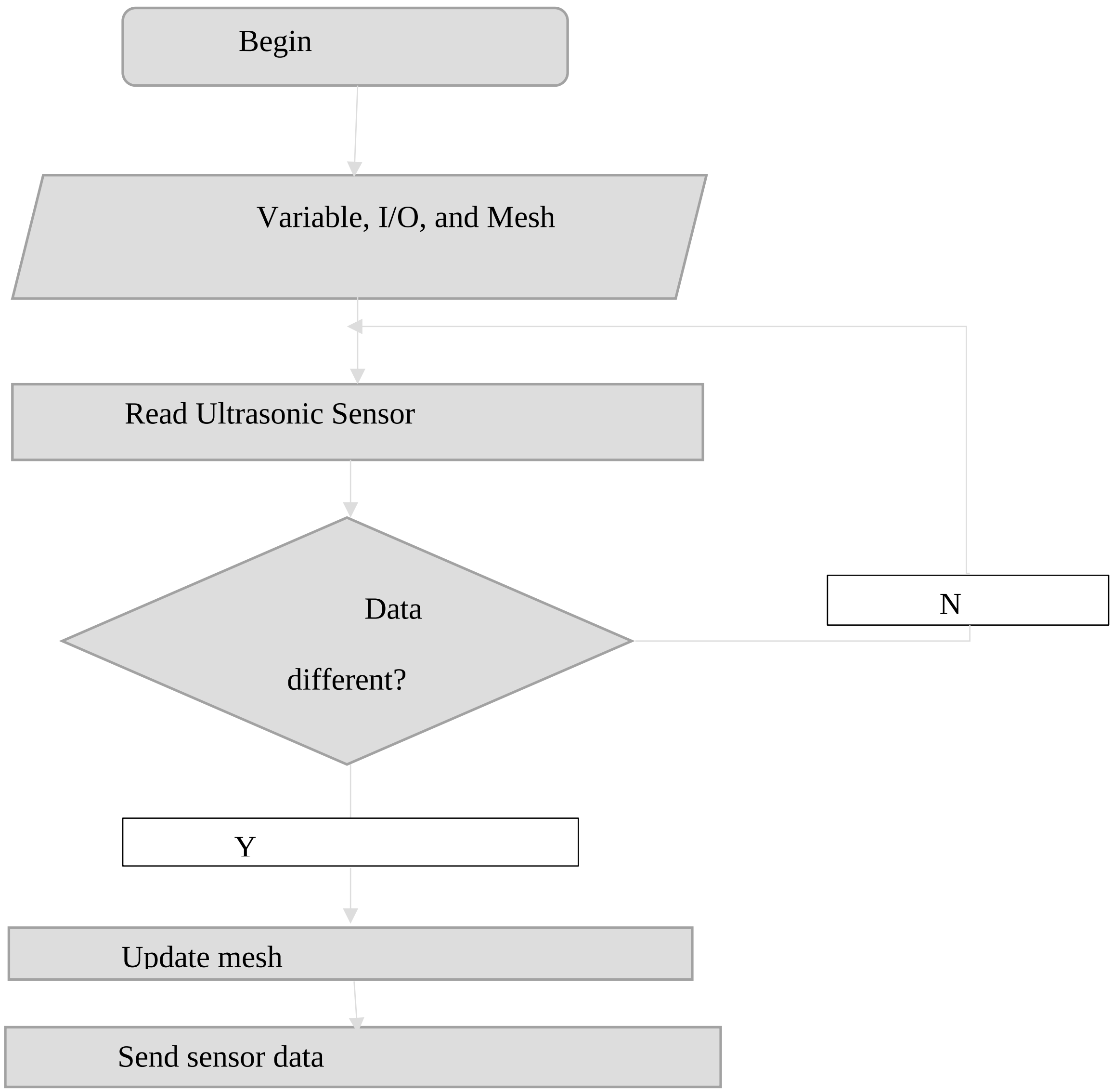


Figure 12: Sensor Flowchart

Figure 12 explains the logical flow for the code on the ESP32 and sensor(s). After the initialization of the relevant I/O ports and variables the ultrasonic sensor will initialize and continuously read and sense the waste level in the receptacle. If the data has changed from the previously stored information, then the data will be overwritten and sent along the network to all the connected nodes connected to the device in the network. Before the information gets sent out along the network, the mesh elements for the node must be updated in the configuration the element structure for the node and then sent out with the element information in a packet containing a unique identifier, the data and the receiver's unique identifier.

There are several header files that were already created for ESP-IDF. The software implementation makes use of the header files that were previously created for BLE mesh networks and adds one that was specifically designed for this implementation. By using what has already been created it lessened the burden of having to create immense amounts of code. To make the code useful in the software implementation for this project, the added header file was used in conjunction with the existing header files to add in the necessary I/O pins and variables, such as Trig Pin and Echo for the ultrasonic sensor.

A BLE mesh allows for almost endless expansion of the network and supports a low-power option. Although the code for building this option is much longer than creating a Wi-Fi mesh, the low power option is worth the extra time on the code. The short range of the BLE network may be a disadvantage, but the modularity of the network is worth it.

The implementation chassis would need to be compact and utilize the space required to keep all the wiring contained in the chassis. The initial thought is to make the chassis flat to make it harder to damage from outside forces, but by making it flat, there will be a few issues that could

arise like allowing for optimal sunlight. In the future different options and solutions can be explored to solve the potential problems with the chassis.

References

Abdalslam. (2023, March 1). *Auto Repair Statistics, Trends And Facts 2023*. Retrieved from Abdalslam: https://abdalslam.com/auto-repair-statistics.

An Assessment of Solar Energy Conversion Technologies and Research Opportunities. (n.d.). Retrieved 4 21, 2023, from Stanford University – Global Climate Change & Energy Project: http://gcep.stanford.edu/pdfs/assessments/solar_assessment.pdf.

Chandrea, R. P., & Tawami, T. (2020). Design of Smart Trash Bin. *IOP Conference Series: Materials Science and Engineering*.

Espressif. (2023, 02 16). *ESP-IDF Programming Guide.* Retrieved from ESP: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/esp-idf-en-v5.1-dev-3462-g045163a2ec-esp32.pdf.

