

Fundamentos de la Programación

Proyecto 2 – Fase 2: *RadioBots*

v1.1

Ingeniería de Robótica SW
ETSIT, URJC

Diciembre de 2019

Resumen

En un aula se han colocado varios *micro:bit* baliza ocultos (en adelante balizas ocultas) cada uno de los cuáles está transmitiendo periódicamente **mensajes con un identificador de baliza**. El número total de balizas es desconocido, pudiéndose deducir en función de cuántos identificadores distintos se reciban. Los mensajes que envían las balizas son transmitidos con la máxima potencia por lo que pueden ser recibidos en cualquier lugar del aula por tu robot.

Tu trabajo consistirá en **programar el robot para que encuentre las balizas ocultas**. El robot aprovechará la información de la potencia con la que recibe los mensajes de las balizas ocultas para orientarse: tratará de moverse en la dirección que haga que se incremente la potencia con la que se van recibiendo los mensajes de la baliza que pretende localizar.

Cuando el robot esté suficientemente cerca de una baliza podrá intercambiar con ella una serie de mensajes, con el objetivo de que la baliza le envíe un **mensaje secreto**. Este mensaje le servirá al robot para demostrar ante el árbitro que ha localizado la baliza cuando llegue a la meta.

Tras obtener los secretos de todas las balizas el robot deberá **dirigirse hacia la meta**, que es un lugar específico del aula **señalizado mediante una luz potente** en el techo. Debajo de la luz habrá un rectángulo negro en el suelo que podrá ser usado por el robot para saber con certeza cuándo ha llegado a la meta. Cuando el robot haya llegado a la meta enviará un mensaje a un *micro:bit* árbitro. Dicho mensaje contendrá un identificador del robot, y todos los secretos que haya obtenido de las balizas que haya localizado. El árbitro contestará con un mensaje de confirmación para que el robot sepa que el mensaje que le envió al árbitro ha sido recibido.

Para detectar las paredes u otros **obstáculos** el robot utilizará el sensor láser, esquivándolos mientras se mueve en busca de los *micro:bit* baliza ocultos, o cuando una vez localizados todos tiene que llegar a la meta.

En la **Fase 3** de este proyecto, que será opcional, los robots deberán **colaborar en equipos de 3 robots** para tratar de localizar cooperativamente las balizas, repartíendose la tarea entre todos. Al concluir deberán encaminarse todos a la meta, y una vez en la meta, enviar al *micro:bit* árbitro los mensajes secretos de todas las balizas descubiertas por el equipo. El equipo ganador será el que antes consiga localizar todas las balizas, debiendo regresar a la meta todos los robots del equipo, contabilizándose como tiempo del equipo el instante en el que el último robot del equipo envía el mensaje al árbitro.

Cambios

Versión 1.1: Reestructurados los formatos de los mensajes.

1. Introducción

Un transmisor de ondas de radio puede utilizarse para enviar información a un receptor. La telefonía móvil, la radio, la TV, las redes WiFi y Bluetooth, transportan información multimedia de este modo.

Pero hay muchos otros usos de las ondas de radio. Por ejemplo un receptor de radio puede averiguar en qué posición geográfica se encuentra analizando la potencia y el retardo con los que recibe las señales emitidas por balizas emisoras cuya localización geográfica es conocida de antemano por el receptor. Tres ejemplos de este tipo de uso de las ondas de radio son: Los sistemas de localización en exteriores mediante los satélites GPS o Galileo, las radiobalizas utilizadas en aeronavegación, y los sistemas de localización en interiores para robots que aprovechan las ondas de radio de las tecnologías WiFi y Bluetooth para que el robot conozca su posición.

También puede utilizarse la información de la potencia con la que se recibe la señal de radio de un emisor para que el receptor pueda averiguar donde se encuentra el emisor. Este mecanismo es utilizado por las fuerzas de orden público para localizar delincuentes analizando la potencia con la que se reciben las señales emitidas por el teléfonos del delincuente en diferentes receptores. También se usa esta técnica para detectar transmisiones de radio no autorizadas que afectan al uso del espectro radioeléctrico.

Este proyecto se enmarca dentro de este segundo tipo de aplicaciones que utilizan la potencia de la señal transmitida por emisores de localización desconocida.

1.1. Uso de la radio del *micro:bit*

Los *micro:bit* tienen una radio que emite y recibe en la banda de 2.4GHz, la misma utilizada por WiFi y Bluetooth¹. Con MicroPython puedes importar el módulo `radio` que ofrece varias funciones para utilizar la radio: encenderla y apagarla, recibir y enviar mensajes a otros *micro:bit*, conocer la potencia de la señal que codifica un mensaje que se recibe, configurar la potencia con la que se quiere transmitir un mensaje, configurar el grupo en el que se envían los mensajes, etc.

