



Pinchadiscos *Inteligente*: Sensorización

Pablo Lopez Matencio, <pablo.lopez@upct.es>

Objetivos

El pinchadiscos inteligente es un sistema que, en tiempo real, interpreta si una canción es del agrado del público. Para ello, detecta el movimiento producido al bailar; si existe una correlación entre ese movimiento y el ritmo (tempo) de la canción, infiere que esa canción “gusta”, y, si no, “no gusta”.

Estas páginas presentan el entorno de trabajo del laboratorio y los dispositivos que utilizaremos para detectar el movimiento y obtener la señal que dicho movimiento produce.

El profesor acompaña en a este documento Jupyter Notebooks con código en Python donde se explica cómo obtener la *frecuencia dominante* o *principal* de una señal, así como los ejercicios que debe realizar.

El objetivo es realizar una función en Python que calcule la frecuencia dominante del movimiento cada vez que la llamemos.

Entorno de trabajo

La práctica se realizará en grupos de dos alumnos. El software necesario para la elaboración de los ejercicios estará disponible para todos los grupos.

- Raspberry Pi
- x-BIMU 802.15.4 XStick USB interfaz.
- x-BIMU *inertial measurement unit* (IMU) con interfaz xBee.

El alumno trabajará con el escenario siguiente:

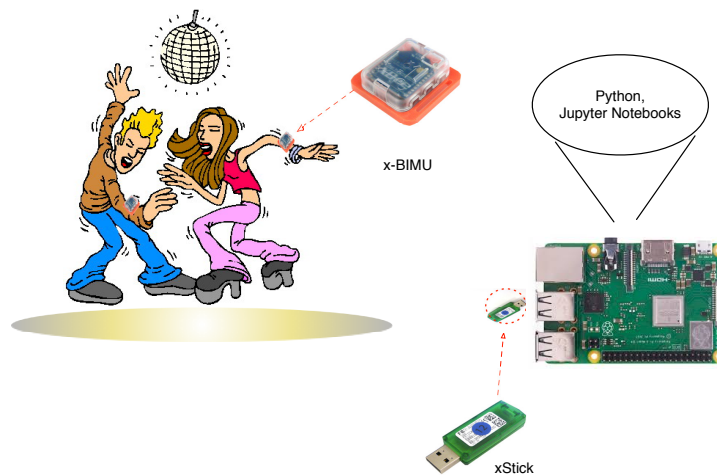


Figura 1: Esquema general del sistema

Duración y Evaluación

- La duración del curso.
- Cada grupo tiene que entregar los ejercicios propuestos en clase.

1 Sistemas de Inteligencia Ambiental

El concepto y la definición de los sistemas de inteligencia ambiental (AmI) fueron presentados por primera vez en el marco de los proyectos estratégicos de la EU. El informe *Software technologies, embedded systems and distributed systems*, que, en su página 9, presenta el concepto de AmI como:

“... The emphasis of AmI is on greater user-friendliness, more efficient services support, user empowerment and support for human interactions. People are surrounded by intelligent intuitive interfaces that are embedded in all kinds of objects. The Ambient Intelligent environment is capable of recognising and responding to the presence of different individuals. Ambient Intelligence works in a seamless, unobtrusive and often invisible way.”

En suma la visión de un sistema de AmI es aquel que ayuda al individuo (de forma personalizada) en la labor que esté desempeñando, sin interferir en ella. Para ello, el sistema debe

1. tener capacidad de capturar los datos necesarios del entorno y del individuo,
2. tener *la inteligencia* necesaria para entender la situación en que se encuentra y
3. *tomar una decisión* para ayudarlo.

Nuestro objetivo en la presente práctica es desarrollar un sistema de AmI que monitoree el movimiento de las personas de una sala donde suena música (p.ej. una discoteca o un pub). Esta parte se integra con la parte de análisis del tiempo y de toma de decisiones.

2 Introducción al entorno de trabajo

En esta sección introducimos algunos conceptos básicos necesarios para comprender la funcionalidad de los sensores de medición inercial (IMUs), cómo configurarlos y utilizarlos.

2.1 Sensores de movimiento inercial

El sensor x-BIMU es un dispositivo IMU / **ARHS** lo que significa que dispone de giróscopo, acelerómetro y magnetómetro en los tres ejes.