Gao, T., Liu, Q., & Wang, J. (2014). A comparative study of carbon footprint and assessment standards. *International Journal of Low-carbon Technologies, 9*(3), 237-243. doi:10.1093/ijlct/ctt041.

Gayanthika, W., Maduranga, G. K., Silva, A. I., Wikramarathne, S. D., & Ranasinghe, R. M. (2019, 01). Smart Dustbin for Waste Management. *International Journal of Environmental Science and Development, 10*, 118-121. doi:10.18178/ijesd.2019.10.4.1159.

Hwhenvironmental. (2023, April 18). *Facts and Statistics About Waste For 2023*. Retrieved from Hwhenvironmental: https://www.hwhenvironmental.com/facts-and-statistics-about-waste.

IFC International. (2009, December). *The City of Atlanta, Georgia Smart Waste Management - ilsr.org.* Atlanta.: U.S. EPA Green Wast Solutions. Retrieved from Institute for Local Self Reliance.

Kentucky Department for Environmental Protection, Division of Waste Management. (2018-2022). *https://www.lexingtonky.gov/departments/waste-management.* Retrieved from https://www.lexingtonky.gov: https://drive.google.com/file/d/1DmIjAtkDhU-_si7EywenbWQguCKwHTXw/view.

KIM PARKER, J. M. (2022, February 16). *COVID-19 Pandemic Continues To Reshape Work in America*. Retrieved from https://www.pewresearch.org/: https://www.pewresearch.org/social-trends/2022/02/16/covid-19-pandemic-continues-to-reshape-work-in-america.

KRS 224.43-315. (2023, April 12). Retrieved from Kentucky Revised Statute: https://apps.legislature.ky.gov/law/statutes/statute.aspx?id=45875.

Lesson 2 – BLE profiles, services, characteristics, device roles and network topology. (2023, March 30). Retrieved from Embedded Centric: https://embeddedcentric.com/lesson-2-ble-profiles-services-characteristics-device-roles-and-network-topology.

MSW Consultants. (2017). *SOLID WASTE MANAGEMENT SYSTEM EVALUATION.* Louisville Metro Government.: Louisville Metro Government.

Nanto, D. B. (2019). The Eco-Smart Can V2.0. *Electronic Theses and Dissertations Paper 3589*, 73.

Pardini, K., Rodrigues, J. J., Diallo, O., Das, A. k., de Albuquerque, V. H., & Kozlov, S. A. (2020). A Smart Waste Management Solution Geared towards Citizens. *Sensors*, 2380.

RF Wireless World. (2023, January 22). Retrieved from Advantages of BLE (Bluetooth Low Energy) | disadvantages of BLE (Bluetooth Low Energy): https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-BLE-Bluetooth-Low-Energy.html.

Richards, E. (2020, February 18). *TP4056 Linear Lithium Ion Battery Charging Module*. Retrieved from Microcontrollers Lab: https://microcontrollerslab.com/tp4056-linear-lithium-ion-battery-charging-module.

RRS. (2022). *Waste Characterization Study.* Retrieved from https://www.lexingtonky.gov/: https://www.lexingtonky.gov/departments/waste-management.

Ryan, J. (2017, February 1). *Should Louisville Metro Provide Cell Phones To Employees?* Retrieved from Louisville Public Media: https://www.lpm.org/news/2017-02-01/should-louisville-metro-provide-cell-phones-to-employees.

Sharma, N., Singha, N., & Dutta, T. (2015, September 1). *Smart Bin Implementation for Smart Cities*. Retrieved from semantic scholar: https://www.semanticscholar.org/paper/Smart-Bin-Implementation-for-Smart-Cities-Sharma-Singha.

Simple Mobile. (2023, April 12). *Blu® View 2*. Retrieved from Simple Mobile: https://shop.simplemobile.com/shop/en/simplemobile/phones/simplemobile-vzn-blu-view-2-b131dl.

The Nature Conservatory. (2023, January 21). Retrieved from The Nature Conservatory: https://www.nature.org/en-us/get-involved/how-to-help/carbon-footprint-calculator.

Wang, Y., Xu, Y., Zhang, B., Zhang, J., & Su, X. (2020). The Design and Implementation of the Smart Trash Can based on the Internet of Things. *Journal of Physics: Conference Series*, 022003.

Wicaksono, M. F., & Rahmatya, M. D. (2021). Smart Cluster Housing Monitoring with ESP32, ESP-Mesh and Django. *Journal of Engineering Research*.

Writer, S. (2020, March 31). *Reference*. Retrieved from How Many Feet Are in a City Block: https://www.reference.com/science-technology/many-feet-city-block-3cd5b079ba790626.

Appendix A: ESP32 data sheet

1 Overview

ESP32-WROOM-32 is a powerful, generic Wi-Fi + Bluetooth + Bluetooth LE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.

At the core of this module is the ESP32-D0WDQ6 chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz. The chip also has a low-power coprocessor that can be used instead of the CPU to save power while performing tasks that do not require much computing power, such as monitoring of peripherals. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, Hall sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S and I2C.

Note:

* For details on the part numbers of the ESP32 family of chips, please refer to the document *ESP32 Datasheet*.

The integration of Bluetooth®, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is all-around: using Wi-Fi allows a large physical range and direct connection to the Internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 μA, making it suitable for battery powered and wearable electronics applications. The module supports a data rate of up to 150 Mbps, and 20 dBm output power at the antenna to ensure the widest physical range. As such the module does offer industry-leading specifications and the best performance for electronic integration, range, power consumption, and connectivity.

The operating system chosen for ESP32 is freeRTOS with LwIP; TLS 1.2 with hardware acceleration is built in as well. Secure (encrypted) over the air (OTA) upgrade is also supported, so that users can upgrade their products even after their release, at minimum cost and effort.

Table 1 provides the specifications of ESP32-WROOM-32.

