Luke Howe-Kerr

Mr. Baar

Chemistry

17 January 2014

Determining the Practical Range of Bluetooth Low Energy on

an iPhone and a Raspberry Pi

Today’s world is booming with new technology from wearables to drones to technology-infused cars. This multitude of technological advances is becoming so ingrained into our daily lives that our world requires a way to seamlessly and wirelessly interconnect them. Bluetooth Low Energy may just be that system, but its practical range remains uncertain in many situations.

**Background**

In 2011, Bluetooth developed their newest version, 4.0 with Low Energy capabilities. This newest version of Bluetooth uses 50% - 99% less power than previous versions of Bluetooth, while retaining most of the same features. Apple was the first to integrate this version into its devices in 2012, and it expanded to all major mobile operating systems by early 2013. The Low Energy feature, nicknamed Bluetooth Smart, uses so little energy that simple devices, such as a heart monitor, can run on a single coin battery for years, enabling a world full of interconnected devices, everything from your toothbrush to your shoes to your car. Bluetooth Low Energy is the dawn of the “Internet of Things,” a theory that people and the objects in their environment will all become connected wirelessly (Bluetooth 1).

Another innovation changing our world on a smaller scale is the Raspberry Pi, a credit card sized computer capable of running a full version of Linux. The Raspberry Pi Foundation was established in the UK to “break the paradigm where without spending hundreds of pounds on a PC, families can’t use the Internet.” Originally created for educational purposes with children, the Raspberry Pi has expanded to uses in hospitals, museums, third world countries, and hobbyists of all ages (About 1).

**Real World Applications**

The use of Bluetooth Low Energy has already exploded throughout the world of technology. In late summer of 2013, Apple introduced iBeacon, a technology that utilized Bluetooth Low Energy to enhance shoppers’ experiences. It wasn’t until a few months after its release, in mid November, that iBeacon really fell into the public spotlight when Apple installed them throughout their stores. If a customer downloads the appropriate app on their phone, they can easily be notified when a purchase is ready to pick up or can wirelessly be given deals and promotions. Within the next two months, iBeacon technology was installed in Safeway grocery stores across the northwest, and a few coffee shops have even adopted it.

Many of our daily lives are already being influenced by smart home technology, and from an unexpected source: a thermostat. Nest has created a smart thermostat that connects to our world through wifi to maximize energy efficiency within your home with one flaw. When you have left your home for a period of time, the Nest shuts off the heating or cooling system. However, it relies solely on a single motion detector to determine if someone is, in fact, home. In an attempt to fix this problem, my mentor, Jeff Squyres, wrote a program on a Raspberry Pi that scanned the local wifi for Internet traffic from specific phones. This program determined that if it noticed Internet traffic, then somebody was home, and communicated that to the Nest. However, as phones become more and more efficient, less and less Internet traffic occurs in the background, making the program impractical. Based on the results achieved within this research project, our ultimate goal is to modify the program to also utilize Bluetooth Low Energy’s proximity sensor to scan for phones as well, creating the most accurate determination of whether or not a person is home or not.

**Experiments**

Throughout this research project, a number of experiments were performed to determine the optimal application and range of Bluetooth Low Energy while running on the iPhone and the Raspberry Pi. The first four tests I performed were to determine the ideal orientation of the Raspberry Pi as well as provide a baseline range for the Bluetooth Low Energy. In each test, both the Raspberry Pi and the iPhone were placed on the ground, and the Bluetooth Low Energy signal was tested every 5 feet until the signal could not be detected by running a scanner program on the Raspberry Pi. The scanner program recorded data such as the name of the device, its Bluetooth address, the number of times it saw it, and a time stamp. The only difference between the four experiments was the direction of the Raspberry Pi. First, I placed it with the Bluetooth nub facing the iPhone. For each consecutive test, the Raspberry Pi was rotated 90o. Next, using whichever orientation of the Raspberry Pi received the best signal, I tested the range again while placing the Raspberry Pi at a height of 4 feet off of the ground.

At this point, all of the tests remained stationary, but in the real world, it must be just as affective while moving as while stationary. To test this, I used the same setup as in the first four tests and simply moved in a straight line parallel to the Raspberry Pi where the closest point is the measurement of the 5-foot increments. I repeated this experiment 2 more times in order to tests speeds of running, walking, and slow.

