

TIoT

TECNOLOGÍA PARA EL INTERNET DE LAS COSAS

PROYECTO

Maite GIL 4.627.958-2

maigipra@gmail.com

Leonardo Steinfeld, Julián Oreggioni

INSTITUTO DE INGENIERÍA ELÉCTRICA

FACULTAD DE INGENIERÍA

UNIVERSIDAD DE LA REPÚBLICA

July 30, 2024

1 Objetivos

El primer objetivo es estudiar y definir cuestiones de protocolo de comunicación y hardware a utilizar:

- “Protocolo de comunicación BLE” [?]
- “Hardware: MCU MSP430 F5529 + módulo BLE Proteus-III-SPI”, [?],

La elección del hardware se debe a que los resultados y análisis de este proyecto complementará al trabajo de tesis vinculado al estudiante.

Se va a crear una aplicación sencilla, con receptor en la PC, que permita enviar datos desde el dispositivo Periférico y puedan ser leídos desde la PC.

Se va a caracterizar el consumo del módulo BLE. Se va a utilizar la placa de Nordic Semiconductor Power Profiler Kit II. Para ello se va a obtener una medida preliminar de consumo. Se va a identificar y analizar los resultados para cada estado del módulo: bajo consumo, inactivo y conectado.

Se van a analizar parámetros (BLE y del sistema) que sean configurables para obtener más de un escenario de transmisión de datos y comparar el consumo entre los distintos casos.

Se va a buscar alcanzar el throughput indicado por el fabricante del módulo BLE.

Finalmente, se van a relevar medidas y generar un reporte con los resultados obtenidos.

2 Hardware

El hardware a utilizar es un microcontrolador de Texas Instruments MSP430 F5529, un módulo BLE de Wurth Elektronik Proteus-III-SPI. Las placas se comunican vía SPI de 4 líneas: CLK, MOSI, MISO y CS (chip select).

Se van a utilizar las placas de desarrollo de los chips mencionados. En primer lugar para el microcontrolador MSP430 se va a utilizar la placa MSP-EXP430F5529LP. Para el módulo BLE se va a utilizar la placa Family Evaluation Board que contiene el módulo Proteus-III-SPI. Del lado de la PC se va a utilizar un Dongle USB de la misma compañía Wurth Elektronik eiSos, que contiene un módulo BLE Proteus-III.

Para la caracterización de consumo se va a utilizar la placa de Nordic Semiconductor Power Profiler Kit II. Tiene dos modos de configuración: como amperímetro o como medidor de fuente. Se va a utilizar el modo medidor de fuente, por lo que se va a alimentar al módulo BLE Proteus desde la PPKII, y medir la corriente en tiempo real. Se configura este voltaje en 5V.

Se utiliza el analizador lógico de Saleae Logic Pro 8 para interceptar la comunicación SPI, la señal digital GPIO salida del MSP430 que se alterna durante las transiciones entre estados del módulo, y otras señales para debug.

La conexión entre el hardware mencionado se observa en la figura 1.