Accelerómetro Un **acelerómetro** es un sensor que mide aceleración (la velocidad de *cambio* respecto al tiempo), y está representado por un vector con magnitud y dirección (X, Y, Z). El acelerómetro mide en unidades de g , que es la aceleración de la gravedad ($9.8m/s^2$). El acelerómetro instalado en x-BIMU es del tipo MEMS, y puede medir aceleraciones de hasta $\pm 16g$. Los acelerómetros pueden medir vibraciones, ligeras inclinaciones, choques y movimiento de objetos.

Gir6scopo Un gir6scopo mide velocidad angular en grados por segundo (dps). Al igual que el aceler6metro es un vector con magnitud y direcci6n (X, Y, Z). El dispositivo x-BIMU contiene un procesador de movimiento **IMU-3000** de tecnologa MEMS, cuyo rango de medida es de $\pm 2000dps$, permitiendo medir con precisi6n tanto movimientos r6pidos como lentos. F6sicamente un **gir6scopo** es un sistema cuyas las diferentes partes se mueven libremente en el espacio de forma que se conserva el momento angular cuando una fuerza externa se aplica sobre 6l.

Magnet6metro El **magnet6metro** es un sensor que permite medir intensidad y direcci6n (X, Y, Z) del campo magn6tico en un ‘punto’. El rango de aplicaciones es variado, por ejemplo, permite determinar el **recorrido realizado por una persona** en espacio abierto (en ausencia de GPS), o medir los efectos de de las tormentas solares en la magnetosfera terrestre (ver **enlace**). La unidad de medida es Gauss (G) en honor a su inventor **Carl Friedrich Gauss**. En x-BIMU el magnet6metro est6 basado en tecnologa MEMS. La reducci6n en el tama1o actual de estos sensores ha permitido incorporarlos a **tel6fonos** m6viles y plataformas de juegos.

Por lo tanto, los sensores incorporados en x-BIMU miden orientaci6n y movimiento, no posici6n.

La interfaz de comunicaci6n est6 compuesta por el m6dulo inal6mblico **Digi XBee 802.15.4**. Esto nos permite utilizar hasta 12 x-BIMUs transmitiendo simult6neamente, cada uno a un *interfaz* USB diferente, y por un canal diferente. Podemos ajustar la latencia para cada canal y as6 disponer de los datos capturados en, *cuasi* tiempo real.

Para ello, el interfaz USB emplea **controladores *future technology devices* (FTDI) de puerto serie para USB**. Por defecto, la latencia est6 fijada a $16ms$. Este valor permite mantener en el PC una sincronizaci6n precisa de los paquetes recibidos del x-BIMU. A6n podr6amos modificar y reducir el valor de la latencia hasta $1ms$ (el m6nimo) v6a Device Manager > USB Serial Port Properties > Advanced (ver Figura 2).

El alumno mantendr6 los valores de latencia por defecto y s6lo los reducir6, siguiendo el procedimiento anterior, si percibe que el sistema responde con *lentitud*.

2.2 x-BIMU terminal

El software **x-BIMU TERMINAL**, o, simplemente **TERMINAL**, nos permite ajustar los par6metros de configuraci6n de x-BIMU tales como canal de radiofrecuencia, selecci6n del puerto COM, arrancar y parar la captura de datos, etc. Tal y como se muestra en la Figura 1 el **TERMINAL** lo usamos para controlar x-BIMU y visualizar