Todos los mensajes se envían mediante radiado o *broadcast* en un mismo canal de radio en el que por omisión están sintonizadas todas las radios de los *micro:bit*. Esto quiere decir cualquier *micro:bit* recibe todos los mensajes que envían los demás.

Cuando no se quiere que todos los *micro:bit* reciban todos los mensajes se pueden utilizar los **grupos**. En la interfaz de programación de la radio del *micro:bit* se puede seleccionar un número de grupo comprendido entre 0 y 255. Por omisión todos los *micro:bit* utilizan el grupo 0.

Si un *micro:bit* configura en su radio otro número de grupo y envía mensajes, sólo los *micro:bit* que hayan configurado ese mismo número de grupo recibirán esos mensajes².

De esta forma por ejemplo se puede seleccionar el grupo 7: `radio.config(group=7)`

En esta práctica se utilizan varios grupos distintos que podrás configurar con esta sencilla llamada.

2. Estrategia de búsqueda de una baliza

En esta sección comenzaremos suponiendo que hay una sola baliza y en la siguiente sección consideraremos estrategias para cuando hay más de una baliza.

El objetivo es que el robot se mueva hasta que esté cerca de la baliza, es decir, hasta que reciba con una potencia suficientemente alta los mensajes periódicos que transmite la baliza.

La antena de radio del *micro:bit* que lleva el robot no le permite saber de qué dirección provienen los mensajes que se reciben. Sin embargo, el valor de la potencia de la señal con la que se reciben los mensajes de la baliza puede utilizarse para decidir en qué dirección ha de moverse el robot: la estrategia que puede seguir el robot para conseguir el objetivo es continuar moviéndose hacia adelante mientras que la potencia con la que se reciben los mensajes de la baliza sea mayor o igual a la de los mensajes recibidos con anterioridad. Sin embargo, cuando la potencia con la que se reciben los nuevos mensajes sea menor que la potencia con la que se recibieron los mensajes anteriores, el robot girará hacia un lado para intentar que en los siguientes movimientos hacia adelante se reciban con mayor potencia los mensajes, pues ello será indicativo de que se está acercando a la baliza.

Hay que tener en cuenta que la propagación de las señales está sometida a interferencias ocasionadas por obstáculos que las ondas se encuentran en su camino, así como a rebotes de la señal en los objetos, que pueden hacer que no siempre la recepción de una señal con mayor potencia implique que se está más cerca del receptor. El robot puede evitar este tipo de distorsiones de la siguiente forma: en lugar de tener en cuenta la medida de potencia de todos y cada uno de los mensajes recibidos para decidir si sigue hacia adelante o si gira, puede tomar la decisión en función de un estadístico como la media de un conjunto de medidas de potencia. De esta forma se atenuarán las medidas distorsionadas de potencia que puedan estarse recibiendo.

En el siguiente vídeo puedes ver una simulación del comportamiento de un robot que intenta aproximarse al *micro:bit* baliza que aparece dibujado, parando cuando recibe mensajes con una potencia suficientemente grande al haberse acercado suficientemente al *micro:bit* baliza:

¹La radio de los *micro:bit* puede utilizarse con la jerarquía de protocolos de comunicación Bluetooth, si bien la interfaz que usaremos en la práctica no usa Bluetooth sino un protocolo mucho más sencillo cuya interfaz de programación fue introducida en el enunciado de la Fase 1

²En realidad todos los *micro:bit* siguen recibiendo la señal de todos los transmisores que están en su radio de alcance, pero el SW del receptor elimina los mensajes que no vienen dirigidos al grupo en el que está configurado el receptor.

Un *micro:bit* baliza oculto

Haz click encima de esta figura para que comience la animación.

Figura 1: Puedes controlar el vídeo: pararlo, avanzar y retroceder. OJO: evince tiene algunos problemas con la reproducción. Utiliza la aplicación okular para visualizar este documento PDF. Utiliza el comando `sudo apt install okular` para instalar okular en Ubuntu 18.04. Si tienes problemas, la película está en el aula virtual de la asignatura

2.1. Formato de mensaje con el identificador de la baliza

Todas las balizas envían periódicamente mensajes que contienen su identificador. Estos mensajes se envían destinados a un grupo cuyo número es 41.