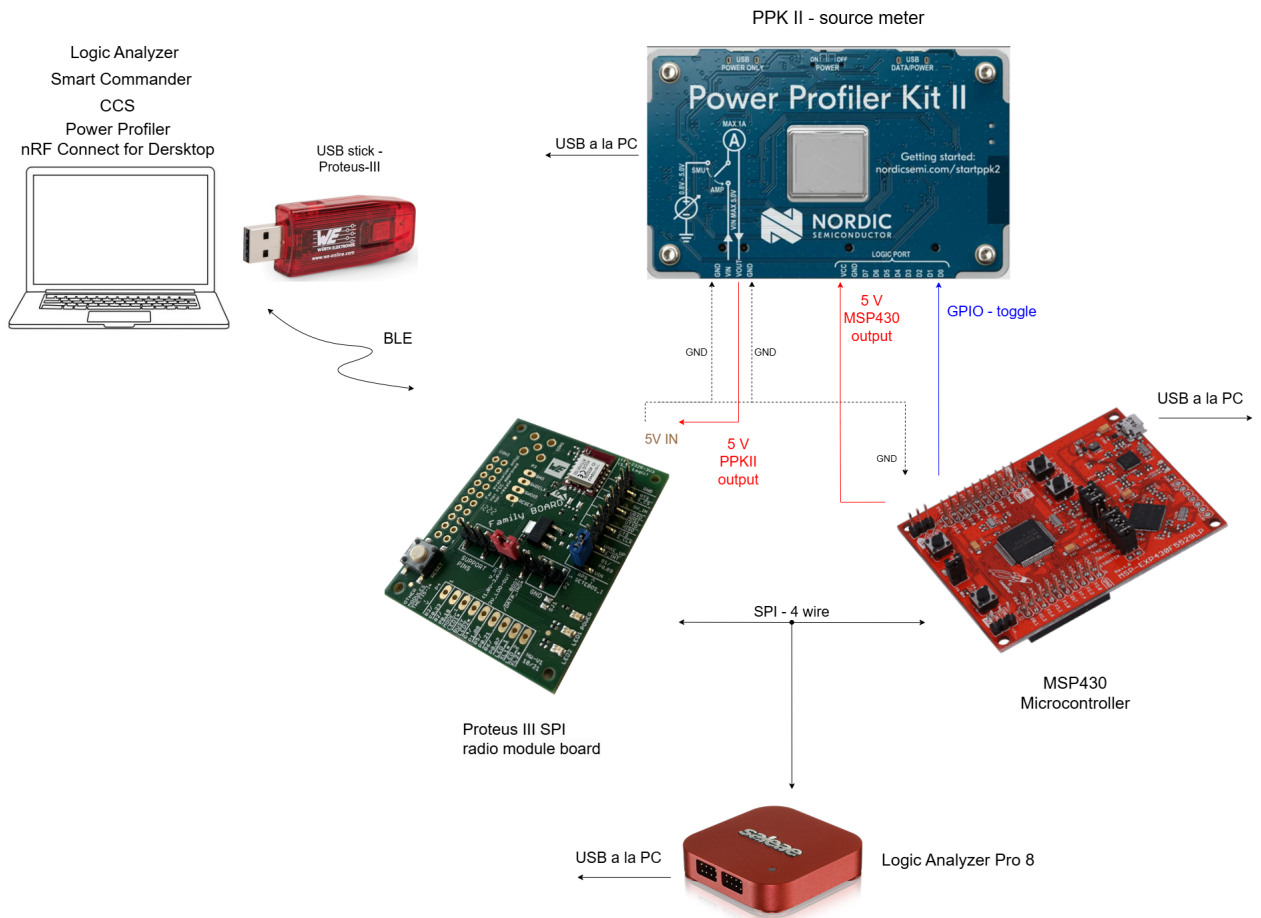


Figure 1: Diagrama de conexión de Hardware

Las conexiones del HW se pueden apreciar en la figura 2

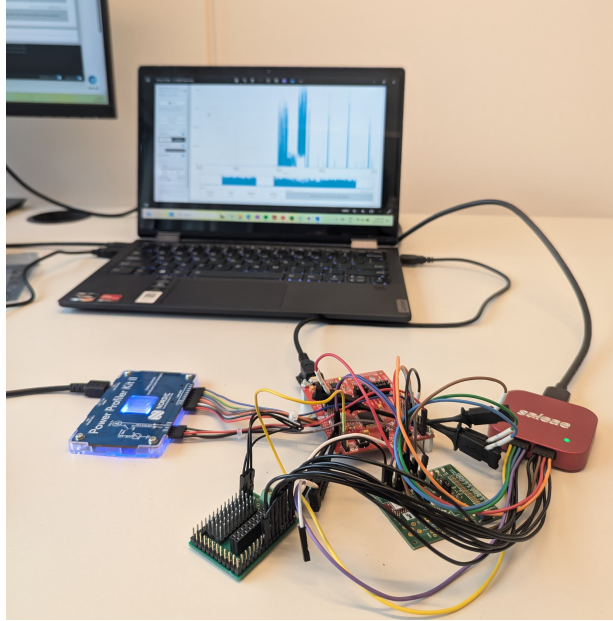


Figure 2: Foto de conexión de Hardware

3 Software

Para la comunicación, manejo y configuración del hardware mencionado se utilizan los siguientes software instalados en Windows.

- Code Composer Studio v.12.1.0, de Texas Instruments: para programar y debug el microcontrolador MSP430.
- Smart Commander v1.3.3.0, de Wurth Elektronik: para configurar el módulo BLE del USB Stick, y para la transmisión y recepción de datos de la conexión establecida entre los módulos BLE. [3]
- Power Profiler, de Nordic Semiconductor: para la configuración y muestreo de la corriente eléctrica del sistema conectado a la PPKII. Este software corre dentro de nRF Connect for Desktop v5.0.1. [1]
- Logic 2 v2.4.14, de Saleae: para configurar y visualizar las entradas digital y analógicas del analizador lógico Logic Pro 8.[2]

4 Funcionamiento del sistema

Se presenta el diagrama de bloques del sistema para comprender el funcionamiento del sistema, y para presentar los términos que se van a utilizar. Se puede ver en la figura 3. Por un lado, tenemos el dispositivo Central conformado por la PC con los software previamente mencionados, y la conexión USB al Dongle con la Proteus-III. Este dispositivo va a tener intercambio de datos mediante protocolo BLE con el dispositivo periférico: módulo radio BLE Proteus-III-SPI + el microcontrolador MSP430. En líneas punteadas y colores con menor intensidad se presenta el resto de la configuración del sistema que comprende el proyecto de tesis vinculada al estudiante.

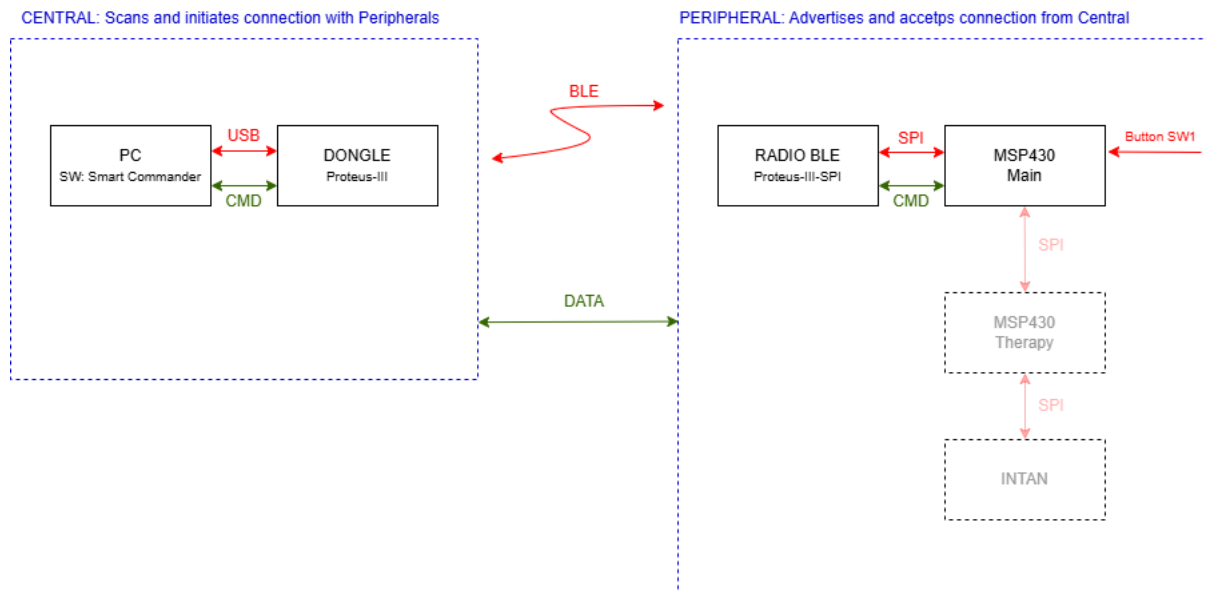


Figure 3: Diagrama de bloques del sistema

El Central incluye una PC conectada a la corriente de manera constante. Por otro lado, el Periférico es el dispositivo que va a utilizar batería, por lo que se busca tener el menor consumo posible. Esto implica tener el módulo BLE apagado (o bajo consumo), hasta que se presione el botón SW1 del MSP430 para encender y permanecer a la escucha de solicitudes de conexión. Del lado del dispositivo Central, el USB Stick va a escanear los módulos que estén en rango y obtener la dirección BTMAC de cada módulo, y con esta información se puede enviar una solicitud de conexión al módulo BLE de interés.

El periférico se va a encontrar en un estado de bajo consumo. Al presionar el botón SW1, el módulo BLE del Periférico se enciende y se va a un estado inactivo (o IDLE), en el que envía mensajes de Advertising para ser descubierto por otros módulos. En este caso queremos que sea descubierto por el Central. El Central va a descubrir al Periférico durante el escaneo de módulos, y le va a enviar la solicitud de conexión. En este momento existe una transmisión de mensajes entre los módulos BLE que forman parte de la negociación de la conexión. Aquí se definen parámetros y configuraciones que serán de gran relevancia para la velocidad de transmisión de los datos, la seguridad de la conexión y del consumo del sistema. Algunos de ellos se van a mencionar más adelante y son:

- Intervalo de conexión.
- Advertising Timeout.
- Intervalo de Advertising.
- Ventana de Escaneo.
- Intervalo de Escaneo.

