Anexo de software - TFM Beacons Tracking

Rubén Arce

12 de diciembre de 2020

Índice

Índice de figuras

1. Introducción

Este proyecto engloba programación de 3 sistemas muy diferentes para que transmitan información entre ellos sin problemas de comunicación. Para lo cual se ha empleado el lenguaje de programación C para el firmware de los microcontroladores y Python para poder llevar a cabo los cálculos de distancia y visualización de los beacons.

Se han llevado a cabo pruebas en interiores y exteriores, en una casa, en un supermercado y en un lugar con un número elevado de dispositivos BLE y a continuación se muetra el procedimiento y resultados.

2. Firmware

2.1. Emisor BLE o beacon

2.2. Receptor gateway o master

3. Herramientas de visualización

3.1. Tecnologías

Para generar las pantallas y animaciones he empleado Python con su librería "Pygame", en lo que respecta a los cálculos y gráficas se ha empleado Matplotlib y para hacer operaciones de una forma rápida "NumPy".

3.2. Procedimiento

Se han llevado a cabo TODO simulaciones, en todas ellas la metodología ha sido la siguiente:

1. Escucha de los datos
2. Parseo de los mismos así como almacenamiento en estructuras de datos que permitan luego fácil utilización.
3. Filtraje, cálculos de posicionamiento y ajustes.
4. Visualización y escalado de los resultados.

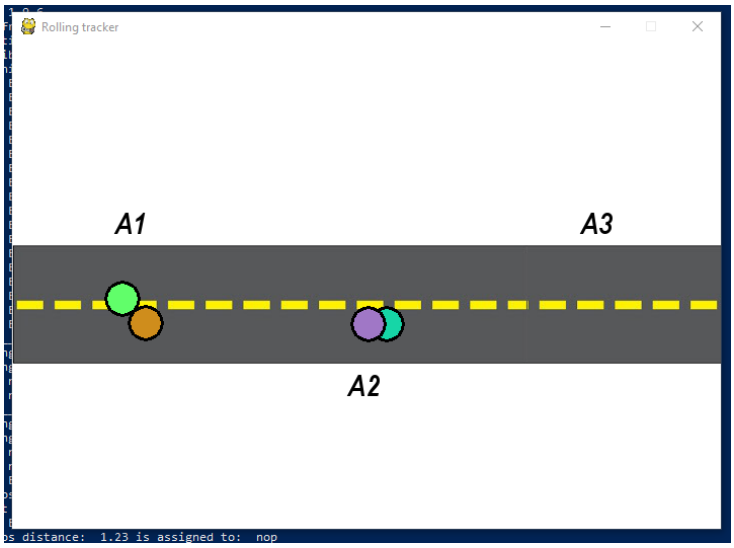

Figura 1: Prueba piloto, carretera

3.3. Simulaciones

3.3.1. Prueba de concepto

Para comprobar si la tecnología era capaz de llevar a cabo el posicionamiento de un equipo en un mapa de estableció un sencillo escenario en el que había tres equipos que escuchaban y 4 beacons. En la siguiente imagen se puede apreciar el sencillo escenario.

En la imagen podemos ver los coloridos beacons próximos entre si y como si que es posible situarlos con relativa preción. En la imagen vemos que los masters tienen los normbres A1, A2 y A3 respectivamente.

3.3.2. Estabilidad del sistema

Tras esta primera prueba se optó por ver hasta que punto la tecnología es precisa, para lo cual se trazaron gráficas comparativas con los datos obtenidos a través de los equipos. En primer lugar se dejó durante un par de horas encendidos los equipos para ver las fluctuaciones.

Como podemos ver en la imagen las diferencias entre equipos estáticos son acusadas, es por ello por lo que queda demostrada la necesidad de desarrollar un filtro que anule este ruido radioeléctrico.