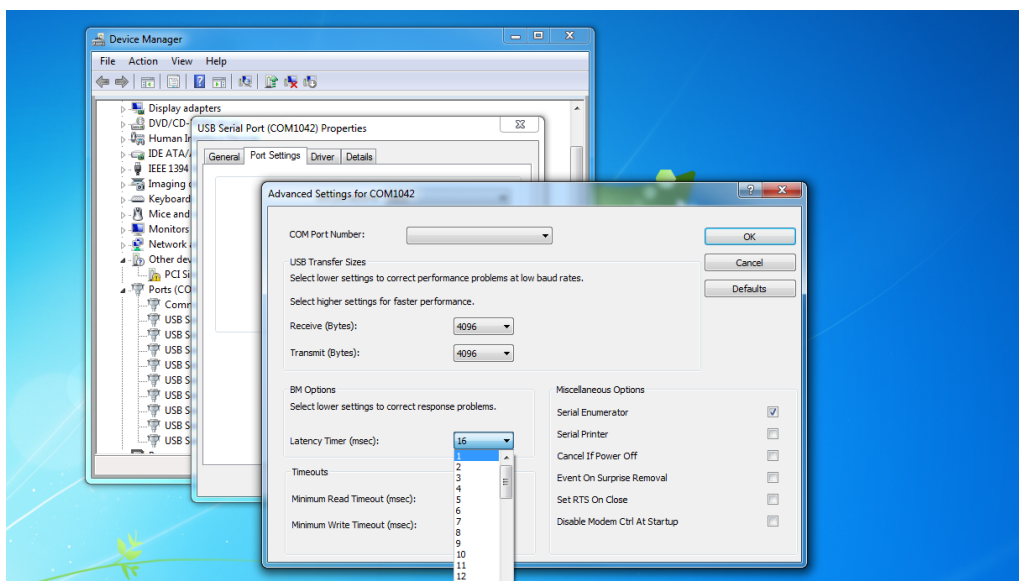


Figura 2: Configuración del puerto USB

datos. El software necesario para la realización de esta práctica puede descargarlo de la página del Aula Virtual.

Para comenzar a recibir datos capturados por x-BIMU, debemos realizar lo siguiente:

1. Instalar **x-BIMU TERMINAL** (si no lo está ya).
2. Insertar el interfaz USB (**xSTICK**) en el puerto USB de la computadora.
3. En la ventana principal de **TERMINAL**, seleccionar el puerto serie **COM3** (ó el que asigne el PC) dentro del menú **Serial Port** (Figura 4). Si no aparece algún puerto, entonces seleccione *Refresh*.
4. Encender x-BIMU. Para ello, presionar la superficie ‘cuadrada’ de la esquina superior derecha del dispositivo (ver Figura 5), hasta que la luz comience a parpadear, y entonces, deje de pulsar. El alumno deberá seguir la operativa de este punto para, al terminar de sesión de trabajo, apagar el dispositivo y que no agote la batería.

El color del led dentro de x-BIMU indica el canal por el que está transmitiendo datos. Por ejemplo, color ‘azul’ indica ‘canal 12’.

Una vez que x-BIMU está encendido comienza a enviar datos al xSTICK, y comienzan a mostrarse dichos datos por la ventana de **TERMINAL**. Primero, muestra un ‘OK’ al terminar el test inicial del dispositivo. Las siguientes líneas son

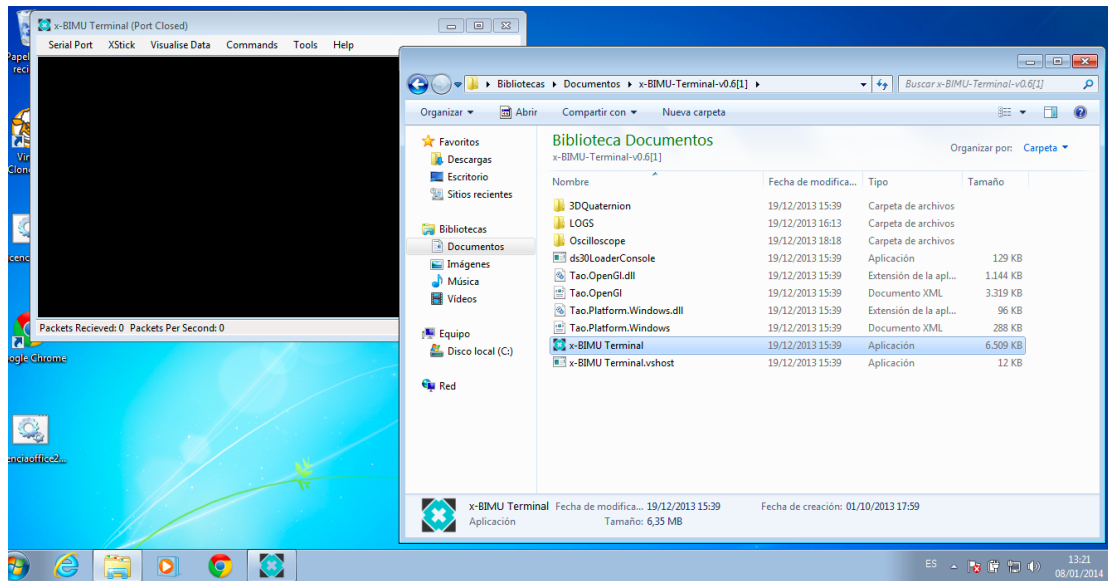


Figura 3: Ejecutable (.exe) del programa TERMINAL. En segundo plano, ventana principal de TERMINAL