Los intervalos de conexión, son intervalos de tiempo en los que se habilita al módulo BLE a transmitir datos. Es decir, una vez establecida la conexión entre los módulos, los datos no se transmiten en cualquier momento. Se debe esperar este tiempo “Intervalo de conexión” para que se de un evento de conexión y se pueda transmitir los datos del módulo. Esto es controlado por el protocolo BLE. El parámetro “Advertising Timeout” es el tiempo durante el cual se envían mensajes de advertising para ser descubierto por otros módulos en rango. “Intervalo de Advertising” define la frecuencia con la que se transmiten los mensajes de Advertising, entre medio de cada transmisión de mensajes el módulo se va a un modo suspendido. Los últimos dos parámetros mencionados “Ventana de Escaneo” y “Intervalo de Escaneo” son parámetros relacionados al modo escaneo (o modo scanning) que realiza, en este caso, el dispositivo Central para buscar módulos en rangos.

El Periférico va a transicionar entre 3 estados:

- IDLE (o inactivo)
- SLEEP (o bajo consumo)
- CONNECTED (o conectado)

La transición entre estados la maneja el Host del módulo Proteus-III-SPI, es decir el microcontrolador MSP430. Como se mencionó el módulo BLE y la MSP430 se comunican vía SPI. El Host configura y se comunica con el módulo BLE mediante comandos HEXA ya definidos en el manual. Existen comandos de solicitud = REQ, comandos de acknowledge = CNF, y comandos que indican el resultado de la solicitud = IND. A modo de ejemplo, la secuencia de comandos en la solicitud de conexión se presentan en la tabla 1.

Nombre Comando	Dirección CMD	Función	Señal de inicio	CMD	Largo	Payload	Check sum
CMD.CONNECT_REQ	Host => Proteus	Solicitud de conexión	0x02	0x06	0x06 0x00	6 Bytes: BTMAC	1 Byte
CMD.CONNECT_CNF	Host <= Proteus	ACK a la solicitud de conexión	0x02	0x46	0x01 0x00	1 Byte: Status	1 Byte
CMD.CONNECT_IND	Host <= Proteus	Indica el status de la conexión	0x02	0x86	0x07 0x00	1 Byte: Status 6 Bytes: BTMAC	1 Byte
CMD.CHANNELOPEN_RSP	Host <= Proteus	Avisa que el canal se abrió exitosamente	0x02	0xC6	0x08 0x00	1 Byte: Status 6 Bytes: BTMAC 1 Byte: Max payload	1 Byte

Table 1: Secuencia de comandos en la solicitud de conexión.

El diagrama de estados del Periférico se puede apreciar en la imagen 4. Se puede identificar la transición entre estados, en azul se muestran los comandos o eventos que permiten efectivamente la transición. En verde se muestra lo que dispara el envío de comandos. Estos son 2 Timers configurados desde la MSP430, con tiempos distintos y configurables que permiten al sistema: irse al estado SLEEP luego de 8 segundos desde el estado IDLE, e irse al estado IDLE desde el estado CONNECTED luego de 6 segundos. Estos Timers permiten regular los tiempos de permanencia en los estados que tiene un consumo alto.

El funcionamiento del Periférico es el siguiente. Comienza en $t=0$ en el estado IDLE. Permanece allí durante un tiempo T_{Idle} configurado en 8 segundos. Cuando dispara el timer, el Host envía el comando CMD_SLEEP_REQ y envía al módulo Proteus al estado SLEEP. Va a permanecer en este estado hasta que se presione el botón SW1 de la MSP430. Una vez presionado SW1 se levanta al módulo Proteus con una señal WAKE_UP pin (salida del MSP430 y entrada a la Proteus). El módulo permanece en estado IDLE durante T_{Idle} . En este estado el dispositivo está transmitiendo mensajes de Advertising para ser encontrado por el Central, y se queda esperando por una solicitud de conexión. Si recibe la solicitud de conexión previo al tiempo T_{Idle} , el módulo se va al estado CONNECTED. Permanece en este estado durante un tiempo $T_{connected}$. Aquí se queda esperando por datos de parte del Central. Estos datos representan comandos de solicitud de información específica. Si durante el tiempo $T_{connected}$, el Periférico recibe datos del Central, entonces se mantiene en el estado CONNECTED, recibe los datos, y pasa a transmitir datos hacia el Central.

Para poder verificar la transmisión y recepción de los datos se decide de manera arbitraria que el Central va a transmitir los datos **0xABCD**. El Periférico va a recibir datos y si coincide con este valor

predeterminado, va a enviar una secuencia de datos que va a corresponder a un **contador de 0 a 1000**, y **de 1000 a 0**. Esto genera un gran flujo de datos desde el Periférico al Central, y es sencillo de verificar la correcta recepción de los datos desde la PC, desde el software Smart Commander.

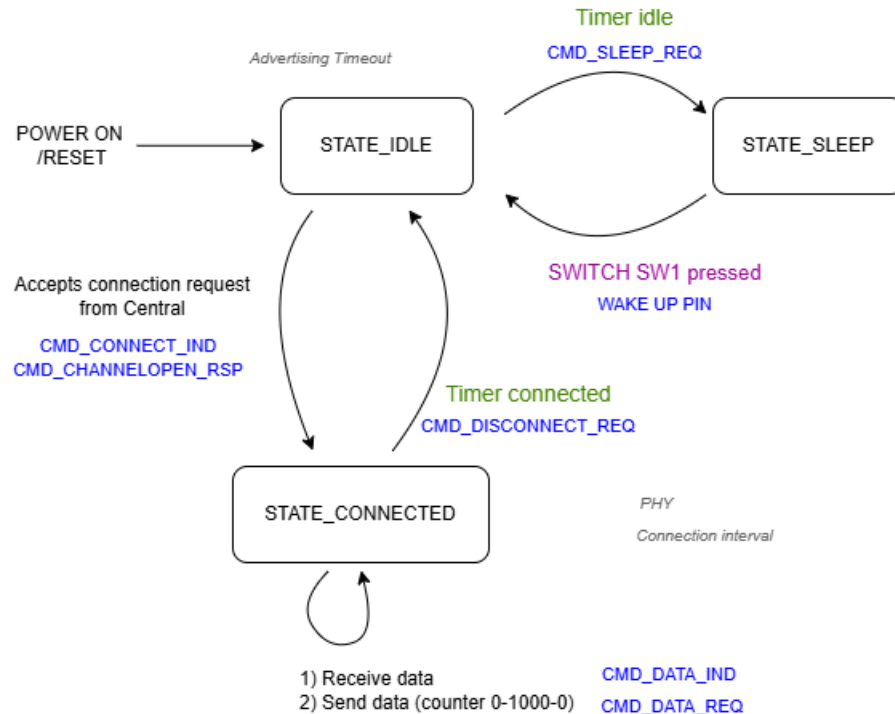


Figure 4: Diagrama de estados del Periférico

Para programar el contador 0-1000-0 se representa cada número decimal mediante 2 bytes. Esto se realiza así para poder representar todos los números, hasta el 1000, de la misma manera. Se podría a futuro utilizar Nibbles para mejorar la eficiencia de la transmisión de datos. Los datos que se se envían, en HEXA, van del: 0x00 0x00 al 0x03 0xE8, y luego al 0x00 0x00 nuevamente.