Para recibir los mensajes con los identificadores de las balizas tienes que configurar la radio para que filtre sólo los mensajes enviados a través de ese grupo: `radio.config(group=41)`.

Estos mensajes consisten en un String con el siguiente **formato**:

`00:idBaliza`

En primer lugar aparece el prefijo `00`, seguido del carácter `:` y del identificador de la baliza, que es un número impar entre 1 y 255, excepto el número 41 y el 43. Así, por ejemplo, el beacon con id 7 enviaría este mensaje: `00:7`.

Recuerda del enunciado de la Fase 1 que los 3 primeros bytes del mensaje no forman parte del String por lo que los tienes que descartar antes de obtener el String cuando recibes con la función `radio.receive_full()`. Esta función proporciona, además del mensaje recibido, la potencia de la señal del mensaje recibido.

3. Estrategias de búsqueda de múltiples balizas

Cuando haya más de una baliza, todas ellas estarán transmitiendo periódicamente su identificador en el grupo 41. Por tanto el robot tendrá que decidir hacia qué baliza dirigirse en cada momento.

El robot puede saber cuántas balizas hay contando los identificadores distintos que recibe en los mensajes que emiten las balizas. Recuerda que en cualquier posición del aula podrás recibir mensajes de todas las balizas.

Existen diversas estrategias para abordar la búsqueda de múltiples balizas. Una de ellas consiste en elegir una de las balizas aún

no localizadas, y seguir la estrategia de la sección anterior hasta encontrarla. En el siguiente vídeo puedes ver una simulación de este tipo de estrategia, con 10 *balizas*. En este caso, para que la simulación sea más comprensible, hemos dibujado un círculo rojo debajo del *micro:bit* baliza cuya potencia está tomando como referencia en cada momento el robot para moverse hacia él.



Figura 2: Puedes controlar el vídeo: pararlo, avanzar y retroceder. OJO: evince tiene algunos problemas con la reproducción. Utiliza la aplicación okular para visualizar este documento PDF. Utiliza el comando `sudo apt install okular` para instalar okular en Ubuntu 18.04. Si tienes problemas, la película está en el aula virtual de la asignatura

Como puedes ver en el vídeo, el robot utiliza una estrategia no óptima: a veces pasa cerca de uno de los *micro:bit* baliza pero lo ignora, a pesar de que la potencia con la que se están recibiendo sus mensajes, al estar más cerca, será mayor que la potencia con la que está recibiendo los mensajes enviados por el *micro:bit* baliza que tiene como objetivo (el del círculo rojo en la simulación), al estar éste más lejos.

Se puede utilizar una estrategia mejor que hará que el robot recorra menos distancia y por lo tanto emplee menos tiempo para localizar todas las balizas. En este otro vídeo puedes ver al robot utilizando una estrategia en la que va cambiando de baliza objetivo cada vez que recibe un mensaje de una baliza con una potencia mayor que la del *micro:bit* que estaba fijado hasta entonces como objetivo. En la simulación puedes apreciar cómo el robot va cambiando la baliza objetivo pues el círculo rojo va cambiando según el robot se acerca a nuevas balizas:

Haz click encima de esta figura para que comience la animación.

Figura 3: Puedes controlar el vídeo: pararlo, avanzar y retroceder. OJO: evince tiene algunos problemas con la reproducción. Utiliza la aplicación okular para visualizar este documento PDF. Utiliza el comando `sudo apt install okular` para instalar okular en Ubuntu 18.04. Si tienes problemas, la película está en el aula virtual de la asignatura

4. Obtención del secreto de una baliza tras llegar el robot a sus alrededores

Utilizando alguna de las estrategias de búsqueda de balizas explicadas en secciones anteriores, el robot se aproximará hasta estar lo suficientemente cerca de cada una de las balizas.

El intercambio de mensajes entre el robot y la baliza para obtener el secreto se produce en un grupo distinto, con un número de grupo igual al identificador de la baliza.

Para poder obtener el secreto de una baliza el robot tiene que enviarle un mensaje pidiendo que le envíe el secreto. La baliza contestará enviando un mensaje secreto con el nivel de potencia más bajo, para asegurarse de que sólo reciben sus mensajes con el secreto los robots que estén suficientemente cerca.

Si tras enviar el mensaje de petición de secreto no se recibe la respuesta de la baliza, puede ser porque se haya perdido el mensaje de petición o el de respuesta. Pero también puede ser porque el robot no se encuentre lo suficientemente cerca de la baliza. Por ello, tras enviar varias veces el mensaje de petición de secreto sin obtener respuesta, el robot debería intentar acercarse más a la baliza.

