ECE 499 - Senior Project Proposal
Panduit Sponsored - Wireless Client Location Determination

Nicholas Pratt
Nathan Ruetten
Michael Stanczyk

November 17, 2016

Table of Contents

Abstract - With the anticipated volume of wireless client endpoints expected for the Internet of Things (IoT), determining where the physical endpoint is, will be a highly desirable capability. Today's location based capabilities are not very granular (on the order of 10m or so), what is more desired is a very robust location based capability to have a granularity of less than 1m in 3-space. Additionally the capability must be easily deployable, easy to use and be low cost. The operating environment should be assumed to be within a building (where the RF from the outside of the building is severely attenuated as it enters the building – hence traditional GPS signals are greatly attenuated). This project should deliver a proposal of a solution (with a comparison of various competitive techniques) followed by a prototype to prove the concept.

I. Introduction

Many organizations currently utilize various types of wireless sensors for different applications. It can be advantageous to know the physical location of these wireless sensors. Sensors outdoors can utilize the Global Positioning System (GPS) for fairly accurate location information, but the GPS signals are greatly attenuated indoors making this solution impractical. Sensor locations can be manually assigned, but as greater numbers of sensors are implemented, it would be much easier to have this location be determined automatically. Due to these constraints, a wireless sensor location determination capability will most likely utilize 802.11ac Wi-Fi.

The initial work on this project focused on literature review and the finding and comparing of various methods of wireless location determination. After comparing several methods from a theoretical standpoint, a single method was picked, and the building of a prototype of that method is the focus of the remaining work.

II. Review of Literature

The initial phone call with the sponsoring contact, Dr. Ron Nordin at Panduit [1], and some background research provided a foundation upon which several methods of location determination were decided.

Dr. Nordin explained five key properties that a robust wireless location determination system would have.

1. The system should support sensors that can simply be placed and turned on with no additional setup required.
2. The system needs to work with any wireless sensor. The infrastructure itself must handle all location determination calculations and measurements.

3. The system must conform to all governmental regulations for wireless protocols.
4. The system should continuously monitor the environment for new, missing, or changed sensors.
5. The system should be accurate to within an ideal 1 meter^2.

Even though this is a project sponsored by Panduit, Dr. Nordin encouraged the idea of using open-source software.

According to Junjie Liu [2], a wireless based indoor location determination system consists of “beacon stations that emit the wireless signal and the user devices that receive the signal or vice versa”. In the case of this Panduit project, the user devices are the wireless sensors and the beacon stations must be created. The logic for computing location would reside with the beacon stations as the wireless sensors must remain unchanged.

While there are several other wireless technologies that could be used for location determination including ZigBee, Frequency Modulation (FM), Bluetooth, and RFID, WiFi is the most fruitful in terms of research. FM has the greatest range, but would not provide enough signal data to determine location accurately enough. Bluetooth and RFID are much shorter-range and would require a significant deployment cost to provide enough beacons. ZigBee has similar if not better specifications than WiFi but requires a dedicated and complex mesh infrastructure. While several of these technologies could be combined, the complexity of such a network would likely exceed the scope of this project.

Liu also provides several criteria for evaluating the performance of a wireless client localization system. These include:

1. Accuracy - The goal for this system is within 1 square meter.
2. Precision - The system aims to give reliable results on a consistent basis.
3. Coverage - How much area is covered by a single beacon?
4. Update Interval - The system aims to provide as close to real-time data as possible.
5. Computational Cost - The system aims to be low-power to improve battery life and decrease cost.
6. Infrastructure - What is the expense for deploying and maintaining the infrastructure itself?
7. Offline Computing - The offline expense for updating and maintaining the infrastructure.
8. Localization Time - The time it takes to determine location.

The above criteria will be used to estimate the performance of each system in theory, and then will be directly measured for the prototype system.

III. Standards and Patents

Any wireless location determination system would make use of the IEEE 802.11ac standard [3]. This standard expands on the 802.11 family of wireless networking standards. The most significant feature added in 802.11ac is the standardization of beamforming which was originally introduced in the now deprecated 802.11n standard.