La Proteus-III cuenta con una limitación del Payload en cada comando de envío de datos. Se pueden enviar por vez 243 bytes. Esto significa que el contador se va a fragmentar en diversos paquetes de solicitud de datos, cada uno con 242 bytes, representando 121 valores decimales por paquete.

5 Caracterización de consumo

Para la caracterización de consumo, se define un Caso Test que representa la transición entre estados y permite evaluar el correcto uso de los Timers y los tiempos definidos. El Caso Test consiste en la secuencia SLEEP - IDLE - CONNECTED, segmentado. Es decir, en primer lugar se prueban las transiciones automáticas disparadas por los Timers al no recibir ninguna solicitud externa como el envío de solicitud de conexión desde el central, o la transmisión de datos desde el Central. A continuación se van agregando estos eventos externos que permiten la permanencia del módulo en los estados. Los tres esenarios del caso Test son:

- 1: en verde - SLEEP - IDLE-SLEEP
- 2: en azul - SLEEP - IDLE - CONNECTED(no action) - IDLE - SLEEP
- 3: en rojo - SLEEP - IDLE - CONNECTED(RX & TX data) - IDLE - SLEEP

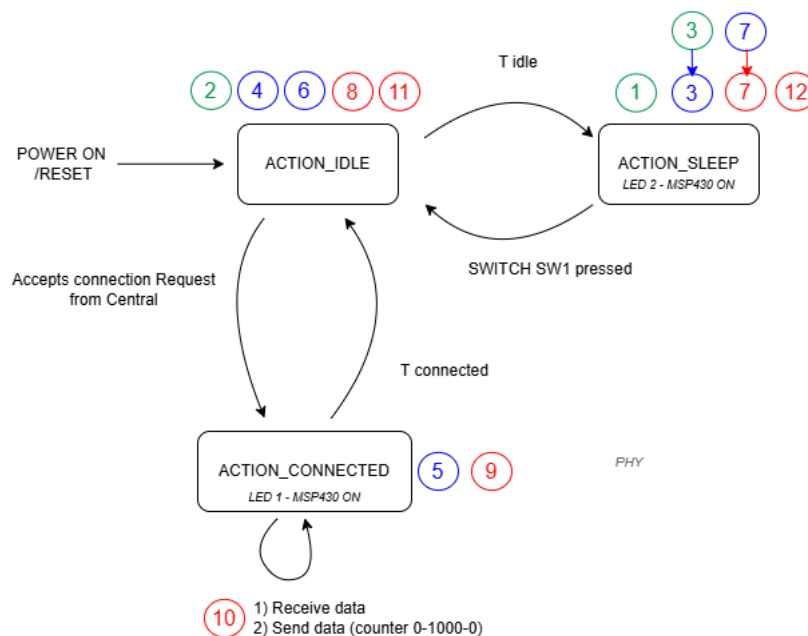


Figure 5: Diagrama de estados del Periférico - Caso test para la caracterización de consumo

Se utiliza el software Power Profiler para medir el consumo del módulo. Para identificar la transición entre estados, se utiliza una señal digital GPIO salida del MSP430, y entrada a la PPKII, la cual se alterna en cada transición de estado. De esta manera se puede obtener el consumo por cada estado.

Se obtiene el consumo para cada estado. Esto se realiza de manera manual, haciendo Zoom en la gráfica obtenida en el Power Profiler, hasta obtener la muestra correspondiente que coincide con el cambio de valor de la señal digital GPIO, y seleccionando la ventana de tiempo que permanece en el mismo estado. Del software se obtiene la corriente promedio [mA], la corriente máxima [mA], el tiempo de la ventana seleccionada [s] y la carga [mC]. Para comprender el Caso Test, se grafica la corriente promedio de cada estado a lo largo del tiempo total. Esto se puede apreciar en la figura 6.

Se puede observar de la gráfica anterior la diferencia de consumo por estados. Para ello, resulta interesante comparar este y otros parámetros entre estados. Se grafica la corriente promedio, la corriente la máxima, el tiempo de permanencia en cada estado evaluado en las medidas, y la carga. Se presentan estos resultados en la figura 7.

Si bien el módulo transiciona entre 3 estados: SLEEP, IDLE y CONNECTED, se diferencian los momentos de recepción y de transmisión de datos dentro del estado CONNECTED, para tener un análisis

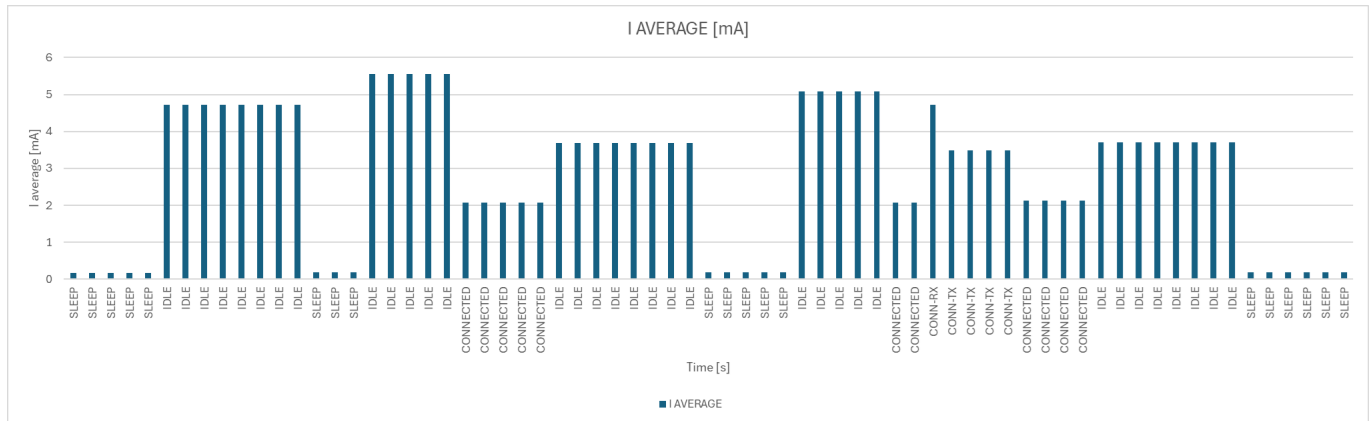
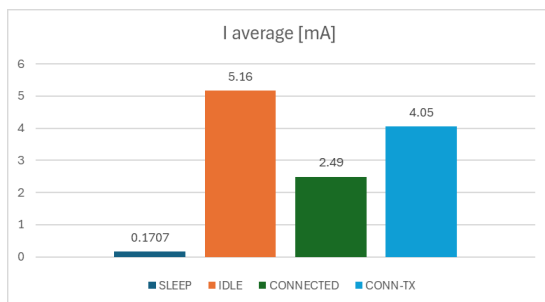
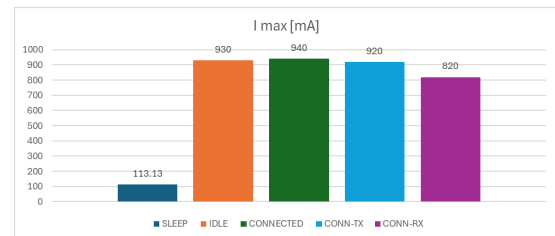