Table 1: ESP32-WROOM-32 Specifications

Categories	Items	Specifications
Certification	RF certification	See certificates for ESP32-WROOM-32
	Wi-Fi certification	Wi-Fi Alliance
	Bluetooth certification	BOB
	Green certification	RoHS/REACH
Test	Reliablity	HTOUHTSUuHAST/TCT/ESD
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps)
		A-MPDU and A-MSDU aggregation and 0.41-*is* guard interval support
	Center frequency range of operating channel	2412"' 2484 MHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and Bluetooth LE specification
	Radio	NZIF receiver with -97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH

Not Recommended For New Designs (NRND)

Categories	Items	Specifications
	Audio	CVSD and SBC
Hardware	Module interfaces	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC, Two-Wire Automotive Interface (TWAI®), compatible with ISO11898-1 (CAN Specification 2.0)
	On-chip sensor	Hall sensor
	Integrated crystal	40 MHz crystal
	Integrated SPI flash	4MB
	Operating voltage/Power supply	3.0 V"' 3.6 V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500mA
	Recommended operating ambient temperature range	-40 °C "' +85 °C
	Package size	18 mm x 25.5 mm x 3.10 mm
	Moisture sensitivity level (MSL)	Level3

2 Pin Definitions

2.1 Pin Layout

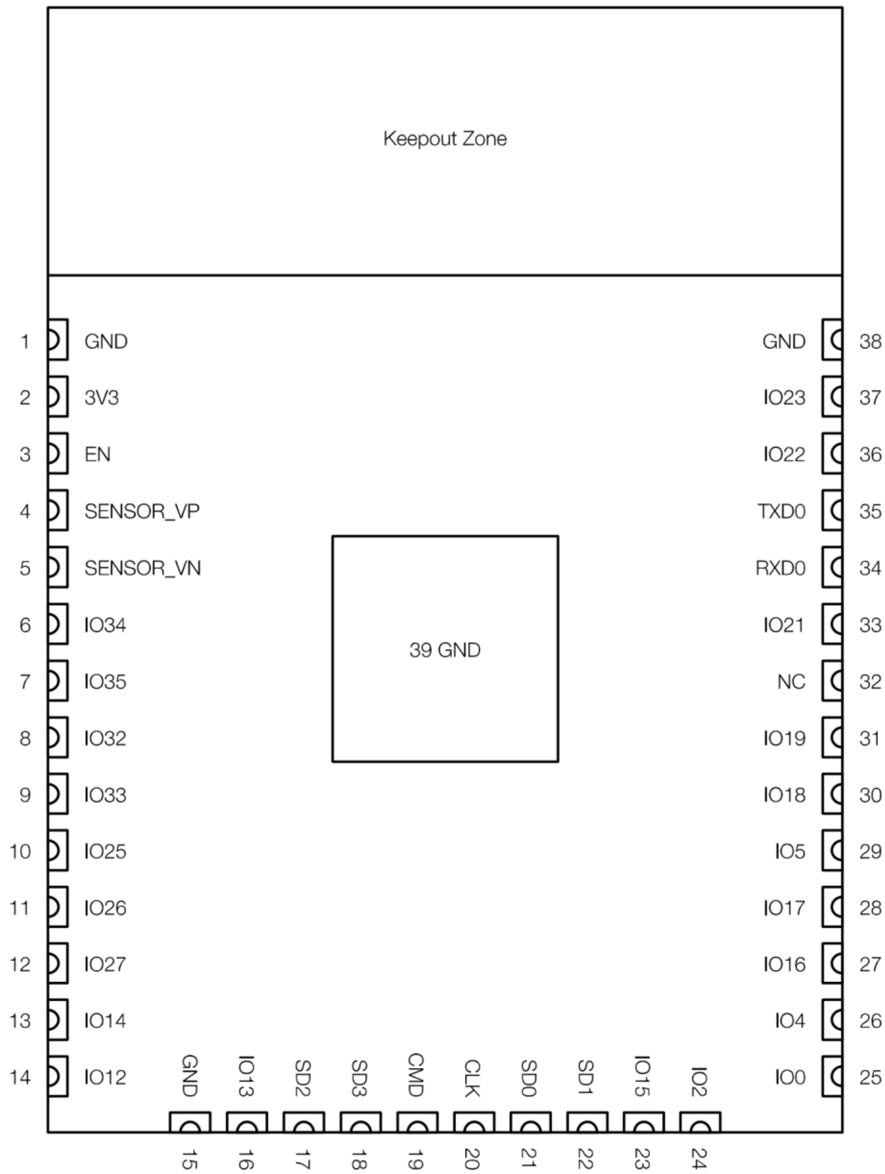


Table 2: Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.

Name	No.	Type	Function
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Appendix B: Ultrasonic sensor data sheet

Ultrasonic Ranging Module HC - SR04

- **Product features:**

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules include ultrasonic transmitters, receiver, and control circuit. The basic principle of work:

(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detects whether there is a pulse signal back.
(3) IF the signal is backed through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

- **Wire connecting direct as following:**

5V Supply

Trigger Pulse Input

Echo Pulse Output

0V Ground

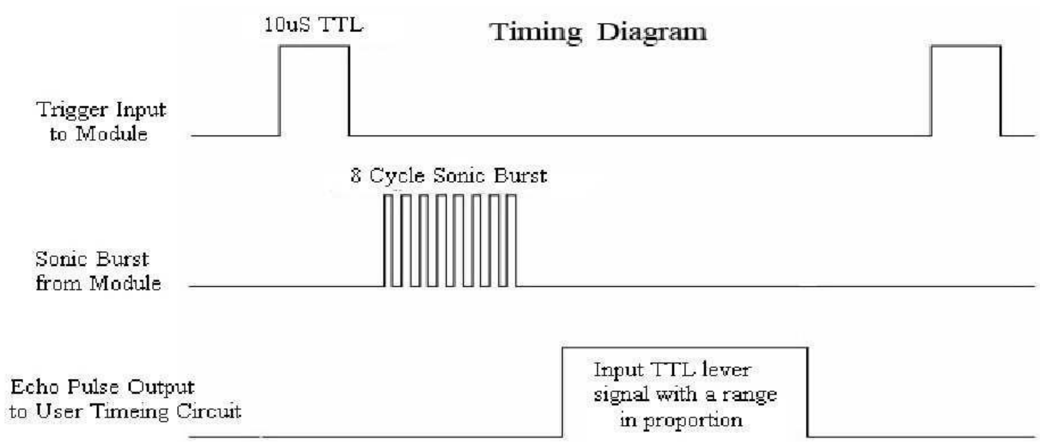
Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
Measuring Angle	**15 degrees**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Vcc Trig Echo GND

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8-cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Appendix C: Solar panel data sheet

Features:

Brand	Fielect
Material	Crystalline silicon
Efficiency	High Efficiency
Style	Compact
Maximum Voltage	5 Volts
Maximum Power	1.2 Watts
Manufacturer	Fielect
Part Number	FLT20191025T-0041
Item Weight	0.81 ounces
Package Dimensions	4.6 x 3 x 0.4 inches
Country of Origin	China
Item model number	FLT20191025T-0041
Size	5V 1.2W

Color	110mm x 69mm 5V 1.2W
Batteries Included?	No
Batteries Required?	No

Appendix D: TP4056 Charging Module Datasheet

TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8

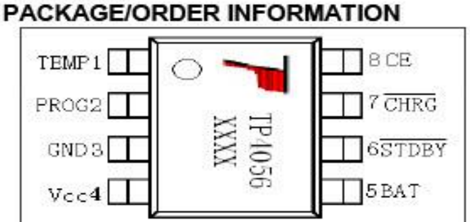
DESCRIPTION

The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries. Its SOP package and low external component count make the TP4056 ideally suited for portable applications. Furthermore, the TP4056 can work within USB and wall adapter.
No blocking diode is required due to the internal PMOSFET architecture and have prevent to negative Charge Current Circuit. Thermal feedback regulates the charge current to limit the die temperature during high power operation or high ambient temperature. The charge voltage is fixed at 4.2V, and the charge current can be programmed externally with a single resistor. The TP4056 automatically terminates the charge cycle when the charge current drops to 1/10th the programmed value after the final float voltage is reached.
TP4056 Other features include current monitor, under voltage lockout, automatic recharge and two status pin to indicate charge termination and the presence of an input voltage.

FEATURES

- Programmable Charge Current Up to 1000mA
- No MOSFET, Sense Resistor or Blocking Diode Required
- Complete Linear Charger in SOP-8 Package for Single Cell Lithium-Ion Batteries
- Constant-Current/Constant-Voltage
- Charges Single Cell Li-Ion Batteries Directly from USB Port
- Preset 4.2V Charge Voltage with 1.5% Accuracy
- Automatic Recharge
- two Charge Status Output Pins
- C/10 Charge Termination
- 2.9V Trickle Charge Threshold (TP4056)
- Soft-Start Limits Inrush Current
- Available Radiator in 8-Lead SOP Package, the Radiator need connect GND or impending

PACKAGE/ORDER INFORMATION



SOP－8

photo	
	**ORDER PART NUMBER** TP4056-42-SOP8-PP
	PART MARKING TP4056

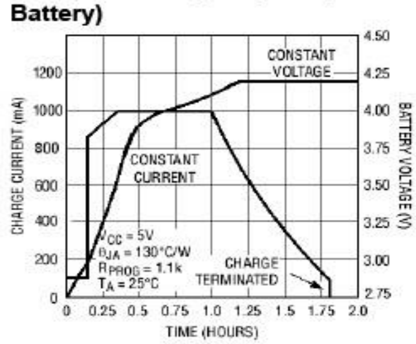
ABSOLUTE MAXIMUM RATINGS

- Input Supply Voltage(V_{CC})：-0.3V～8V
- TEMP：-0.3V～10V
- CE：-0.3V～10V
- BAT Short-Circuit Duration：Continuous
- BAT Pin Current：1200mA
- PROG Pin Current：1200uA
- Maximum Junction Temperature：145℃
- Operating Ambient Temperature Range：-40℃～85℃
- Lead Temp.(Soldering, 10sec)：260℃

APPLICATIONS

- Cellular Telephones, PDAs, GPS
- Charging Docks and Cradles
- Digital Still Cameras, Portable Devices
- USB Bus-Powered Chargers,Chargers

Complete Charge Cycle (1000mAh Battery)



TEMP(Pin 1) :Temperature Sense Input Connecting TEMP pin to NTC thermistor's output in Lithium ion battery pack. If TEMP pin's voltage is below 45% or above 80% of supply voltage V_{IN} for more than 0.15S, this means that battery's temperature is too high or too low, charging is suspended. The temperature sense function can be disabled by grounding the TEMP pin.

PROG(Pin 2): Constant Charge Current Setting and Charge Current Monitor Pin charge current is set by connecting a resistor R_{ISET} from this pin to GND. When in precharge mode, the ISET pin's voltage is regulated to 0.2V. When in constant charge current mode, the ISET pin's voltage is regulated to 2V.In all modes during charging, the voltage on ISET pin can be used to measure the charge current as follows:

$$I_{BAT} = \frac{V_{PROG}}{R_{PROG}} \times 1200 \quad (V_{PROG}=1V)$$

GND(Pin3): Ground Terminal

Vcc(Pin 4): Positive Input Supply Voltage V_{IN} is the power supply to the internal circuit. When V_{IN} drops to within 30mv of the BAT pin voltage, TP4056 enters low power sleep mode, dropping BAT pin's current to less than 2uA.

BAT(Pin5): Battery Connection Pin. Connect the positive terminal of the battery to BAT pin. BAT pin draws less than 2uA current in chip disable mode or in sleep mode. BAT pin provides charge current to the battery and provides regulation voltage of 4.2V.

$\overline{\text{STDBY}}$(Pin6): Open Drain Charge Status Output When the battery Charge Termination, the $\overline{\text{STDBY}}$ pin is pulled low by an internal switch, otherwise $\overline{\text{STDBY}}$ pin is in high impedance state.

$\overline{\text{CHRG}}$ (Pin7): Open Drain Charge Status Output When the battery is being charged, the $\overline{\text{CHRG}}$ pin is pulled low by an internal switch, otherwise $\overline{\text{CHRG}}$ pin is in high impedance state.

CE(Pin8): Chip Enable Input. A high input will put the device in the normal operating mode. Pulling the CE pin to low level will put the YP4056 into disable mode. The CE pin can be driven by TTL or CMOS logic level.