IV. Subsystem Level Function Requirements

Extensive research has led to the development of four possible solutions to this project. Each of these solutions can be evaluated by the criteria described by Liu above, but without a prototype these are only theoretical evaluations.

The first two solutions represent low-level systems which would require custom developed location determination algorithms. As such, the criteria evaluations are heavily dependent on the efficiency of the developed algorithm. The third solution is an open source software package, and the fourth solution is a proprietary on-the-market product.

Linksys AC1900 Dual Band Open Source WiFi Wireless Router

Accuracy	Dependent on developed algorithm - four antennas per router would most likely give more accurate results
Precision	Dependent on developed algorithm
Coverage	Approximately 30 meters from each beacon
Update Interval	Dependent on developed algorithm
Computational Cost	Dependent on developed algorithm and update interval
Infrastructure	Requires extensive infrastructure overhaul - dependent on size of building/warehouse, but each router cost $250
Offline Computing	Variable
Localization Time	Dependent on developed algorithm

USB WiFi Dongle with Linux Computer

Accuracy	Dependent on developed algorithm - most likely least accurate of all solutions
Precision	Dependent on developed algorithm
Coverage	Approximately 30 meters away from each beacon
Update Interval	Dependent on developed algorithm
Computational Cost	Dependent on developed algorithm and update interval. Most likely most power-hungry of all solutions due to separate Linux box
Infrastructure	Fairly costly. Each beacon consists of a USB WiFi dongle and Linux box (may be able to lower cost using Raspberry Pi, etc.)
Offline Computing	Variable
Localization Time	Dependent on developed algorithm

SubPos (Subterranean Positioning System)

Accuracy	Developer website claims within 1 m^2 accuracy depending on hardware and number of nodes
Precision	Fairly consistent
Coverage	Approximately 30 meters from each beacon (WiFi standard)
Update Interval	Can be adjusted to preference
Computational Cost	Dependent on update interval
Infrastructure	Requires additional infrastructure. Each beacon would cost approximately $3 when purchased in bulk
Offline Computing	Variable - beacon locations may need to be explicitly programmed
Localization Time	< 2 seconds

Cisco Hyperlocation [4]

Accuracy	Given as between 1 and 3 meters with extra antenna module
Precision	Fairly precise
Coverage	Recommended 1 AP for every 2500 ft^2
Update Interval	Variable. Cisco gives refresh rate anywhere from 1 to 10 seconds
Computational Cost	Dependent on update interval but tied to current WiFi usage
Infrastructure	Requires minimal overhaul if currently using a Cisco system, but components are expensive and software also expensive
Offline Computing	Physical locations of each AP must be mapped in the software
Localization Time	< 1 second - associated with current WiFi usage

The SubPos system was determined to be the best solution to focus the scope of this project. The SubPos system is a software package licensed under GNU General Public License which can be easily modified, tested, and installed on cheap hardware. More specific details on the SubPos standard can be found at http://subpos.org/. This solution was chosen for the following reasons:

1. **Inexpensive to test and quick to implement** - the SubPos software is free under GPL and hardware components would be inexpensive to buy and test.
2. **Claimed accuracy** - several tests of the SubPos system have claimed accuracy to within one meter or better. This is the desired accuracy of the final system.
3. **3D location determination** - the system is built right away to handle location determination in 3-space rather than simply 2 dimensions.
4. **Easily modifiable** - since the SubPos software is open source under GPL, the system can be modified as necessary to try and match any remaining project goals.

The current negative of the SubPos system at this time is it is written as a client-side software package. Since the software is open source, this can be modified to be server-side to fit the project needs, although this will be a time-consuming process.

The current plan is to test this SubPos system as it is written - a client-side system. Using a phone as the client and ESP8266 modules as the wireless beacons, this will allow testing of the SubPos system accuracy and efficiency.