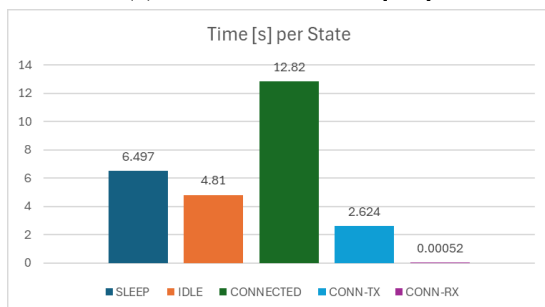
Figure 6: Gráfica de la corriente promedio durante cada estado en el Caso Test



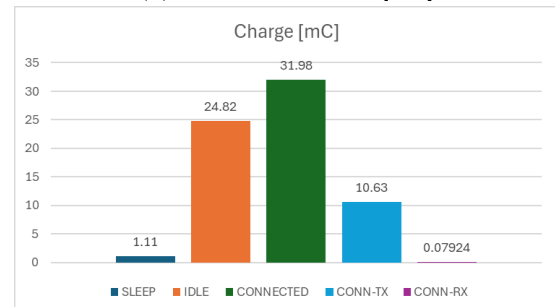
(a) Corriente promedio [mA]



(b) Corriente máxima [mA]



(c) Tiempo de permanencia en cada estado [s]



(d) Carga [mC]

Figure 7: Gráficas de los valores obtenidos por Power Profiler, para cada estado del Caso Test

más enriquecedor. A estas divisiones se les llama CONN-TX y CONN-RX para la transmisión y recepción respectivamente.

En la gráfica 7a se observa la corriente promedio en [mA] para los distintos estados. Se quita CONN-RX por no brindar relevancia en esta medida, y al ocupar un rango de tiempo muy pequeño, la medida resultante es considerablemente mayor y se pierde foco en las medidas de los demás estados. Se recuerda que la recepción de datos es únicamente 0xABCD del Central al Periférico, lo que equivale a aproximadamente 500 us.

De modo general, se puede distinguir la diferencia de consumo existente entre el estado SLEEP y los restantes estados. La corriente máxima es considerablemente menor para el estado de bajo consumo, esto también se observa en su corriente promedio y en la carga, aún cuando el tiempo de permanencia en SLEEP es comparable con los demás estados.

No se observa gran diferencia en la corriente máxima para los estados IDLE y CONNECTED, aunque

si se percibe que la corriente promedio es mayor en IDLE que en CONNECTED aún incluso en el momento de transmisión de datos CONN-TX.

Si se compara la gráfica de la carga con los demás valores obtenidos, se puede sacar conclusiones interesantes y mayor información del consumo del sistema. En líneas generales si se analizan las gráficas 7c y 7d: tiempo de permanencia vs carga se tiene que:

- Si el tiempo de permanencia es largo, y la carga es baja: puede indicar un estado eficiente de bajo consumo. Se puede corroborar que esto ocurre para el estado SLEEP.
- Si el tiempo de permanencia es corto, y la carga es alta: puede indicar que existen picos de consumo alto en períodos cortos. Se puede identificar al estado IDLE para este caso, y a su vez a la transmisión de datos: CONN-TX.

Si se analizan las gráficas 7a y 7d: corriente promedio y carga se tiene que:

- Si la corriente promedio es alta, y la carga también: indica estados de alto consumo. Aquí se identifican los estados IDLE y CONNECTED.
- Si la corriente promedio es baja, y la carga también es baja: indica estados de bajo consumo. Se observa al estado SLEEP en estas condiciones.

Si se analizan las gráficas 7b y 7d: corriente máxima y carga se tiene que:

- Si la corriente máxima es alta, y la carga acumulada también: puede indicar que los picos de corriente contribuyen significativamente al consumo total. Nuevamente, se identifica en este caso los estados IDLE y CONNECTED.
- Si la corriente máxima es alta, pero la carga acumulada no es proporcionalmente alta: puede indicar que los picos de corriente son breves y no contribuyen significativamente al consumo total. Aquí se puede identificar al estado CONN-RX, y se puede considerar también a CONN-TX.

Se realiza una verificación de la corriente promedio obtenido por *Power Profiler*. Se muestra en la figura 8 la gráfica e información que el software devuelve. Se ve una gráfica principal con la corriente en tiempo real. Abajo de esta, se muestran los valores de corriente promedio, corriente máxima, tiempo y carga para las muestras totales obtenidas, y en gris se observan los mismos parámetros para la ventana de tiempo seleccionada. Más abajo de los valores desplegados se observa la señal digital de entrada a la PPKII que se alterna en las transiciones de los estados. Se utiliza un único canal, canal '0'.