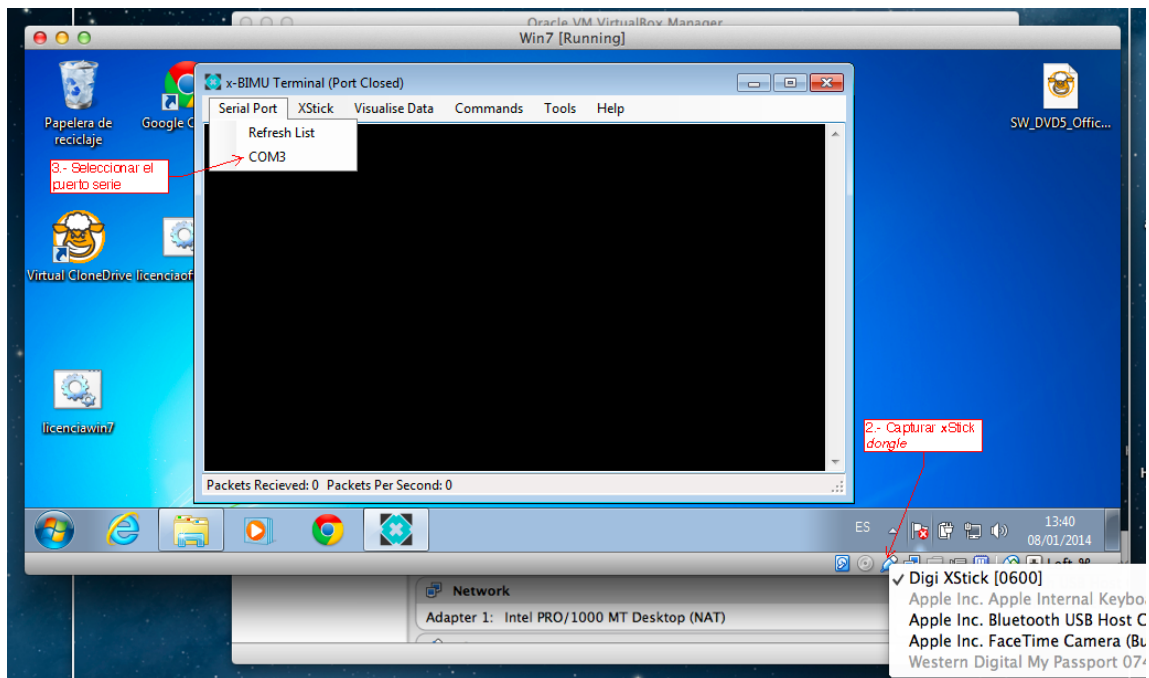


Figura 4: Configuración de la MV y aplicación TERMINAL para recibir datos desde x-BIMU



Figura 5: x-BIMU. Encendido/apagado presionando en la parte superior derecha de datos y cada una de ellas está etiquetada con una ‘letra’, tal y como muestra la Figura 6; cada una de estas líneas corresponde con un paquete de datos, y su significado es el siguiente:

- **Q**, w, x, y, z, contador, checksum \leftrightarrow . Los valores de w, x, y, z son el *quaternion*. Un *quaternion* está formado por cuatro números complejos que representan la orientación de un cuerpo en un espacio 3D.
- **S**, gyrox, gyroy, gyroz, accelx, accely, accelz, magx, magy, magz, contador, checksum \leftrightarrow . Las unidades son ‘grados/sg’ para el giróscopo, ‘mg’ para el acelerómetro, y ‘mGa’ para el magnetómetro. La aplicación refleja dos líneas **S**, es decir, dos paquetes de los sensores. Esto es debido a que, por defecto, la velocidad de muestreo de los sensores es el doble que para el quaternion.
- **B**, voltaje, contador, checksum \leftrightarrow . El voltaje tiene unidades de ‘mV’.