The SubPos client-side system uses a node's Service Set Identifier (SSID) to carry information that the client-side application uses to calculate location. A diagram showing the breakdown of the 248 bit packet is shown in Figure 1. When the client detects several of these SSIDs, it can use the embedded latitude and longitude as well as the RSSI to attempt to locate itself.

0 — 161

24bit	24bit	32bit	32bit	24bit	26bit
Human Readable Tag ("SPS")	Unique Device ID*	Latitude (decimal degrees, 7dp)	Longitude (decimal degrees, 7dp)	Application ID	Altitude (cm relative to sea-level)~

162 — 247

1bit	1bit	1bit	11bit	3bit	13bit	56bit
Below Sea Level (Altitude sign bit)	Offset Mapping Enable	3D Location Enable	TX Power (EIRP dBm, 1dp)^	Path Loss Coefficient Selection	Application Reserved	Coding Masks (0x7F and invalid chars)

* Unique to an area, not necessarily the entire world. Generally unique for an Application ID.

~ Should be height of floor relative to sea-level, not where the AP is mounted, unless 3D location enabled.

^ Decimal value is offset by +100.0 to make unsigned (dBm; -100dBm is the min 802.11 receive strength, 23dBm is max).

Values with decimal points are fixed point representations.

Figure 1 - The SSID data packet description for the SubPos system

V. Engineering Efforts Completed to Date

The preliminary results of these client-side tests are not as accurate as desired, yet still incomplete. To test the SubPos system, four points were chosen on the boundary of Bradley University's quad. These points are mapped in Figure 2.

Figure 2 - The four chosen test locations for ESP8266 nodes

The latitude and longitude of these points were recorded, and the data packet SSIDs were calculated. The ESP8266 modules were then assembled and connected to 3.3 volt regulators. A picture of the four completed modules can be seen in Figure 3.

A USB-Serial cable connected the ESP8266 modules to a computer, and the terminal program putty was then used to communicate with the modules. The modules have connection properties of 8N1 bits, 115200 baud, and no flow control. Each communication line must end with a carriage return and line feed character.

AT commands could then be used to communicate with the devices. An example of AT commands using to test a device is shown in Figure 4.