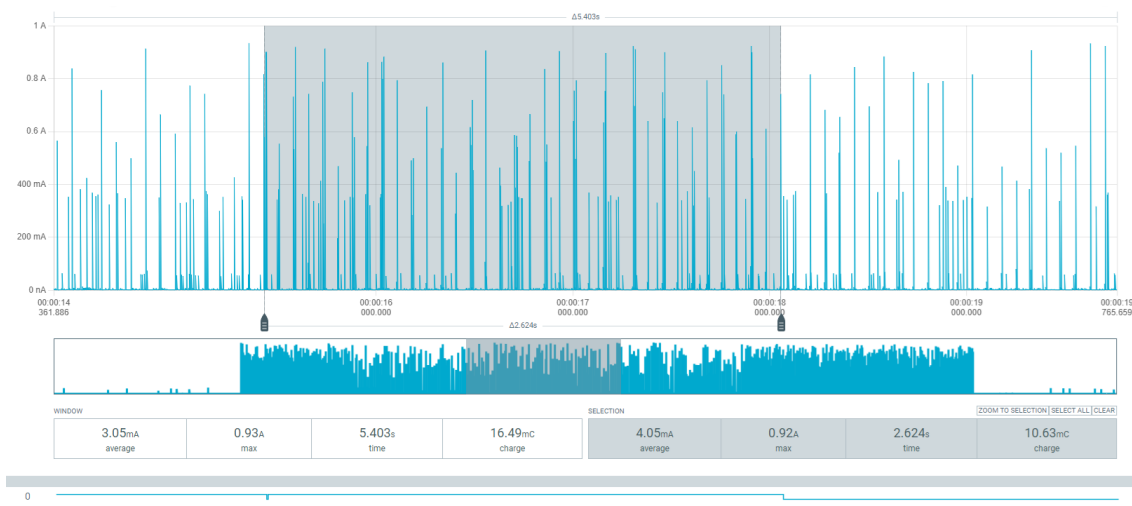


Figure 8: Software Power Profiler para la caracterización del consumo

Para verificar la corriente promedio, se toman las muestras del estado CONNECTED durante la transmisión, es decir: CONN-TX. Se observa en la gráfica que existen picos de consumos altos y luego un

consumo muy bajo. Se selecciona la ventana de tiempo de muestras correspondientes y se exportan los datos desde el software a un archivo .csv para operar con ellos en Excel. Con los valores de corriente para cada muestra, se extraen resultados como la máxima corriente $I_{MAX} = 923.36256mA$, corriente mínima $I_{MIN} = 0.673\mu A$, y la corriente promedio $I_{average} = 4.04946mA$, correspondiente a los 4.05 mA que se muestran en la gráfica 7a.

Para comprender mejor el consumo durante IDLE y CONNECTED, estados con mayor consumo, se pretende identificar lo que ocurre durante cada uno de ellos. En primer lugar, cuando el dispositivo se encuentra en IDLE, se transmiten mensajes de Advertising de manera periódica. Estos mensajes contienen su propia dirección MAC y tienen el fin de ser encontrados por otros módulos BLE. Existen parámetros configurables desde el módulo Proteus que podrían variar significativamente el consumo. Estos son:

- Advertising Timeout
- Advertising Interval

Estos parámetros se puede configurar mediante los registros RF_AdvertisingTimeout (index 0x07) y RF_ScanTiming respectivamente. Se modifican mediante el envío del comando CMD.SET_REQ, y se puede conocer cual es su configuración mediante el comando CMD.GET_REQ. Para profundizar este análisis se envían los comandos de solicitud de información para saber su configuración. Esto se hace al enviar CMD.GET_REQ con el índice 0x07 para el Timeout, y el índice 0x09 para el Intervalo.

Advertising Timeout es el tiempo en segundos en que cual envía mensajes de advertising en el estado IDLE. Luego de este tiempo, el advertising frena y el módulo se va a un modo suspendido. En este caso el módulo está configurado en 0 lo que significa que el módulo realiza advertising de manera indefinida.

Advertising Interval es la frecuencia con la que se envían los mensajes de Advertising, entre medio de cada transmisión el modulo de va a un modo suspendido. By default el valor de la configuración es de 0x01 lo que equivale a un intervalo de Advertising de 40ms. En este mismo registro RF_ScanTiming, se puede a su vez modificar otros tiempos para el proceso de escaneo de módulos en rango como: la ventana de escaneo (tiempo en el cual se escaneo en el estado SCANNING, luego de este tiempo el modulo se va a un modo suspendido). Scan Interval. Dado que el Periférico no es quien escanea, se dejan estos parámetros sin profundizar.

En el estado CONNECTED, se envían a su vez mensajes entre los módulos BLE para permanecer conectados. Estos mensajes que se envían de manera periódica mientras la conexión está establecida, se puede modificar durante la negociación de conexión. Esta negociación ocurre al principio de la conexión. Cuando el dispositivo Central envía la solicitud de conexión hacia el Periférico, se envían a su vez mensajes donde se negocian los tiempos y seguridad de la conexión, a priori estos paquetes no visible para el usuario.

Existen dos parámetros configurables en la Proteus que podrían variar significativamente el consumo del estado CONNECTED. Estos son:

- Connection Interval
- Connection supervision Timeout

Cuando la conexión se ha establecido entre los módulos BLE, los datos se transmiten usando el comando CMD.DATA_REQ, los datos entran en un buffer del módulo y los envía en el próximo evento de intervalo de conexión. Los paquetes de datos se envían 1 por cada intervalo de conexión. Este parámetro es modificable en el registro de configuración del Proteus RF_ConnectionTiming. Aquí se define un rango de conexión de intervalo, un mínimo y un máximo.

Enseguida los datos se han transmitido exitosamente desde el Periférico al Central, el Central envía un comando CMD.TXCOMPLETE_RSP para confirmar la transmisión. Este tiempo (hasta recibir el comando de correcta transmisión) coincide con el Intervalo de conexión que fue negociado durante el principio de la conexión. En el registro RF_Connection_Timing se define el mínimo y máximo intervalo de conexión posible. Para obtener el intervalo de conexión mas rápido posible, se debe configurar el registro RF_ConnectionTiming en 0, esto es 7.5ms. By default está configurado RF_ConnectionTiming

en 0x02: $[15 - 75]$ ms. Esto se averigua consultando mediante el comando `CMD_GET_REQ`, index 0x08.

Estos parámetros son de gran relevancia y van a ayudar a explicar mejor los resultados obtenidos en la siguiente sección.

6 Modificación capa física

Hasta el momento se caracterizó el consumo del sistema básico, se analizó el por qué del consumo en cada estado, y los parámetros que pueden influir significativamente en él. Para esta parte del proyecto se decide modificar la capa física del módulo BLE. Esto se configura mediante un comando de configuración `CMD_PHYUPDATE`, de la siguiente manera:

- index 0x01: 1MBit
- index 0x02: 2MBit
- index 0x04: LE coded

Se menciona brevemente lo que significan estos conceptos. En la parte inferior de Bluetooth LE se encuentra la capa física (PHY). PHY se refiere a las especificaciones de radio de la capa física que rigen el funcionamiento de la radio Bluetooth LE. Esta capa define diferentes esquemas de modulación y codificación adoptados por los transmisores de radio Bluetooth LE que afectan aspectos como el rendimiento de la radio. Esto, a su vez, cambia el consumo de batería del dispositivo o el alcance de la conexión. [4]

1M PHY o 1 Megabit PHY, es la PHY clásica compatible con todos los dispositivos Bluetooth LE. Como su nombre lo indica, utiliza 1 megabit por segundo. Al iniciar una conexión entre dos dispositivos Bluetooth LE, este es el modo que se utilizará para empezar. Luego, los dispositivos pueden solicitar otro modo si ambos dispositivos lo admiten. En este caso todo el intercambio de mensajes hasta solicitud de conexión se realiza en este modo. Luego, una vez establecida la conexión y mediante el envío de comando `CMD_PHYUPDATE` se modifica el modo de la radio.

