Wireless Sensing Beacon System

Shannon Lin Hurd, Chi-En Dai, Frédéric Camail

1 Implementation

We decided to use Python for this homework, as it is the language that all group members were comfortable using. Along with that, it's simple and concise syntax made collaboration on the code easier.

We utilized the bluepy library for Bluetooth communication and RPi.GPIO for LED data transmission. The beacon transmit format we used followed the Eddystone URL specification, and we restricted the message length to be a maximum of 17 bits.

When broadcasting, we use the command of HCItools to let the bluetooth device on the Raspberry PI to broadcast the beacon. For detail, we have to encode the url string to something like ASCII code in hexadecimal and follow the format of eddystone-url. And on the beacon receiver, we use the bluepy library to receive the eddystone-url and the corresponded MAC address.

During LED light data transmission, we sent all information through binary code. We began by sending 8-bit binary string representing the length of the URL, then we converted each character in the URL to its ASCII binary representation and sent it one bit at a time.

The hardware we used included two Raspberry PIs, a grove sensor, and LED lights.

When testing, we always made sure to test in a dark environment. This ensured that no outside light would interfere with our intentional LED signals between the two Raspberry PIs. This would in turn lead to more accurate and consistent results.

2 Improved Design

Rather than sending a Eddystone URL, our improved design aimed to transmit and process a Eddystone UID beacon between the two Raspberry PIs.

2.1 Motivation

Eddystone URL's main purpose is to send users to a specific web page or online resource. It allows for nearby devices to easily receive and access URL content. While this can be very useful in some cases, we were curious about other types of information that can be sent through beacons and how the procedure would differ. We ultimately decided on trying to send and receive an Eddystone UID (Unique Identifier) beacon between the two Raspberry PIs. The advantages of using an Eddystone UID beacon lied in the type of information it sent. It consists of a 10 byte long namespace ID along with an 6 byte long instance ID. Often

times, the namespace ID is utilized to represent a company or organization and the Instance ID is utilized to identify the unique beacon used. Once received, the device will often use this information to trigger certain events, such as open an app or retrieve data from a server. Along with that, it provides more privacy compared to the Eddystone URL, as it doesn't directly reveal specific content. Due to the different data type and use cases compared to Eddystone URL, we were interested in attempting to utilize an Eddystone UID.

2.2 Approaches

We approached this improved design similarly to our homework. We wrote 3 different programs: broadcast program, scan program, and sender program. The broadcast program would broadcast an Eddystone UID using Bluetooth on a Raspberry PI. The scan program would allow a Raspberry PI to scan for any nearby Eddystone UID and extracts the namespace and instance. The sender program allows a Raspberry PI to send the Eddystone UID information through led light transmission. Utilizing two Rapberry PIs

2.3 Implementation Detail

We continued to use Python, for the same reasons stated in part 1.

We also utilized the bluepy library for Bluetooth communication and RPi.GPIO for LED data transmission. The beacon transmit format we used followed the Eddystone UID specification, and we expect a message length of 16 bits according to protocol.

During LED light data transmission, we sent all information through binary code. We began by sending 8-bit binary string representing the length of the URL, then we converted each character in the URL to its ASCII binary representation and sent it one bit at a time.

The hardware we used included two Raspberry PIs, a grove sensor, and LED lights.

When testing, we always made sure to test in a dark environment. This ensured that no outside light would interfere with our intentional LED signals between the two Raspberry PIs. This would in turn lead to more accurate and consistent results.

3 Conclusion & Reference

3.1 Conclusion

As many of our members have never used Raspberry PI, we learned more about how the Raspberry PI hardware functions and the capabilities of it. Through this project, we learned a lot about how Eddystone URL and Eddystone UID can be sent and received. Through learning about these specific beacons, we learned more about the overall beacon protocol. Along with that, we learned how to send, receive, and extract information through LED light transmission.

One difficulty we encountered was finding the MAC address corresponding to the Eddystone URL beacon. We struggled with this because there was a lot of interference from other Bluetooth devices such as phones, computers, and other things that we didn't want to

shut down. In order to solve this problem, we wrote a program that would only accept the device's MAC address that scans to a Eddystone URL.

This project was interesting, however we became curious learning more about how this is used in real and everyday life. We would suggest possibly sharing in-depth information about how these beacons are utilized in real life. This could be through videos, articles, or even an example project of using beacons on a larger and more realistic scale. We would also suggest a possible explanation of strategies for allowing the Eddystone URL length to be longer than 17 bits. As we felt that 17 bits was a bit short, we were curious if we could made the message length longer. Along with that, we would want to know what advantages or disadvantages would occur when changing the length.

3.2 Reference

`https://github.com/google/eddystone/blob/master/eddystone-url/implementations/PyBeacon/PyBeacon/PyBeacon.py?fbclid=IwY2xjawF1raFleHRuA2FlbQIxMAABHXzn8z77ePiwxHkFzt_PUre4SwnOMqBhG5fIL-ZzEMdvgDZYB6Iqwr4i4g_aem_DfVQLxS__Cg8maM7Zc6dgA`

4 Work Distribution

Member Name	Student ID	Work
Chi-En Dai	R13942128	Code of beacon advertising, receiving and decoding, Code debugging, Code of improvement design, Experiments conduct, Demo, Video
Shannon Lin Hurd	T13902202	Improved Design, Report, Demo, Bonus Video
Frédéric Camail	A13922201	Code and test the functionality for sending the message via LED