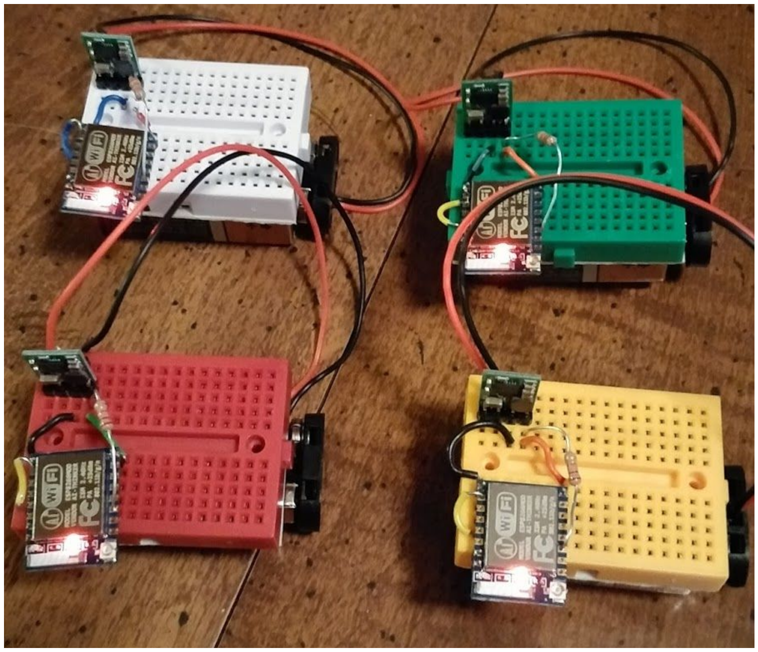

Figure 3 - Four completed ESP8266 beacons

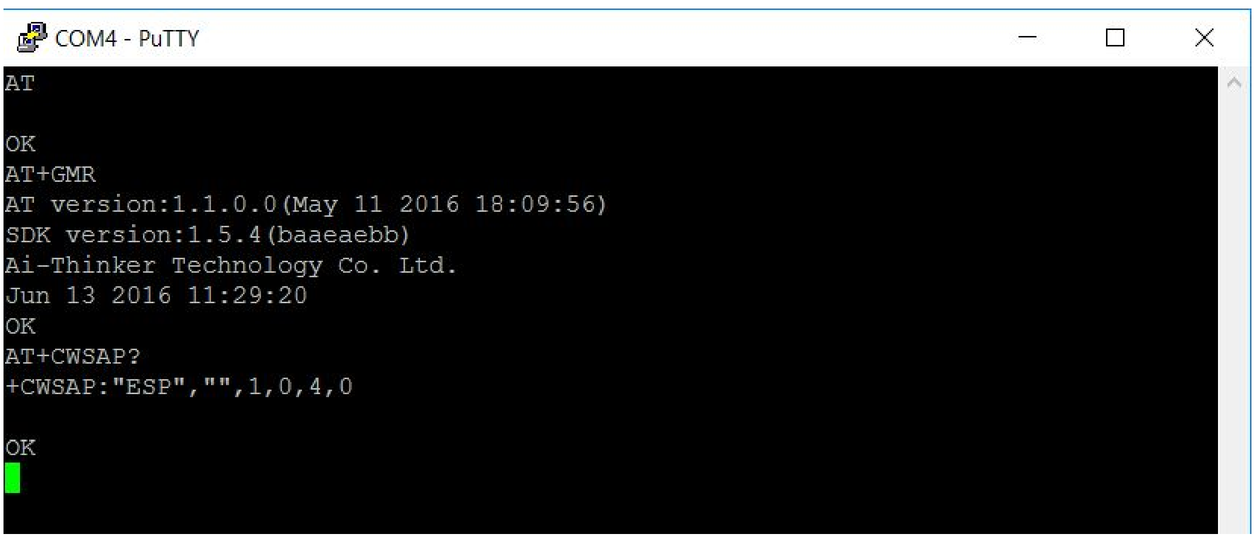

Figure 4 - AT commands being used to test an ESP8266 device

In the example shown in Figure 4, the *AT+GMR* command shows AT and SDK version numbers, and the *AT+CWSAP?* command shows that the ESP8266 is currently setup to broadcast an SSID of “ESP”, no password, on channel 1 of an open network.

By setting the SSIDs of the ESP8266 modules to the hex values calculated for the data packet of each specific node, the client-side SubPos monitor app can then be used to track location. An example of programming the ESP8266 module using the RealTerm program is shown in Figure

5. Note how the terminal screen attempts to display all values as ASCII characters, but some hex values do not correspond to readable letters or numbers. The ones that do will be visible on a phone’s WiFi network selection screen.