2M PHY, o 2 Megabit PHY es un nuevo modo introducido en Bluetooth v5.0. Como su nombre lo indica, duplica la velocidad de datos a 2 Mbps. Dado que los datos se transmiten a una velocidad de datos más alta (más rápido), la radio necesita permanecer encendida durante menos tiempo, lo que reduce el uso de la batería. La desventaja es la disminución de la sensibilidad del receptor, lo que se traduce en un menor alcance de comunicación.

PHY codificado se introdujo para servir a aplicaciones en las que se busca lograr un mayor alcance de comunicación sacrificando la velocidad de datos. El PHY codificado utiliza esquemas de codificación para corregir errores de paquetes de manera más efectiva, lo que también significa que un solo bit está representado por más de 1 símbolo. El PHY codificado utiliza 2 modos, S=2 y S=8. En el modo S=2, 2 símbolos representan 1 bit, mientras que en el modo S=8, se utilizan 8 símbolos para representar un bit. La Proteus cuenta con un solo modo de codificación configurable S2.

Este modo en la Proteus está implementado mediante 1MBit PHY with DSSS and FEC. DSSS: Direct sequence spread spectrum, es una técnica para transmitir mensajes en la radio que generan redundancia en la información. Del lado del receptor se utiliza FEC: Forward Error Correction. Esta es una técnica que utiliza redundancia para corregir los paquetes recibidos con señal perturbada. La combinación de ambas técnicas habilita la transmisión de datos a largas distancias.

Se mide el consumo para los 2 casos nuevos modificando la capa física a través del envío de comandos. Se obtiene del Power Profiler: el tiempo de permanencia en cada estado, las corrientes promedios y máximas y la carga. Se comparan los valores obtenidos para los 3 casos y se grafica los resultados en la imagen 9.

Para completar este análisis se obtiene la medida de RSSI, se observa en la figura 9a. *RSSI* (received signal strength indicator), es decir: indicador de intensidad de la señal recibida. Mide la potencia y la calidad de la radiofrecuencia en el momento en que llega al receptor. Esta medida se indica en valores negativos, cuanto más cercano a cero es el valor, más fuerte y de mayor calidad es la señal. Este valor se puede obtener en el propio comando donde se encuentra los datos que se están recibiendo, este es `CMD_DATA_IND`. En este comando viene la dirección BTMAC del Central (dispositivo que transmite los datos), la intensidad de la señal recibida: RSSI y a continuación los datos que se reciben: 0xABCD.

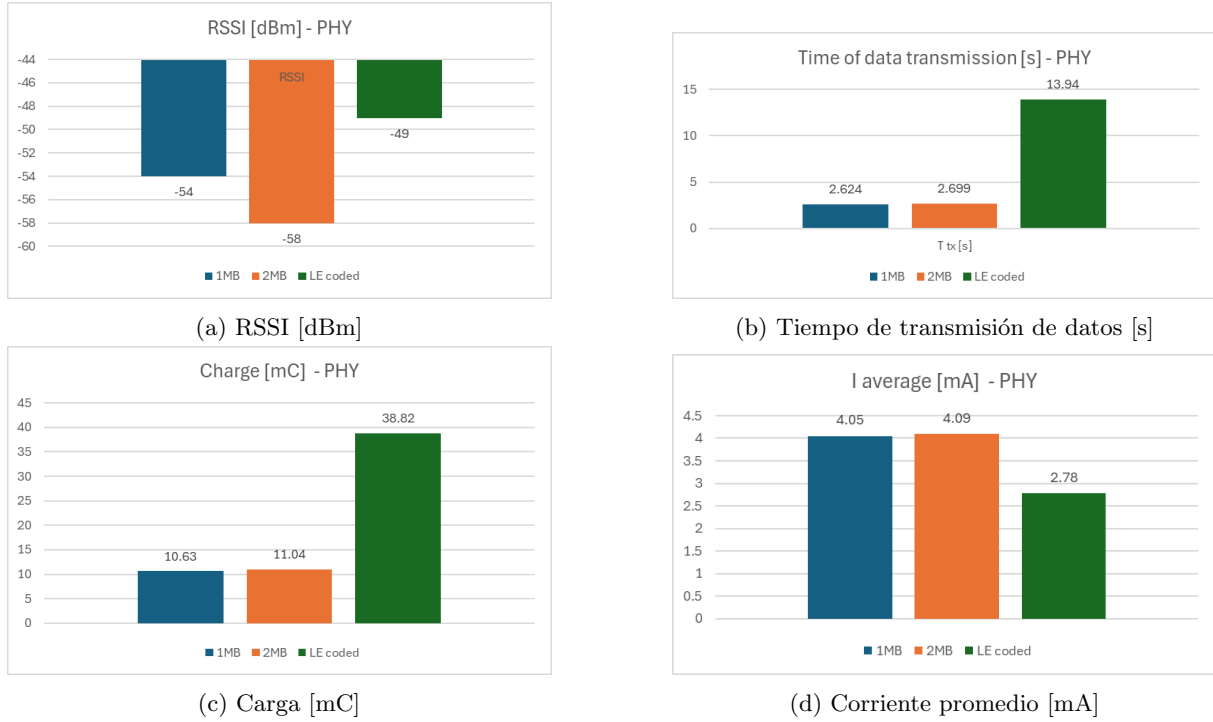


Figure 9: Gráficas comparando los 3 modos de radio: PHY 1MBit, 2MBit, LE-Coded

En primer lugar se compara los valores de RSSI. Para el caso PHY 2MBit se espera que el consumo sea menor (al transmitir los datos más rápidos, la radio estaría menos tiempo prendida) y sacrifica rango de transmisión, esto último se traduce a un RSSI menor que para el caso de 1MBit. Mientras que para PHY codificado ocurre lo opuesto, este modo promete transmisión de datos en mayor rango a costa de mayor consumo, esto se traduce en un mejor o mayor RSSI. Estos resultados se pueden ver en la gráfica 9a.

En segundo lugar se grafica el tiempo de permanencia en el estado CONN-TX, es decir durante la transmisión de datos. Los resultados indican tiempos similares para los casos PHY 1MBit y 2MBit, y un tiempo significativamente mayor para PHY Codificado. Esto se condice con lo esperado, ya que en este modo se transmite un número mayor de símbolos por bit de información.