ELECTRICAL CHARACTERISTICS

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are at T_A=25℃, V_{CC}=5V, unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
V_{CC}	Input Supply Voltage		●	4.0	5	8.0	V
I_{CC}	Input Supply Current	Charge Mode, R_{PROG} = 1.2k StandbyMode(Charge Terminated) Shutdown Mode (R_{PROG} Not Connected, V_{CC} < V_{BAT}, or V_{CC} < V_{UV})	● ● ●		150 55 55	500 100 100	μA μA μA
V_{FLOAL}	Regulated Output (Float) Voltage	0℃≤T_A≤85℃, I_{BAT}=40mA		4.137	4.2	4.263	V
I_{BAT}	BAT Pin Current Text condition:VBAT=4.0V	RPROG = 2.4k, Current Mode RPROG = 1.2k, Current Mode Standby Mode, V_{BAT} = 4.2V	● ● ●	450 950 0	500 1000 −2.5	550 1050 −6	mA mA μA
I_{TRIKL}	Trickle Charge Current	V_{BAT}<V_{TRIKL}, R_{PROG}=1.2K	●	120	130	140	mA
V_{TRIKL}	Trickle Charge Threshold Voltage	R_{PROG}=1.2K, V_{BAT} Rising		2.8	2.9	3.0	V
V_{TRHYS}	Trickle Charge Hysteresis Voltage	R_{PROG}=1.2K		60	80	100	mV
T_{LIM}	Junction Temperature in Constant Temperature Mode				145		℃

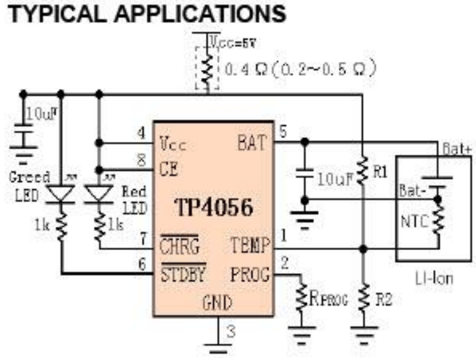
indicator light state

Charge state	Red LED $\overline{CHRG}$	Greed LED $\overline{STDBY}$
charging	bright	extinguish
Charge Termination	extinguish	bright
Vin too low; Temperature of battery too low or too high; no battery	extinguish	extinguish
BAT PIN Connect 10u Capacitance; No battery	Greed LED bright，Red LED Coruscate T=1-4 S	