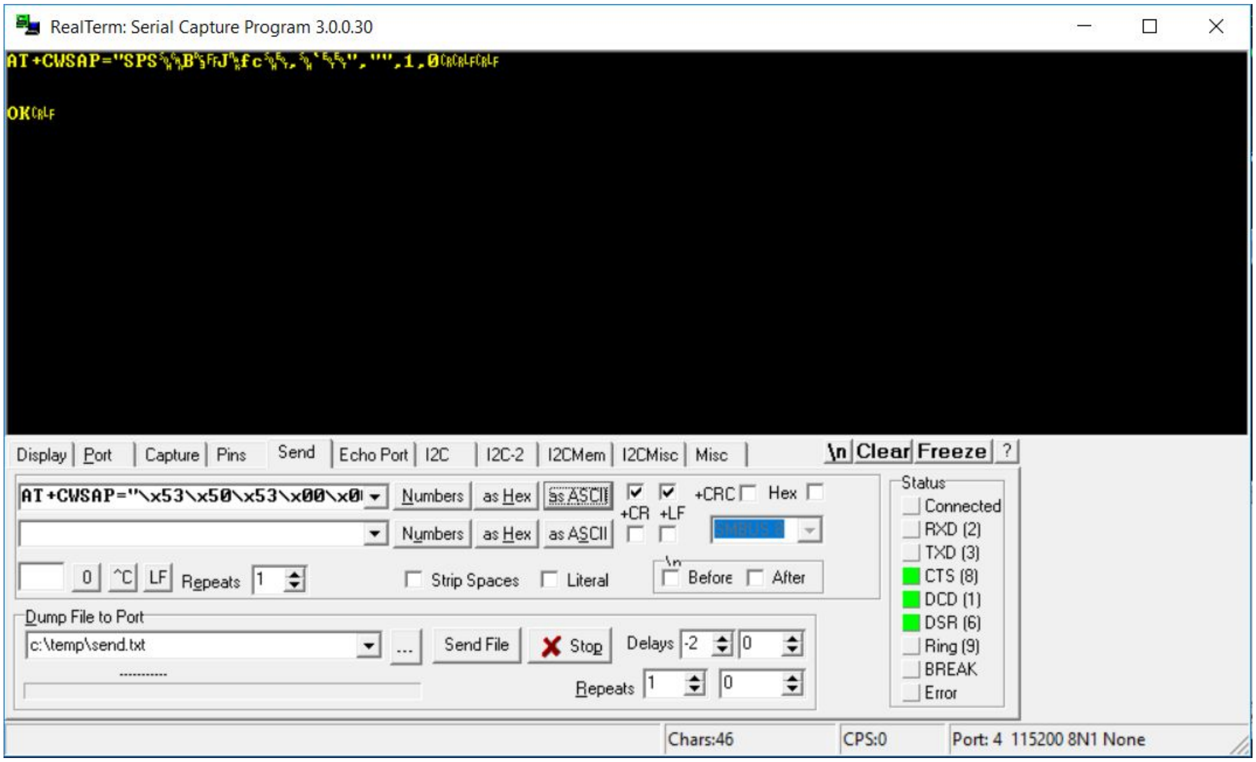

Figure 5 - The RealTerm program being used to set the SSID of an ESP8266 node

Once all nodes are programmed, turning each device on and checking a phone’s WiFi network selection screen shows all four nodes available as seen in Figure 6.

The nodes were then placed on the ground in their recorded locations. Checking the SubPos monitor app showed the four nodes exactly where they were designed to be placed. A screen capture of this result is shown in Figure 7.

The application then attempted to calculate the position of the client using the four nodes. The first attempt at localization is shown in Figure 8 with the blue pin representing the calculated location. As shown, the first attempt at location determination was incorrect by about 1.4 kilometers.

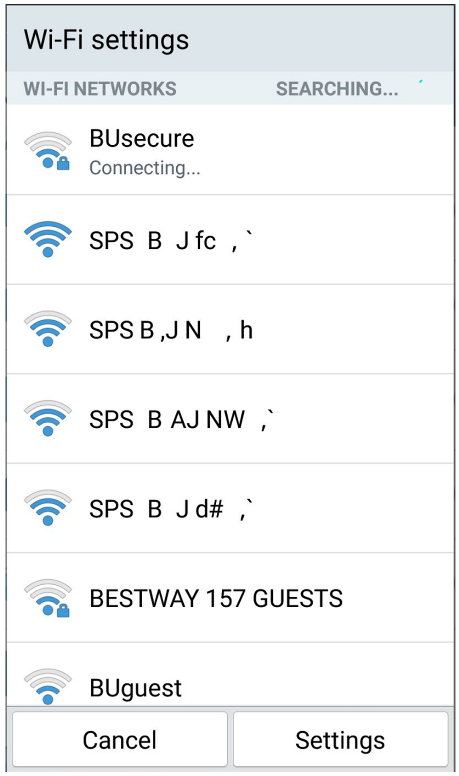

Figure 6 - A list of all available WiFi networks showing the four ESP8266 nodes all starting with "SPS"