In order to transfer this knowledge indoors, I had to perform additional experiments. The first experiment took a very similar setup as the first four experiments, except I placed the Raspberry Pi directly behind a wall. Then I repeated this experiment except I switched locations of the devices, so the iPhone remained behind a wall while I moved the Raspberry Pi. For a third test, I moved both the iPhone and the Raspberry Pi so that the total distance between them remained at 5-foot increments. The last experiment I performed was an indoor movement test. Because of the difficulty of obtaining a variable-free environment was impossible, I simply attempted to maintain steady, measurable distances from the Raspberry Pi as best I could.

**Experimental Setup**

While performing all experiments, I used an iPhone 5 running iOS 7.0.4, and version 1.50 of the LightBlue app. I also used a Raspberry Pi model B running Raspbian with a Plugable USB Bluetooth 4.0 Low Energy micro adapter plugged into it and the BlueZ toolkit 4.99.

I performed all outdoor experiments in my driveway, with the Raspberry Pi located at the edge of the garage, right in front of the garage door, so as to achieve the least-interrupted workspace possible.

**Results/Analysis**

To aid in organization, I gave each experiment a letter to identify it. The letter corresponds to the order of the experiments. Experiments A, B, C, and D (see below) were very interesting. All four experiments tapered off by 30ft.

Experiment A saw the iPhone 54 times per second to begin with, and at 10ft, it dropped off quite quickly, and was only seen about 9 times. At 20ft or 25ft, it did detect the iPhone, but very inconsistently.

Experiment B began strong, but immediately dropped significantly at 5ft. By 15ft, experiment B could not detect the iPhone at all.

Experiment C dropped of quite quickly, but then remained consistent for the farthest of all four of these experiments.

Experiment D performed very similarly to experiment B and was useless beyond 15ft. (*Experiment D2 is not pictured because of technical difficulties when trying to access the data*)

When you compare all four experiments, they all seem to follow the same general pattern at first. However, experiment A distinguishes itself as the strongest signal of the group. Strength isn’t necessarily what we are looking for as much as consistency. Experiment C distinguishes itself as the most consistent of the experiments primarily because of its performance at both 15ft and 20ft, making it the ideal orientation of the Raspberry Pi.

(*The remainder of this section is currently unavailable because of technical difficulties when accessing the data on the Raspberry Pi. The data has been recorded on the Raspberry Pi, but I have yet to be able to access it. The problem will be resolved by this weekend (18th-19th), and the remaining results will be added in the next draft.*)

**Conclusions**

Our goal in this experiment was to determine the ideal setup for consistently detecting an iPhone using Bluetooth Low Energy and determine its range. Based off of the results of the first four experiments, the most consistent orientation of the Raspberry Pi is 180o away from the iPhone, or that of experiment C. Even though experiment A picked up a stronger signal, it lacked the consistency over the long distance that experiment C displayed. (*Because I only have access to a portion of the data, I will refrain from drawing any further conclusions at this time*)

In the future, I would also like to test these same experiments except have the Raspberry Pi broadcast the Bluetooth Low Energy signal. Because the Raspberry Pi is not reliant on a battery, it could potentially broadcast a stronger signal than the iPhone was able to. I would also like to take the results of these experiments and apply them to the current Ruby Slippers program in order to test the quality of Bluetooth Low Energy in the real world.

Works Cited

"About Us." *Raspberry Pi*. Raspberry Pi Foundation, n.d. Web. 14 Jan. 2014. <http://www.raspberrypi.org/about>.

"Bluetooth Smart." *Bluetooth*. Bluetooth SIG, 2013. Web. 14 Jan. 2014. <http://www.bluetooth.com/Pages/Bluetooth-Smart.aspx>.

Hern, Alex. "What is Apple's iBeacon?" *The Guardian*. Guardian News and Media, 13 Jan. 2014. Web. 14 Jan. 2014. <http://www.theguardian.com/technology/2014/jan/13/what-is-apple-ibeacon-retail-tracking>.

Appendix

Source files for the scanner program that was run on the Raspberry Pi to scan for the Bluetooth Low Energy signal of the iPhone.