Los contadores para cada paquete van de ‘0’ a ‘255’. Esto permite identificar si ha habido pérdida de paquetes en la transmisión, y cuántos han sido.

Las *frecuencias de envío* de paquetes pueden ajustarse con los comandos (Commands en TERMINAL) “QR,#” (quaternion), “SR,#” (sensor), “BR,#” (batería), y, los valores que podemos asignar son los mostrados en la Tabla 1. *En esta práctica mantendremos los valores por defecto.*

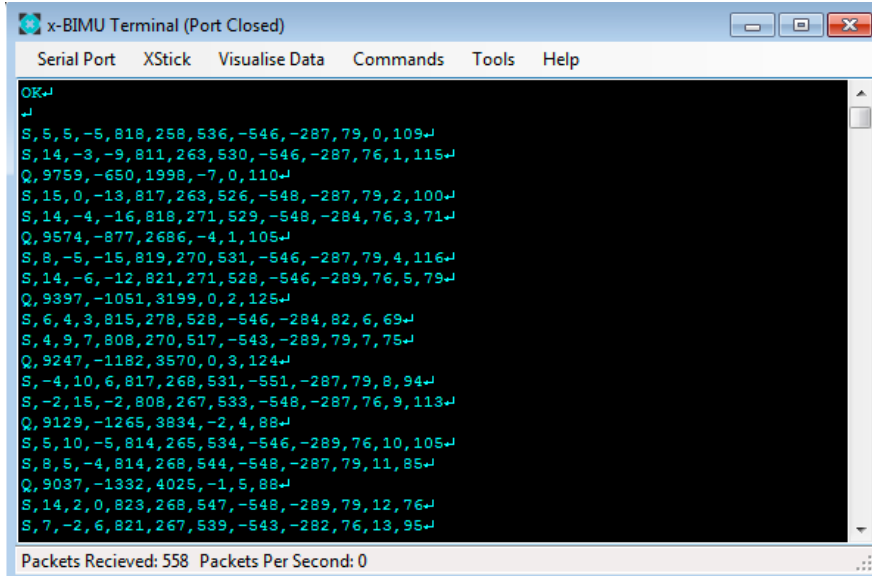


Figura 6: caption

0 = 0 Hz	1 = 1 Hz
2 = 2 Hz	3 = 4 Hz
4 = 8 Hz	5 = 16 Hz
6 = 32 Hz	7 = 64 Hz
8 = 128 Hz	9 = 256 Hz

Tabla 1: Frecuencias de muestreo posibles

Además, la parte inferior de la ventana de la aplicación TERMINAL muestra la cantidad de paquetes recibidos y...

La aplicación TERMINAL nos permite, además, visualizar los datos en tiempo real, y guardar los datos en archivos de ‘log’.

1. Para visualizar los datos, utilice el menú que despliega ‘Visualise Data’ de la ventana principal de TERMINAL. Los comandos que se despliegan permiten visualizar el movimiento de x-BIMU en el espacio 3D (Quaternion), los valores de giróscopo (Gyroscope), acelerómetro (Accelerometer), magnetómetro (Magnetometer), batería (Battery). Tal y como puede ver en la Figura 7, cada uno de estos comandos abre una ventana donde se muestra gráficamente los datos capturados en *casí* tiempo real. Los colores indican la componente representada: *rojo-X*, *verde-Y* y *azul-Z*. El alumno puede cambiar la escala de la gráfica con las flechas $\leftarrow \uparrow \downarrow \rightarrow$ de la ventana de la gráfica.