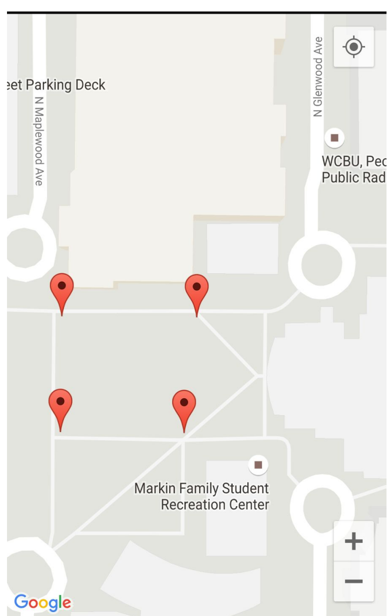

Figure 7 - The four ESP8266 nodes on the corners of Bradley University's quad showing equivalent locations to those planned in Figure 1

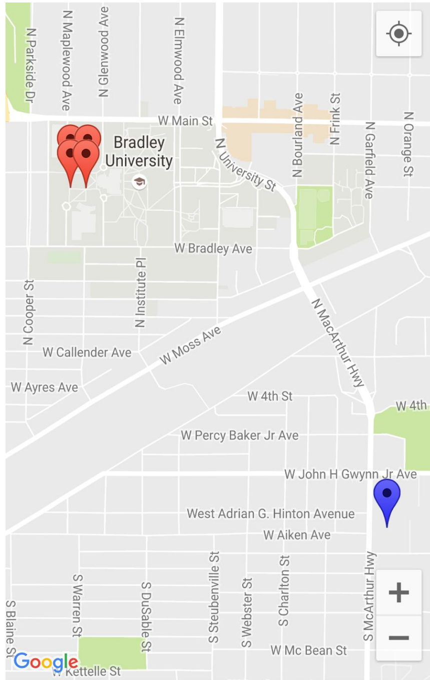

Figure 8 - The first attempt of location determination using four nodes

Seeing as how the client was located approximately equidistant from all nodes in the center of the quad, this is an unacceptable result as the desired accuracy of the system is less than 1 meter. The area being covered on the quad by the four nodes is fairly large, so the four nodes were moved to be about 10 meters apart in a square shape without reprogramming their SSID locations. The hypothesis was that if the nodes help calculate relative location, standing in the middle of this small square would place the location determination in the middle of the quad.

This second test is shown in Figure 9. Again, the location calculated is incorrect, but with an error of only 350 meters. Again, this result is not acceptable, but shows how increasing the density of the nodes increases the location accuracy. As another test, the client was moved close to Node 4, the south-eastern node. Recalculating the client's location gives the result in Figure 10. Note how the client was located in a location to the south-east.

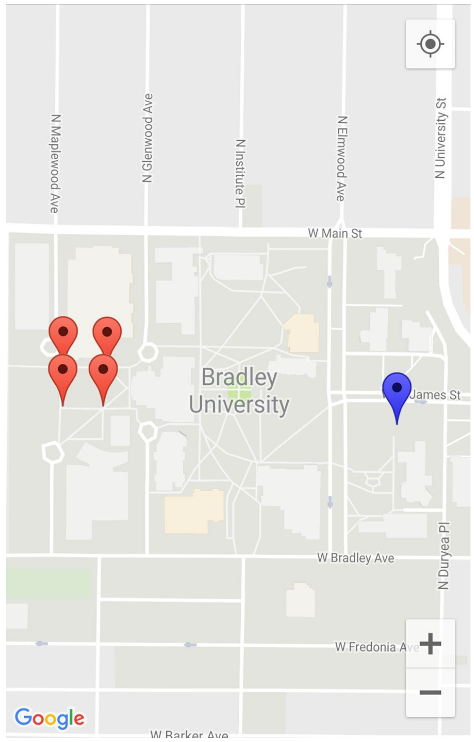

Figure 9 - The larger density node test.