Rprog Current Setting

RPROG (k)	I_{BAT} (mA)
10	130
5	250
4	300
3	400
2	580
1.66	690
1.5	780
1.33	900
1.2	1000

TYPICAL APPLICATIONS

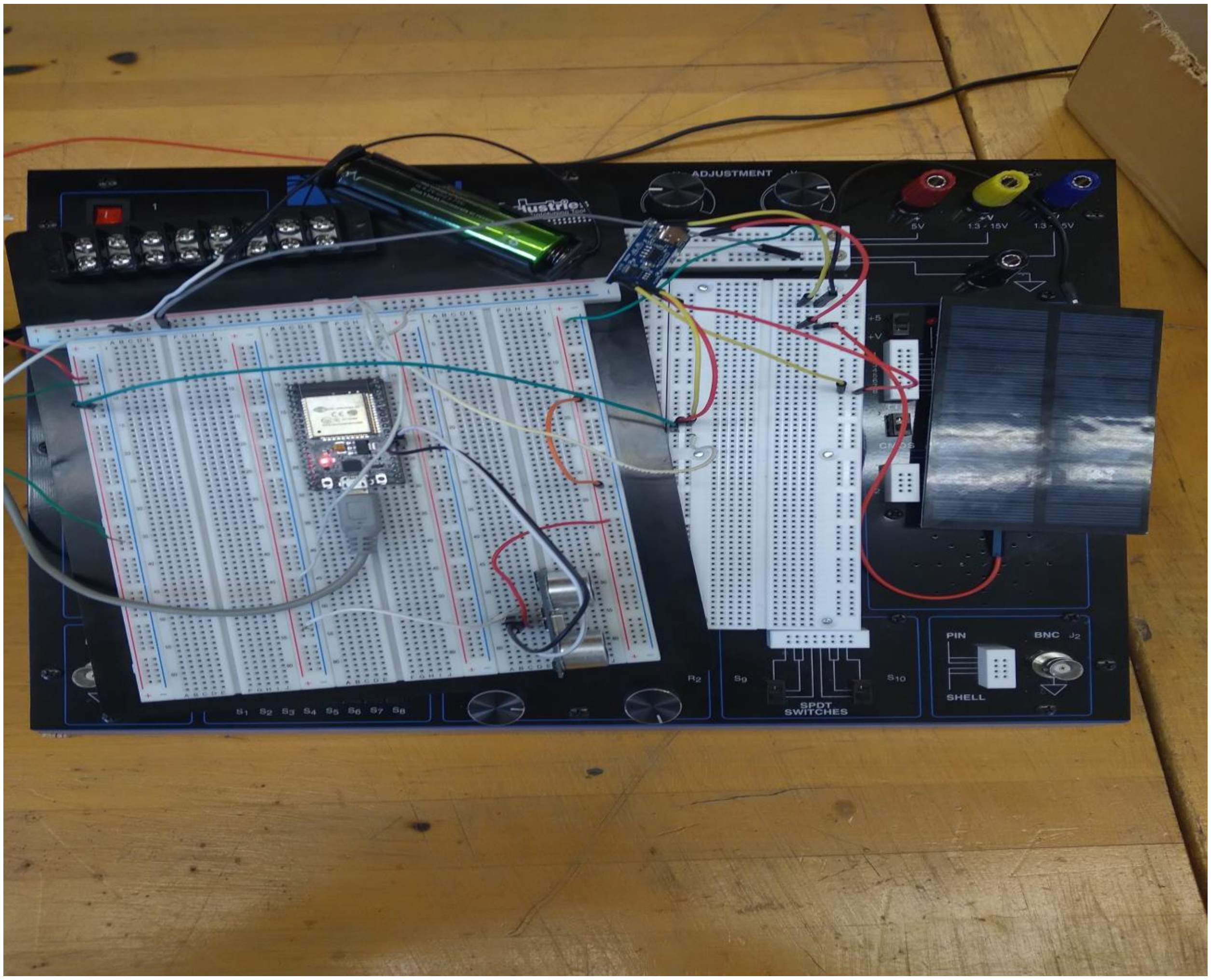


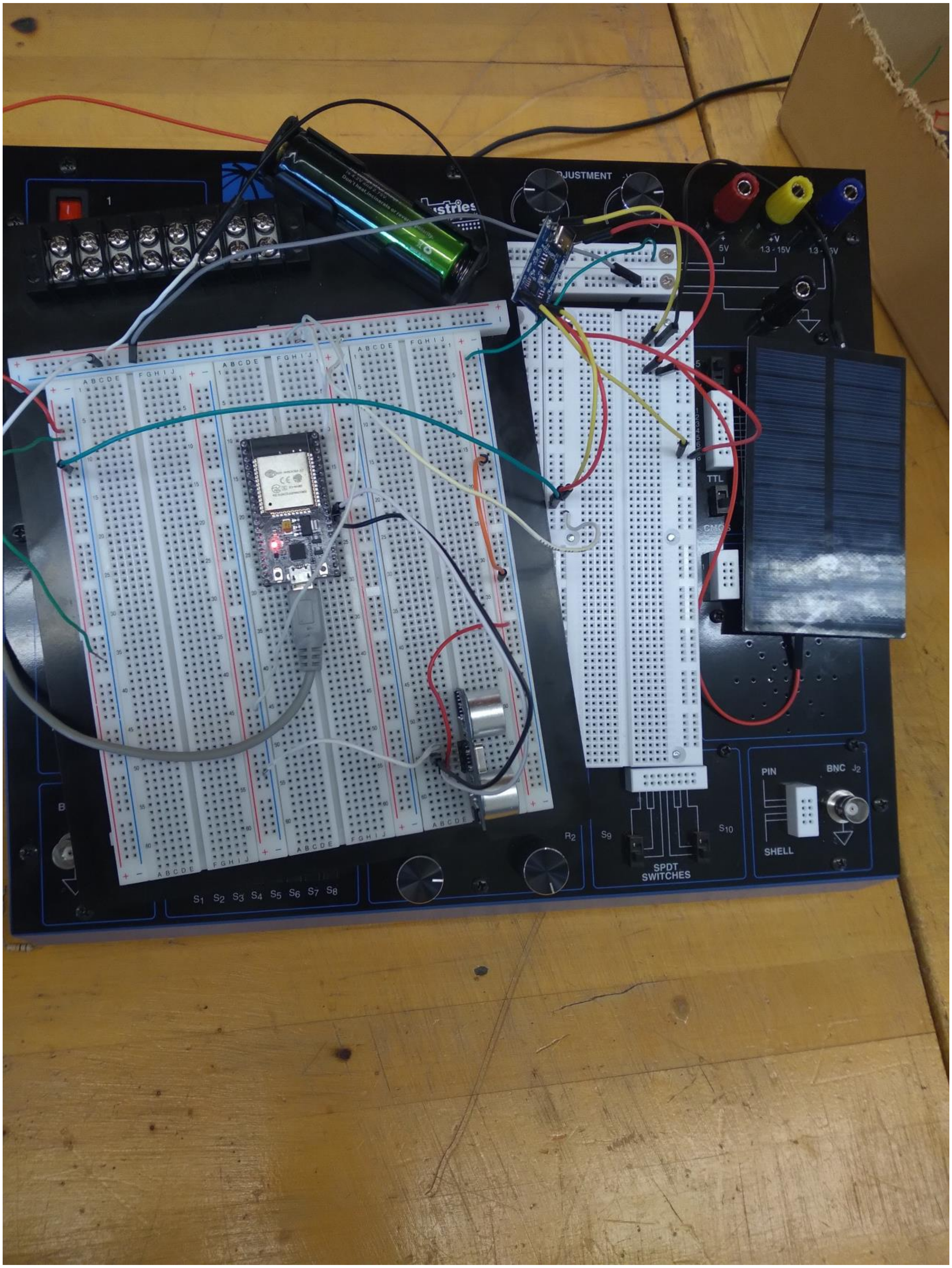
Appendix E : Implementation code

```
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>
#include <BLE2902.h>

bool deviceConnected = false;
const int trigPin = 17;
const int echoPin = 16;

long duration;
long distance;

#define bleServerName "Sensor1"
#define SERVICE_UUID        "6d19a9a4-e8ba-475d-9e06-d468fd97033c"
#define CHARACTERISTIC_UUID "396ac126-23ee-4544-8f07-28d8d0d3ceb5"
#define HEIGHT_UUID "862a9ccb-70fe-4123-9ccd-bed676666945"
BLECharacteristic wasteStatusCharacteristic(CHARACTERISTIC_UUID,BLECharacteristic::PROPERTY_NOTIFY);
BLEDescriptor wasteStatusDescriptor(BLEUUID((uint16_t)0x2902));
BLECharacteristic wasteHeightCharacteristic(HEIGHT_UUID,BLECharacteristic::PROPERTY_NOTIFY);
BLEDescriptor wasteHeightDescriptor(BLEUUID((uint16_t)0x2902));

class MyServerCallbacks: public BLEServerCallbacks {
  void onConnect(BLEServer* pServer) {
    deviceConnected = true;
  };
  void onDisconnect(BLEServer* pServer) {
    deviceConnected = false;
  }
};

void setup() {
  // put your setup code here, to run once:
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(115200);

  BLEDevice::init(bleServerName);
  BLEServer *pServer = BLEDevice::createServer();
  pServer->setCallbacks(new MyServerCallbacks());
  BLEService *sensorService = pServer->createService(SERVICE_UUID);
  sensorService->addCharacteristic(&wasteStatusCharacteristic);
  wasteStatusDescriptor.setValue("Waste Status");
  wasteStatusCharacteristic.addDescriptor(&wasteStatusDescriptor);
  sensorService->addCharacteristic(&wasteHeightCharacteristic);
  wasteHeightDescriptor.setValue("Waste Height");
  wasteHeightCharacteristic.addDescriptor(&wasteHeightDescriptor);
  sensorService->start();
  BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();
  pAdvertising->addServiceUUID(SERVICE_UUID);
  pServer->getAdvertising()->start();
  Serial.println("Waiting a client connection to notify...");
}
```