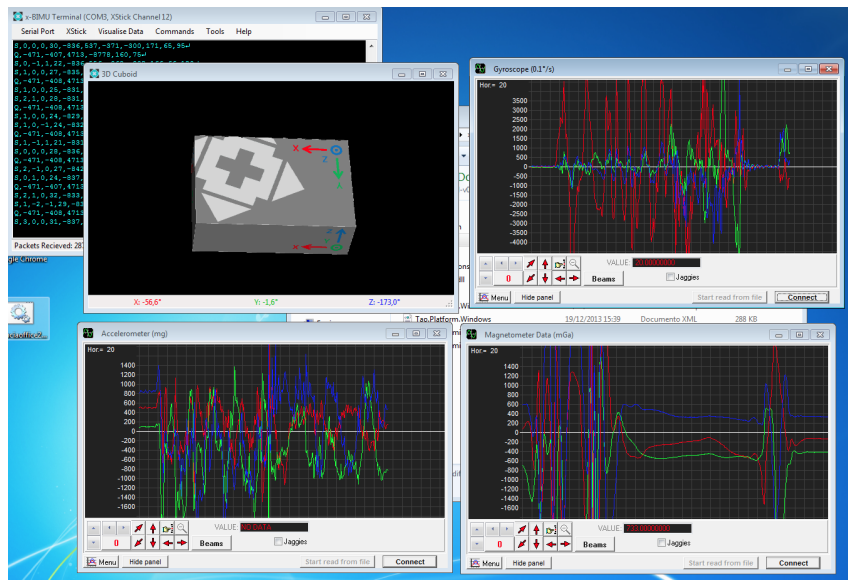


Figura 7: Visualización de datos desde TERMINAL

2. Para activar los archivos de ‘log’ terminal dispone del comando `Tools > Log To File`. Este comando permite seleccionar el directorio donde deseamos guardar los archivos de log. Por ejemplo, podemos seleccionar el directorio ‘LOGS’ (está en un .zip en el aula virtual). Los archivos de log guardan los datos de cada uno de los paquetes que aparecen en la ventana TERMINAL, en tres archivos:

- `LoggedData.csv_Battery.csv` para los paquetes de datos **B**.

- LoggedData.csv_Quaternion.csv para los paquetes de datos Q.
- LoggedData.csv_Sensor.csv para los paquetes de datos S.

siendo este último el de mayor interés para el desarrollo de esta práctica. Antes de abrir el archivo de log debemos parar el proceso. Esta acción ‘cierra’ el archivo y permite su manipulación por otro programa. La Figura 8 muestra un ejemplo del log de paquetes ‘S’.

	A	B	C	D	E	F	G	H	I	J	K
1	time (ms)	gyroscope x (0.1 deg/s)	gyroscope y (0.1 deg/s)	gyroscope z (0.1 deg/s)	accelerometer x (mg)	accelerometer y (mg)	accelerometer z (mg)	magnetometer x (mG)	magnetometer y (mG)	magnetometer z (mG)	packet counter
2	0	-1	0	0	-1	26	995	-87	-915	297	201
3	10	-1	0	0	2	25	990	-87	-912	294	202
4	10	0	0	0	-2	18	1009	-84	-912	294	203
5	10	0	0	0	-6	29	1002	-84	-915	292	204
6	32	0	0	-1	-2	21	1011	-84	-912	294	205
7	32	0	0	0	0	26	998	-87	-915	297	206
8	65	-1	1	-1	-5	23	1002	-87	-915	297	207
9	65	0	3	0	-9	23	1010	-84	-917	294	208
10	98	0	-1	1	-1	22	1001	-84	-912	300	209
11	98	0	0	1	-6	25	1002	-84	-915	292	210
12	131	1	0	0	-6	28	990	-87	-915	294	211
13	131	1	1	-1	-1	21	996	-87	-917	292	212
14	175	0	0	0	-4	24	1001	-84	-912	297	213
15	175	0	0	0	-5	24	992	-84	-912	294	214
16	208	0	0	1	-7	22	1010	-84	-912	294	215
17	208	-1	-1	0	-2	27	1003	-87	-915	294	216
18	219	1	0	0	1	25	1000	-84	-915	294	217
19	241	0	-1	1	-8	21	1000	-87	-912	294	218
20	273	0	1	0	-6	22	1000	-87	-912	297	219

Figura 8: Ejemplo de archivo LoggedData.csv_Sensor.csv

3 Tareas

1. Conectar x-BIMU en el PC (partición de Windows) y probar el software [x-BIMU TERMINAL](#) descrito en la Sección [2.2](#).
2. Descargar desde Aula Virtual el Jupyter Notebook “Test FFT.ipynb”, guardarlo en un directorio de trabajo de la Raspberry Pi, y probar los ejercicios incluidos.
3. Similarmente, realice los 2 entregables propuestos por el profesor y entréguelos en las fechas indicadas en para cada uno en el Aula Virtual.