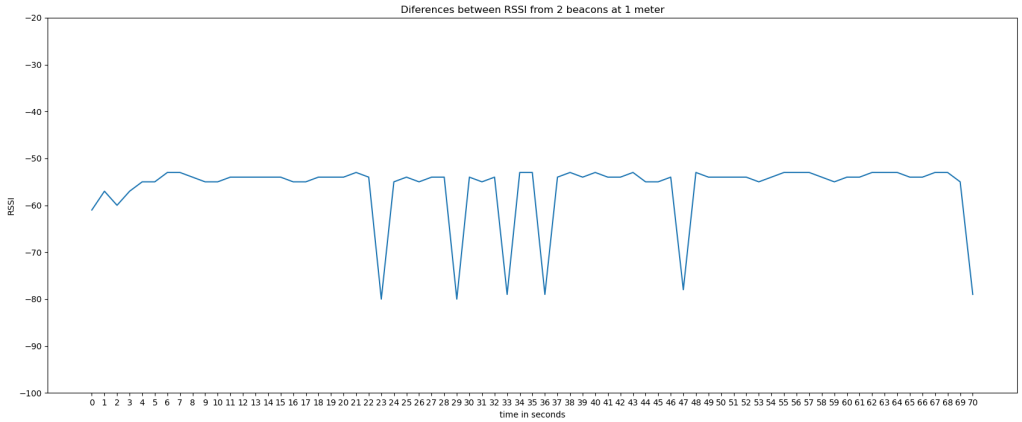

Figura 2: Toma de datos durante 2 horas

3.3.3. Posicionamiento real

El paso siguiente fue comprobar si tras filtrar las señales era realmente capaz de discernir si un Beacon se encontraba en una habitación o en otra tan solo atendiendo a su rssi, para lo cual se ha llevado a cabo una simulación que permite dibujar las dimensiones de las habitaciones, la posición de los beacons e incluso el color de las mismas.

Una vez creada nuestra casa ya solo queda colocar realmente los equipos en lugares elevados y poco acesibles y ejecutar el script. Los equipos estuvieron durante una semana encendidos escuchando beacons y los resultados fueros los esperados.

El caso más curioso de todos es que se dejaron beacons estáticos encima de armarios y la posición no móvil de los mismos se mantuvo en la simulación en todo momento.

Al movernos dentro de la casa los masters de las habitaciones contiguas nos percibían con menor rssi, es por ello por lo que no nos visualizaba dentro de esa estancia.

```
Code > Visualize_Tools > input.txt
1   #Colors of the room -
2   (158, 242, 119)
3   (158, 242, 119)
4   (201, 149, 93)
5   (114, 177, 181)
6   (237, 237, 237)
7   (154, 245, 100)
8   (89, 217, 144)
9   (255, 220, 107)
10
11  #Esp listener coordenates - x, y and name
12  (50 100 A1)
13  (50 200 A2)
14  (50 400 A3)
15  (150 600 A4)
16  (675 50 A5)
17  (650 250 A6)
18  (650 450 A7)
19
20  #Room dimensions - x, y , length, width
21  (50 100 400 50 Corridor)
22  (50 200 350 150 Toilet)
23  (50 400 350 150 Kitchen)
24  (150 600 250 250 Room1)
25  (450 50 250 150 Livingroom)
26  (450 250 250 150 Room2)
27  (450 450 250 300 Room3)
28
```

