# Xarxes

## Pràctica 2: Comunicacions de dades en l’entorn d’“Ambient Intelligence”

### Blai Ras i José Peña

1. ***Objetivos***

En esta práctica hacemos el primer contacto con el que será nuestra herramienta para el resto de prácticas: la mota. Concretamente, tendremos que programarla para que transmita y datos en serie, estudiando así sus protocolos, su recepción, cómo está estructurada su capa física y cómo la distancia se relaciona con la recepción de datos y la ganancia de ellos.

1. ***Material***
   1. La mota
   2. Code Composer
   3. Connector JTAG
   4. Programador MSP-FET430UIF
   5. Sniffer
2. ***Procedimiento***

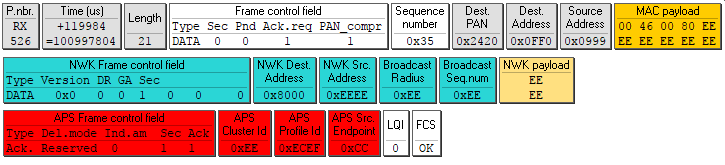
Una vez bajado el código, lo primero que hicimos fue intentar entenderlo. Lo primero que se nos pidió fue comprender el *main* del programa. En él, hay una variable *booleana* llamada “bucleinicial”. Esta, se activa al pulsar el botón *user* de la mota, de manera que ahora podremos enviar la información solo cuando pulsemos dicho botón.

Esta información, concretamente, son tramas que siguen el protocolo IEEE 802.15.4. Se nos pide que modifiquemos el código de tal forma que la mota pueda enviar 100 tramas en un intervalo de 10 milisegundos. Sin embargo, para coger mejores medidas, el profesor nos indicó que probáramos antes con enviar 10. En consecuencia, en primer lugar establecimos un *halWait(10000)* en este método *main*, justo debajo del envío de esta trama. Con esta línea, llamamos a la función *halWait* de nuestra librería, la cual permite parar el tiempo de reloj durante un cierto tiempo. Este tiempo se especifica en microsegundos, así que pasándole 10000, le estamos diciendo que espere (sin hacer nada) 1 segundo, tal y como nos piden. En segundo lugar, hicimos que la mota enviará 10 tramas con un *for* justo debajo del *if* inicial (*if(!bucleinicial)).*

Sin embargo, para que el *sniffer* reciba nuestras tramas, tuvimos que cambiar el canal y la dirección de nuestra mota, ya que ¡había más estudiantes con distintas motas enviando información! Por consiguiente, en apartado de constantes del código, pusimos nuestro canal a 13 y nuestra variable *MYADDR*  a 0x999. De esta forma, sabíamos que la información que llegaba al *sniffer* era nuestra.

1. ***Resultados***

Aquí expondremos los resultados realizados de enviar un cierto número de tramas a cierta distancia, comentando así como afecta esta al recibimiento en el *sniffer*. Primero de todo, definimos cómo está formada una trama (los campos importantes y conocidos):



* **P.nbr** indica el tipo de trama (recepción/transmisión) y el numero de trama correspondiente. En este caso, 526.
* **Time** indica “la fecha” del envió de esta trama, es decir, cuánto tiempo ha pasado desde el envío de esta hasta que se ha recibido.
* **Length** es la longitud de esta trama. Debería ser siempre 21, aunque si hay algún tipo de error es más pequeña.
* **Frame Control field** es un campo con distintas secciones. El primero, type, indica el tipo de información que se ha recibido. En este caso, Data. El penúltimo, Ack. Req significa si ha recibido (un 1) o no (un zero) una solicitud de Ack (*acknowledgement*). El resto de campos no son importantes.
* **Sequence Number** indica, por lo tanto, el número de secuencia.
* **Dest. PAN** (*Personal Area Network)* indica el tipo de red que usamos (protocolo) que usa nuestra mota-*sniffer*.
* **Dest. Adress** indica la dirección de destino. En este caso, es la dirección del *sniffer* del profesor. En cambio, Source Adress es la dirección de origen. Podemos ver 0x0999, que es la especificada en nuestro código.
* **MAC Payload** es nuestro *payload*.
  1. Con 10 tramas:

|  |  |
| --- | --- |
| Distancia (m) | # tramas recibidas |
| 0 | 10 - 0 (10) |
| 1,5 | 21 - 11 (10) |
| 2,5 | 35 - 25 (10) |
| 3,5 | 47- 39 (8) |
| 4,5 | 57 - 47 (10) |
| 5,5 | 80 - 71 (9) |
| 7,5 | 89 - 80 (9) |
| 10,5 | 105 - 96 (9) |
| 14 | 116- 106 (10) |
| 16,5 | 126 - 116 (10) |

Con 10 tramas, a distancias separadas más o menos por 1 metro, tuvimos los siguientes resultados:

A cada distancia, nosotros enviábamos 10 tramas teóricamente. A distancia zero, se recibieron todas. Sin embargo, a 3.5 metros vemos el primer error: una trama no llegó al *sniffer.* Sin embargo, a 4.5 metros vuelven a llegar todas. Esto puede ser debido a una gran cantidad de causas. En primer lugar, puede ser relacionado con la potencia de transmisión de nuestra mota o de la potencia de recepción del *sniffer.* Tal y como esta fórmula indica, la distancia es un factor que afecta a la potencia de recepción:

, donde Prx es la potencia de recepción, Ptx es la potencia de transmisión, Gtx es la ganancia de transmisión, Grx es la ganancia de recepción, λ es la longitud de onda, d es la distancia y n un coeficiente de correlación que oscila entre 0 y 1. Sin embargo, también puede ser debido al ruido de la transmisión o algún defecto de la mota. De hecho, el *sniffer* en ocasiones nos dice cuándo una trama se ha recibo incorrectamente, como a continuación, que podemos ver un *payload* de longitud negativa:

C:\Users\mat.aules\Downloads\payload_negativo.png

En total, de media, se reciben 9,5 tramas por distancia de las 10 que se envían.

En teoría, deberíamos ver una disminución del número de tramas recibidas a medida que nos alejamos del *sniffer*. Sin embargo, con 10 tramas y con esas distancias, no se puede observar en las mejores condiciones este efecto.

* 1. Con 100 tramas

Es por eso que a continuación pasamos a enviar 100 tramas. Las distancias esta vez también las modificamos. Cogemos, de referencia, los azulejos del suelo, para medir aproximadamente cada 5 metros, hasta llegar a 25. Los resultados son los siguientes:

|  |  |
| --- | --- |
| Distancia (m) | # tramas recibidas |
| 1 | 100 - 0 (100) |
| 5 | 200 - 101 (99) |
| 10 | 299 - 201 (98) |
| 15 | 400 - 300 (100) |
| 20 | 401 - 480 (80) |
| 25 | 565 - 481 (84) |

Aquí podemos observar mejor que a medida que nos acercamos, el *sniffer* recibe menos tramas. Concretamente, aquí la media es de 93,5 tramas recibidas por distancia, aunque teóricamente se envían 100. Lo podemos ver claramente en la gráfica: la recta de regresión tiene pendiente negativo, indicando así una disminución del número de tramas recibido en relación a la distancia de envío. Sin embargo, hay dos puntos “surrealistas”, a distancia 10, ya que se reciben 98 (cuando a 15 se vuelven a recibir 100) y a distancia 20, ya que se reciben 80 cuando a 25 metros se reciben 84.

En teoría, la disminución de tramas debería ser inversamente proporcional a la distancia de envío, y por lo tanto, en ningún caso se deberían recibir más tramas que en la anterior.

A parte, hemos decidido calcular la Potencia de recepción teoría por cada distancia con la ecuación de Friis. Lo hacemos con dB y no Watts:

Respecto a la ecuación de Friis, en nuestro caso, al valer GTX, GRX y n 1, no repercuten en la ecuación y los podemos obviar. Además, podemos sustituir el valor PTX por 1 mWatt, o lo que es equivalente, 0,001 Watt. El resultado sería:

|  |  |  |
| --- | --- | --- |
| Distancia (m) |  | Prx (dB) |
| 1 | -40,04 | -38,04 |
| 5 | -54,02 | -52,025 |
| 10 | -60,04 | -58,04 |
| 15 | -63,56 | -61,56 |
| 20 | -66,06 | -64,06 |
| 25 | -68,004 | -66,004 |

Podemos ver como la potencia de recepción disminuye a medida que nos alejamos. Esto tiene todo el sentido del mundo: la potencia disminuirá si nos alejamos del *sniffer*, es decir, el señal recibido será menor. Además, coincide con los -65 dBm que nos indica el enunciado de la práctica.

1. ***Dificultades presentadas y como se han solucionado***
   1. Entender el *main*: tardamos un poco a entender que el *main* no se ejecutaba hasta pulsar el botón de nuestra mota, para luego enviar las tramas (el condicional *bucleInicial*.
   2. Colocación del *for*: al disponernos a programar el *for* para enviar 10 o 100 tramas, lo pusimos precisamente antes de este condicional. En consecuencia, seguía enviando únicamente una trama.
   3. También tuvimos problemas al compilar nuestro programa, ya que el Code Composer se quejaba del tipo de *debugador*. Por desgracia, no supimos cómo se arregló.
2. ***Conclusiones***

En esta práctica hemos aprendido el funcionamiento básico de nuestra mota, su protocolo de envío y el tipo de información (la trama) que está transmite. Para entenderlo, hemos hecho una simple recogida de datos que varían en función de la distancia. De esta forma, hemos tomado contacto con la ecuación de Friis, la cual nos ha servido para relacionar la ganancia, la potencia y la distancia de una transmisión.