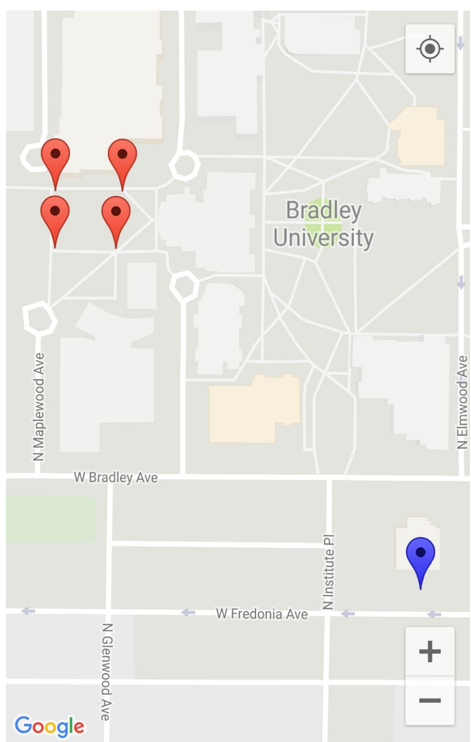

Figure 10 - The location test with the client next to the south-eastern node

VI. Parts List

The parts list for the SubPos system client-side test can be seen in Table 1.

Table 1. SubPos System Client-Side Parts List

Quantity	Description	Price	Ext. Price
6	HiLetgo ESP8266 Serial WIFI Wireless Module ESP-07 Wireless Module	$6.99	$41.94
3	Highfine 2 x 6dBi 2.4GHz 5GHz Dual Band WiFi RP-SMA Antenna	$9.69	$29.07
2	3.3 V Regulator (5)	$5.71	$11.42
2	9 volt battery clip (5)	$5.19	$10.38
1	USB to UART cable	$5.99	$5.99
1	Header pins (40pinsx25pc)	$8.08	$8.08
1	Elegoo 6PCS 170 tie-points Mini Breadboard kit	$6.49	$6.49
1	8 Count - Duracell MN1604 9V Volt 6LR61 Duralock Coppertop Alkaline Batteries	$12.99	$12.99
6	3.3V, 500mA Step-Down Voltage Regulator D24V5F3	$4.25	$25.50
		Total	$151.86

VII. Deliverables for ECE 499, Division of Labor, Schedule for Completion

For the coming semester, there is not a Gantt chart simply because there are many uncertainties with this project that could terminate the plan of the Gantt chart entirely. For example, if the SubPos system doesn’t meet the required specifications, such as the 1 meter accuracy, it would be better if the groundwork for the next senior project is laid out than to continue on the plan. As a result, this group must behave much like an agile team until the specifications can be meet in the client-side tests. The tests will be improved by adding more nodes and correcting the path-loss coefficient. They are planned to be completed by the beginning of February. Although the lack of a solid plan will damper the progress of the project, the cross-functions of the group keep the project very mobile for each step. Assuming the SubPos system continues working, Michael and Nathan will focus on flipping the SubPos system from client side to server side, where the ESP8266 modules are wireless nodes passing information onto a server. Nick will focus on reprogramming the ESP8266 modules for finishing the client-side testing.

VIII. Discussion and Future Directions

If the SubPos system does not work out, the next solution to start working on is the Linksys AC1900 Dual Band Open Source WiFi Wireless Router. If this were to happen there would be no way for the complete solution to be finished before May. The solution requires creating the location algorithm from scratch. This algorithm would dictate how accurate, efficient, and robust the system would be. Thus, the group would be focused on developing the algorithm the best in the remaining time via literature search. This would be in preparation for another senior project group another year to pick up the baton.

References

[1] Note from the initial material received from Dr. Nordin followed by a teleconference with him on 10/03/2016.

[2] Liu, Junjie. "Survey of Wireless Based Indoor Localization Technologies." Washington University in Saint Louis., 30 April 2014. Web. 17 Nov. 2016.

[3] http://standards.ieee.org/findstds/standard/802.11ac-2013.html

[4] "Cisco Hyperlocation Module with Advanced Security Data Sheet." Cisco. Cisco Systems, Inc., 12 Sept. 2016. Web. 17 Nov. 2016.