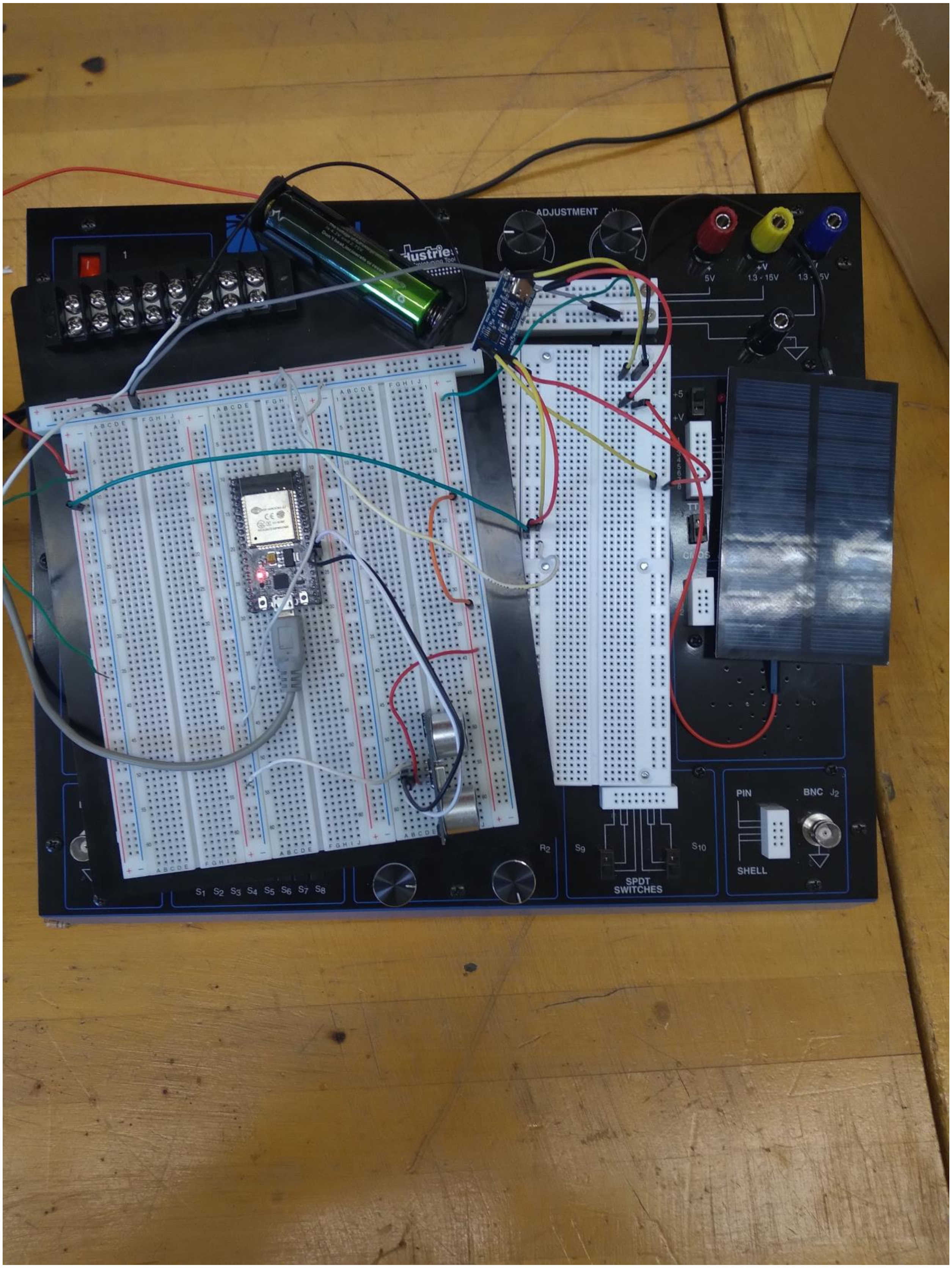
```
void loop() {

  char distanceTemp[7];

  // put your main code here, to run repeatedly:
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
  dtostrf(distance,6,2,distanceTemp);
  if(distance <= 10)
  {
    Serial.print("FULL ");
    wasteStatusCharacteristic.setValue("Full");
    wasteStatusCharacteristic.notify();
    Serial.print("Distance: ");
    wasteHeightCharacteristic.setValue(distanceTemp);
    Serial.println(distance);
  }
  else if (distance > 32)
  {
    Serial.print("EMPTY ");
    wasteStatusCharacteristic.setValue("EMPTY");
    wasteStatusCharacteristic.notify();
    Serial.print("Distance: ");
    wasteHeightCharacteristic.setValue(distanceTemp);
    Serial.println(distance);
  }
  else
  {
    Serial.print("HALF FULL ");
    wasteStatusCharacteristic.setValue("HalF Full");
    wasteStatusCharacteristic.notify();
    Serial.print("Distance: ");
    wasteHeightCharacteristic.setValue(distanceTemp);
    Serial.println(distance);
  }
  delay(1500);
}
```