Figura 3: Archivo de configuración de diseño de casa

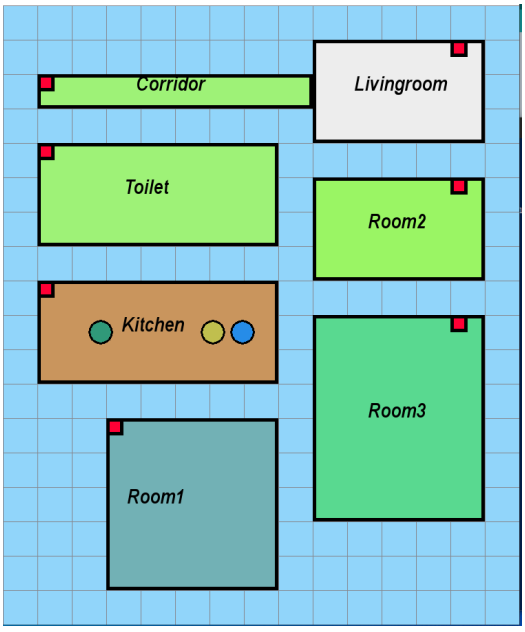

Figura 4: Ejemplo de 3 personas dentro de una misma habitación

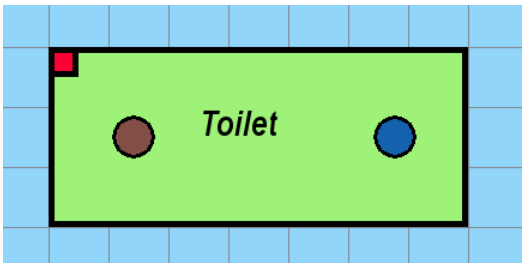

Figura 5: Beacons estáticos durante la simulación

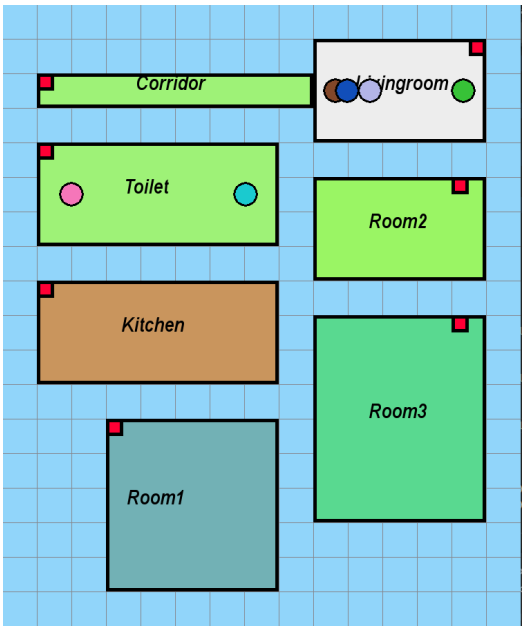

Figura 6: Beacons en movimiento

3.3.4. Prueba en supermercado

Una vez testada la tecnología se llevó a cabo una prueba real en un supermercado, para llevar a cabo la prueba se instalaron únicamente 2 masters y 6 beacons en carros en movimiento. Como suele pasar la prueba real no se parece tanto a las pruebas anteriores y podemos comprobar como el algoritmo se complica más en asuncia de numerosos masters escuchando beacons, en este caso como solo se disponían de 2 los cálculos no siempre eran tan precisos y el filtro implementado en el caso de un desplazamiento rápido de un equipo de punta a punta del supermercado no era el más óptimo.

Podemos ver como los puntos amarillos son los carros de la compra percibidos por ambos beacons y los rojos son aquellos que solo son escuchados por un master, de tal modo que le punto en el que se encuentran es menos precio.

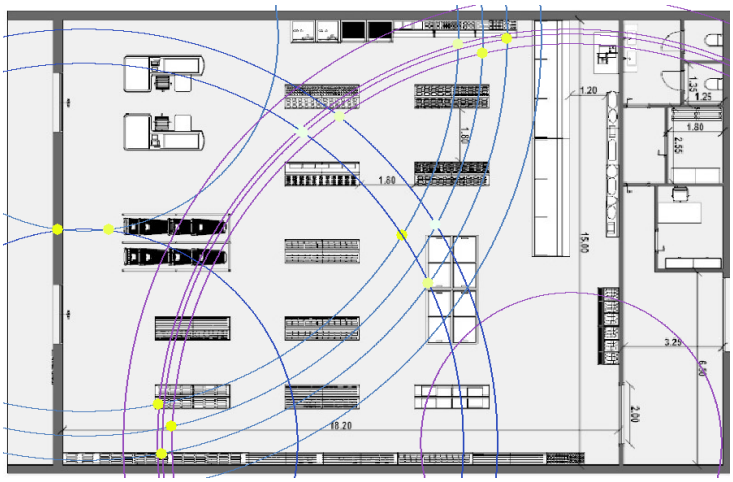

Figura 7: Beacons en movimiento dentro de supermercado captura 1

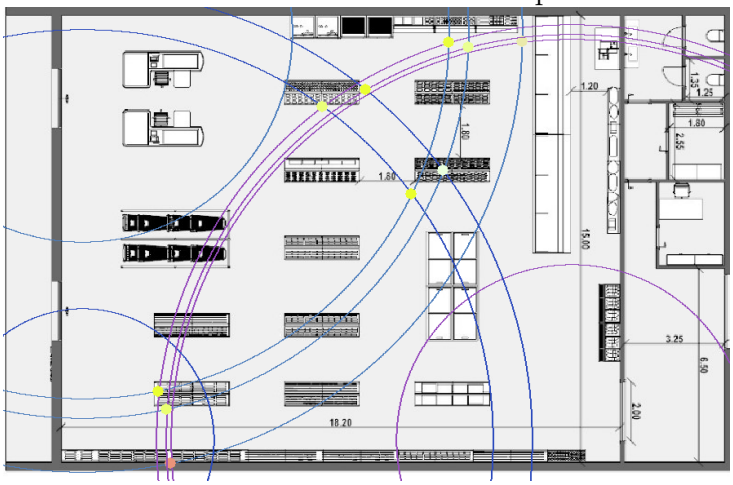

Figura 8: Beacons en movimiento dentro de supermercado captura 2

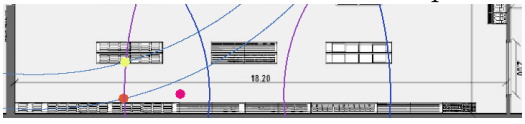

Figura 9: Beacons en supermercado percibidos solo por un master