4.1. Formato del mensaje de petición de secreto que envía el robot a la baliza

El identificador que transmite periódicamente la baliza es el grupo en el que puede obtenerse el secreto. El robot usará ese grupo para enviar un mensaje de petición de secreto.

Estos mensajes consisten en un String con el siguiente **formato**:

```
01:dameSecreto
```

En primer lugar aparece el prefijo `01`, seguido del carácter `:` y del texto literal `dameSecreto`.

Estos mensajes se deben enviar con la potencia mínima. Para configurar la potencia mínima de transmisión ejecuta esta llamada:

```
radio.config(power=0, group=X)
```

siendo X el grupo igual al identificador de la baliza.

NOTA: Ten en cuenta que, aunque cambies de grupo con una nueva llamada a `config`, en la cola de mensajes recibido puedes tener todavía algún mensaje dirigido al grupo anterior.

4.2. Formato del mensaje con el secreto que envía la baliza al robot

Tras recibir el mensaje de petición de secreto que le envía el robot, la baliza envía como respuesta un mensaje utilizando el mismo grupo, que incluye el secreto en un String.

Estos mensajes consisten en un String con el siguiente **formato**:

```
02:secreto
```

En primer lugar aparece el prefijo `02`, seguido del carácter `:` y de un texto cuyo contenido será el secreto de esa baliza.

Este mensaje lo envía la baliza con la potencia mínima por lo que sólo si está muy cerca de la baliza el robot podrá recibirlo.

5. Evitar obstáculos

Mientras que el robot se está moviendo siguiendo su estrategia para llegar hasta una baliza o hasta la meta debe detectar si hay un obstáculo en su trayectoria. Para ello utilizará el sensor láser o de distancia. El obstáculo puede ser una pared o cualquier otro objeto que se interponga en su camino, como por ejemplo otro robot, una mesa, una silla, etc.

Cuando el robot detecte un obstáculo deberá llevar a cabo alguna acción que le permita evitar el obstáculo antes de seguir con su estrategia de acercamiento a su destino, ya sea éste una baliza o la meta.

6. Llegada a la meta

Tras haber obtenido los secretos de todas las balizas, el robot debe dirigirse hacia la meta. La meta está señalizada mediante una luz en el techo y un rectángulo negro en el suelo. El robot puede utilizar los dos sensores de luz de su parte superior para dirigirse hacia la meta siguiendo la intensidad creciente de la luz. Al tener dos sensores derecho e izquierdo en su parte superior, el robot puede seguir una estrategia que le acerque más rápidamente a la meta que cuando su objetivo es acercarse a una baliza: si percibe más luz por el sensor derecho que por el izquierdo, puede girar a la derecha; si por el contrario percibe más luz por el sensor izquierdo que por el derecho, puede girar a la izquierda.

El rectángulo negro en el suelo puede ser utilizado por el robot para detectar cuándo ha llegado a la meta con los sensores de línea de su parte inferior.

6.1. Formato del mensaje que envía el robot al árbitro en la meta

Tras parar en la meta, el robot enviará los secretos al *micro:bit* árbitro, que estará **recibiendo en el grupo 43** mensajes enviados por el robot.

Estos mensajes consisten en un String con el siguiente **formato**:

```
03:loginAlumno,idBaliza1:secreto1,idBaliza3:secreto3
```

En primer lugar aparece el prefijo `03`, seguido del carácter `:` y del login del alumno en el aula de prácticas. A continuación, separados por comas, irán las parejas de identificador de baliza y secreto, separados por el carácter `:`.

El robot deberá reenviar el mensaje al árbitro hasta que consiga recibir el mensaje de confirmación.

Estos mensajes se deben enviar con la potencia mínima. Para configurar la potencia mínima de transmisión ejecuta esta llamada:

```
radio.config(power=0, group=43)
```

6.2. Formato del mensaje de confirmación que envía el árbitro al robot

Tras recibir el anterior mensaje el árbitro contestará con un mensaje de confirmación al robot.

Estos mensajes consisten en un String con el siguiente **formato**:

`04:loginAlumno`

En primer lugar aparece el prefijo `04`, seguido del carácter `:` y del login del alumno que envió el mensaje al árbitro.

7. Pautas de implementación

Recuerda todo lo aprendido en la asignatura. Aborda el problema dividiéndolo en partes, y escribiendo código limpio en funciones con nombres adecuados que realicen una única tarea que quede reflejada en su nombre.