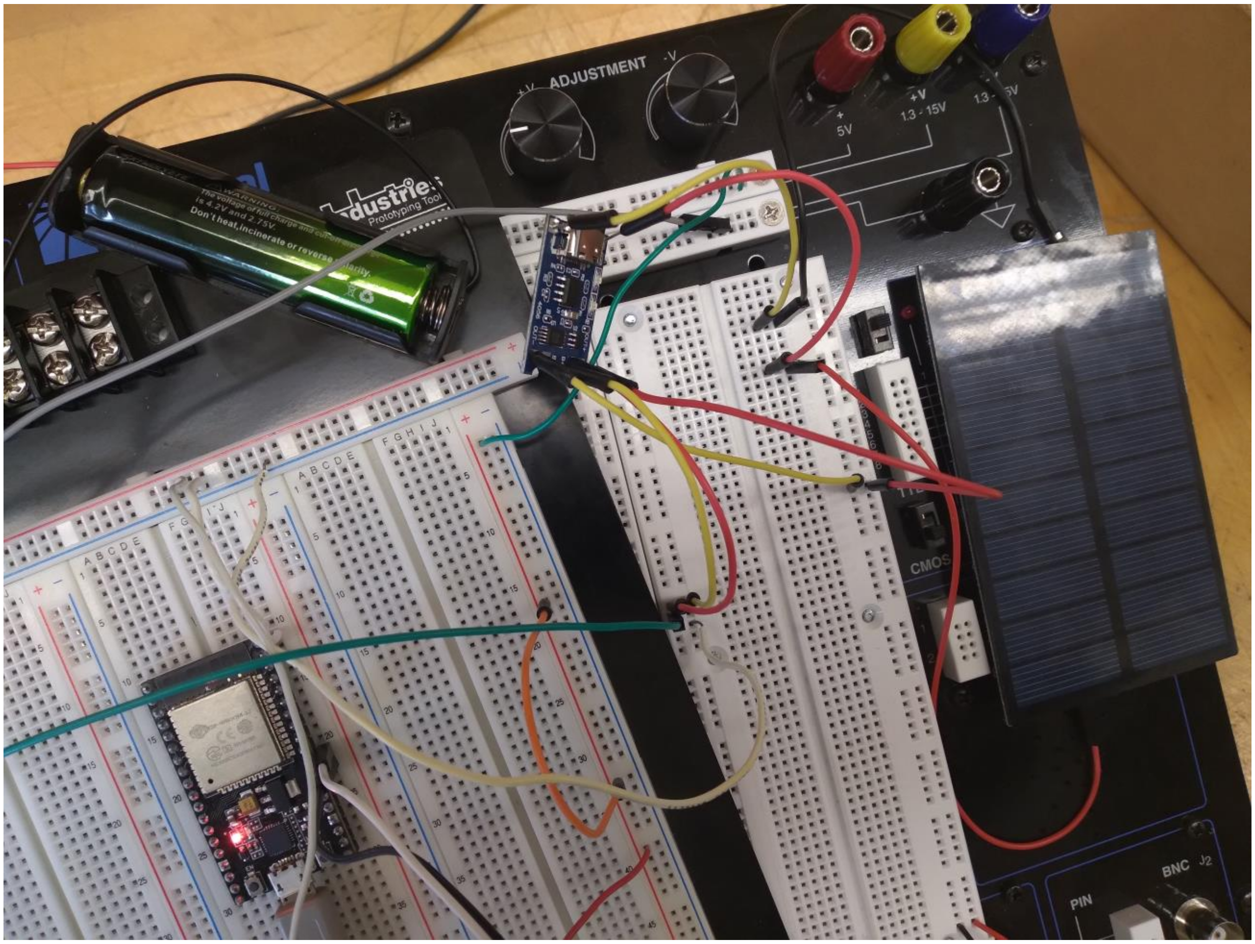
Appendix F: Experimental Setup photos

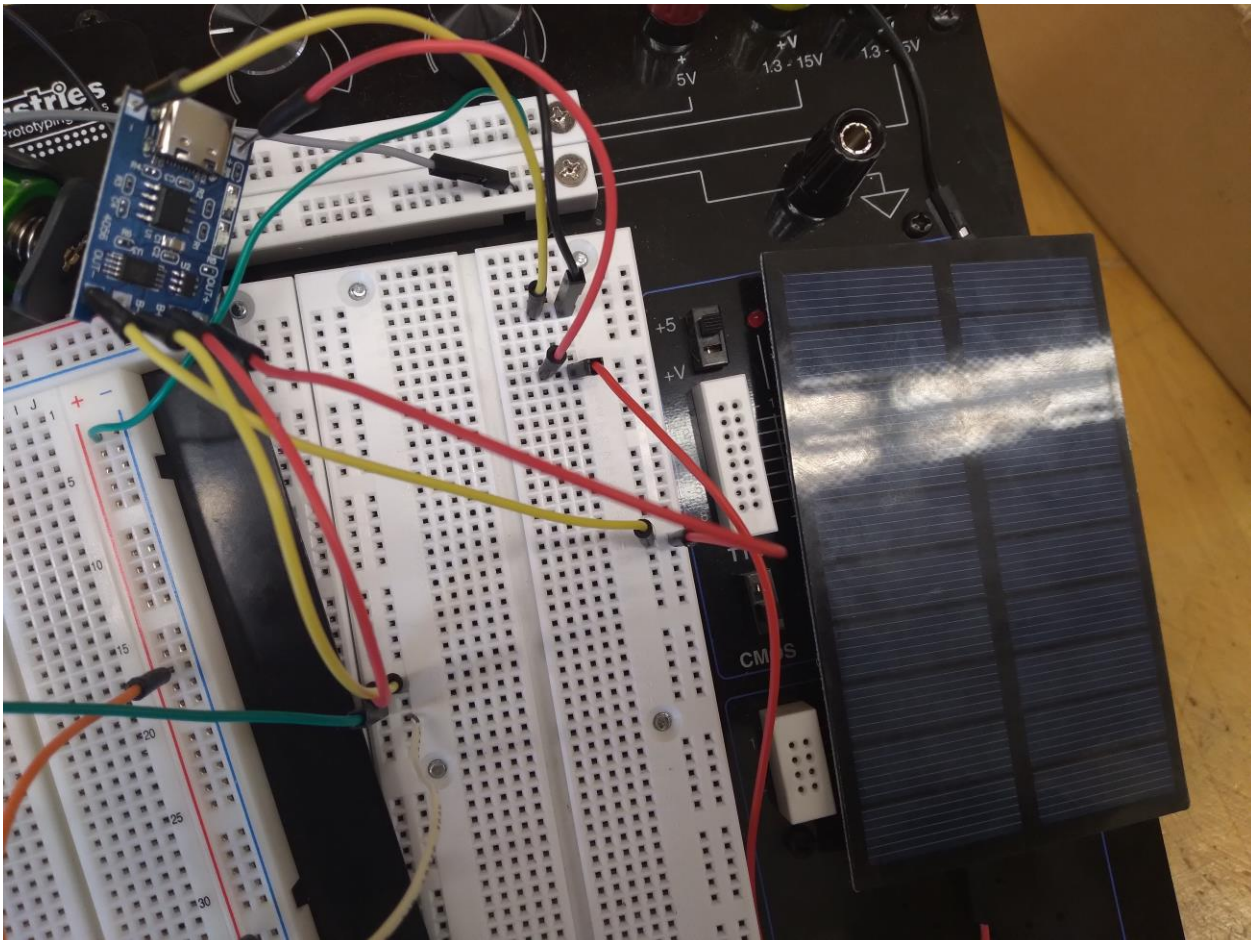




PIN
BNC J2
SHELL
SPDT
SWITCHES
TTL




ADJUSTMENT
+V
-V
5V
1.3 - 15V
Prototyping Tool
CMOS
PIN
BNC
J2


+
5V
+V
1.3 - 15V
+5
+V