No intentes escribir todo el código de una vez. Céntrate en uno de los subproblemas y depura tu código hasta que funcione antes de intentar abordar el siguiente problema.

Exponemos a continuación algunas pautas que pueden ayudarte a estructurar el código.

El robot puede almacenar en una variable un String que refleje el **estado** en el que se encuentra mientras que intenta alcanzar el objetivo final de llegar a la meta con los secretos obtenidos de las balizas. De esta forma en todo momento el programa podrá comprobar en qué estado se encuentra para decidir qué hacer a continuación.

Hasta que no llegue al estado final en el que ha conseguido transmitirle al árbitro los secretos, el robot tendrá que estar continuamente comprobando los valores de sus sensores y de los mensajes recibidos por un lado, y por otro tendrá que realizar una u otra acción dependiendo de los valores de los sensores, de los mensajes recibidos, y del estado de la misión en el que se encuentre.

Por ello es buena idea que estructures el programa del robot con un bucle en el que en cada iteración se haga lo siguiente:

1. lee y almacena los valores de los sensores, y lee y procesa todos los mensajes de radio que puedan haberse recibido (lee todos los que haya hasta que la llamada `radio.receive_full()` devuelva `None`), y actualiza variables que contabilicen el paso del tiempo
2. en función tanto del estado actual del robot, como de los valores leídos de los sensores, de los mensajes de radio recibidos, y de las variables que contabilizan el paso del tiempo, el programa decide qué acciones debe realizar el robot (enviar mensaje de radio, moverse, girar, actualizar variables que contabilizan el paso del tiempo,...) y a qué nuevo estado debe pasar (que puede ser el mismo en el que se encontraba).
3. finalmente se llevan a cabo las acciones y se actualiza la variable que almacena el estado del robot

El bucle terminará una vez el robot haya alcanzado el estado final que refleje que ha alcanzado su objetivo.

Aparte de la variable que almacena el estado en el que se encuentra el robot y las variables en las que se almacenan los valores de los sensores y los de los mensajes recibidos y las variables que contabilizan el paso del tiempo, se pueden utilizar otras variables para almacenar por ejemplo el identificador de la baliza objetivo, los identificadores y potencias de recepción de mensajes de las balizas, etc.

A modo de ejemplo, el robot puede comenzar en un estado inicial `buscando-baliza`, pasando después por la siguiente secuencia de estados: si la potencia de nuevos mensajes enviados por la baliza se incrementa respecto a valores leídos anteriormente, pasará al estado `buscando-baliza`, realizando la acción de moverse hacia adelante, y si se decrementa la potencia, pasará también al estado `buscando-baliza`, pero realizará la acción de girar hacia un lado. Pasará al estado `baliza-encontrada` cuando la potencia de la señal sea suficientemente grande, realizando la acción de enviar el mensaje de petición de secreto. Si pasa cierto tiempo y sigue en el estado `baliza-encontrada` sin que se haya recibido el mensaje con el secreto³, pasará al estado `baliza-encontrada` y realizará la acción de enviar de nuevo el mensaje de petición de secreto. Si el tiempo acumulado en este estado es excesivo⁴, pasará al estado `buscando-baliza` para acercarse más a la baliza. Tras recibir el secreto, si aún quedan balizas, volverá al estado `buscando-baliza`, y si no pasará al estado `buscando-meta`. En función de

³Para saber cuánto tiempo ha pasado en el estado `baliza-encontrada` desde que se envió el mensaje se puede utilizar una variable que se inicializa a 0 cuando se pasa a este estado y se incrementa en una unidad cada vez que se da una vuelta al bucle estando en este estado sin que se haya recibido el mensaje con el secreto.

⁴Se puede por ejemplo utilizar otra variable que acumule el número de veces que se ha enviado el mensaje de petición de secreto sin haber recibido respuesta

los valores detectados por los sensores de intensidad lumínica, pasará al estado `buscando-meta` realizando la acción de girar a la derecha, o girar a la izquierda, y se moverá hacia adelante. Cuando estando en el estado `buscando-meta` detecte el rectángulo negro de la meta, pasará al estado `en-meta-mensaje-enviado-a-arbitro`, realizando como acción el envío del mensaje al árbitro. Si no consigue recibir el mensaje de confirmación del árbitro después de un tiempo, volverá al estado `en-meta-mensaje-enviado-a-arbitro` y enviará el mensaje de nuevo.

No tienes por qué utilizar el mismo número de estados ni los mismos nombres. Considera el párrafo anterior como un ejemplo ilustrativo.