Con respecto al tiempo de permanencia entre los dos primeros casos 1MBit y 2MBit se concluye que el tiempo es prácticamente el mismo cuando deberíamos ver un menor tiempo para 2MBit debido a que lo que se aceleró es el *tiempo de conexión*, y no se modificó el *intervalo de conexión*. Este intervalo se podría modificar mediante el comando CMD_SET_REQ como se explicó en la sección anterior. Al permanecer el mismo intervalo de conexión para los dos casos, no se puede apreciar la mejora en la transmisión de datos para el caso 2MBit. Recordemos que Proteus envía un paquete de datos por intervalo de conexión.

Se incluyen las gráficas 9c y 9d: carga y corriente promedio para cada caso. Se observa que la corriente promedio es similar para los casos 1MBit y 2MBit, mientras que es menor para el caso de PHY codificado. Sin embargo, la carga para el último caso PHY codificado es considerablemente mayor que para los otros 2 casos, esto se debe a que el tiempo de permanencia en el estado CONN-TX es mayor.

Se considera de relevancia poder analizar el Throughput alcanzado para los 3 casos y compararlos con los valores brindados por el fabricante. **Throughput** es la tasa de transferencia efectiva del sistema. Los nombres de los modos de radio determinan la velocidad de transmisión de paquetes ideal. Sin embargo, a la hora de implementar la comunicación existen diversos parámetros que influyen significativamente en esta medida.

Para calcular el Throughput del sistema se utiliza el número de bytes que se transmiten por paquete-

tes: 242 bytes por paquete; y el tiempo de transmisión de los datos. Este tiempo se obtiene desde el software del analizador lógico donde se observan los comandos SPI capturados. El tiempo se define en esta aplicación desde que se copian los datos a enviar en el buffer del módulo BLE (al finalizar la transmisión del comando CMD_DATA_REQ) hasta que se recibe el comando de que la transmisión fue exitosa (CMD_TXCOMPLETE_RSP). A priori no se puede identificar el tiempo en donde comienzan los intervalos de conexión. En la tabla 2 se muestran los valores calculados, y los valores del manual del fabricante.

Radio mode	Bytes TX [kB]	T [ms]	Throughput [kB/s] - calculado	Throughput [kB/s] - del fabricante
1MBit	0.242	139.51	1.734	25.7
2MBit	0.242	139.50	1.735	64.3
LE coded	0.242	814.51	0.297	3.14

Table 2: Cálculo del Throughput en [kB/s] para los 3 modos de radio: PHY 1MBit, 2MBit, LE-Coded

Se puede notar que los valores obtenidos para este proyecto son considerablemente menores. Pero es necesario destacar las distintas condiciones en las que se toman las medidas. En primer lugar, el fabricante utiliza la máxima velocidad de SPI posible 8MHz, mientras que en este proyecto se utilizó el reloj SMCLK de la MSP430 de 1MHz. A su vez, utiliza un modo de la Proteus "High throughput mode" que permite transmitir por payload hasta 924 bytes (4 tramas de radio, cada una con payload de 241 bytes cada uno), mientras que en este proyecto se utiliza el modo estándar con un payload máximo de 243 bytes. También, el fabricante configura el intervalo de conexión con el mínimo valor posible 7.5ms, mientras que en esta aplicación se deja la configuración por defecto con un intervalo de conexión de 15ms a 75ms. Por todas estas razones se puede decir que no son comparables las medidas, por haberse realizado en condiciones significativamente distintas. Sería conveniente hacer las configuraciones necesarias para acercarse al escenario propuesto por el fabricante y así comparar los valores calculados.

7 Casos de uso

Este proyecto pretende sumar un análisis adicional al proyecto de tesis vinculado al estudiante. Se puede suponer que para la aplicación final se precisa que se envíen datos 1 vez por día. Es decir, el escenario de uso sería: el módulo ble del Periférico permanece apagado durante la mayor parte del día, o en modo de bajo consumo, y se prende una única vez para enviar los datos, y luego volver a a bajo consumo. Con este caso de uso, resulta de baja relevancia las configuraciones de parámetros que se mencionaron para hacer la comunicación más eficiente, ya que es un intervalo de tiempo prácticamente insignificante en el consumo total de 24 horas. Si por otro lado, se considera un caso de uso donde se requiere transmitir estos datos una vez por hora durante las 24 horas, se obtiene sin duda un consumo mayor. Se analizan ambos resultados para observar si el consumo puede considerarse despreciable o considerable. Este análisis permite brindar noción del consumo necesario en distintos casos de uso.

Los parámetros que se consideran para este análisis son los valores obtenidos para el caso de PHY 1MBit del sistema total, sin diferenciar por estados. Los valores se despliegan en la tabla 3.

	$I_{average}$ [mA]	I_{max} [mA]	T_{total} [s]	Carga [mC]
1MBit	2.38	940	37.05	88.24

Table 3: Parámetros de Power Profiler para todo el sistema (sin diferenciar estados)

Para comparar los casos de uso se calculan y comparan: la carga total diaria; el consumo energético calculando la energía diaria (se supone un voltaje de alimentación 3.3V); y la corriente total promedio por día consumida. Estos valores obtenidos se muestran en la tabla 4.

Caso de uso	Carga diaria Q [mC]	Energía diaria E [mJ]	Corriente diaria $I_{average}$ [mA]
1 vez por hora [24 hs]	2118	6989	24.51
1 vez al día [1 h]	88	290	1.02

Table 4: Parámetros de consumo según distintos casos de uso

Para analizar estos resultados con mayor claridad, se elige de manera arbitraria los parámetros de una batería de 100 mAh, y 3.3V. Para calcular la capacidad de la batería en mWh, se calcula la capacidad en mAh \times Voltaje. Para una batería de 1000 mAh a 3.3V: $100mAh * 3.3V = 330mWh$. Para obtener la duración de la batería para cada caso se debe conocer la capacidad de la batería y el consumo energético diario. La energía que se calculó en mJ se convierte a mWh:

- Caso de uso: 1 vez por hora.
 - Energía diaria: 6989 mJ
 - Consumo energético: 1.9414 mWh
- Caso de uso: 1 vez al día.
 - Energía diaria: 290 mJ
 - Consumo energético: 0.0806 mWh

Para obtener la duración de la batería:

$$Duracion_bateria = \frac{capacidad_bateria[mWh]}{Energia_diaria[mWh/dia]} \quad (1)$$

Para el caso de uso 1 vez por hora:

$$Duracion_bateria_24h = 330mV1.9414mW/dia \approx 170dias \quad (2)$$

Para el caso de uso 1 vez al día:

$$Duracion_bateria_1h = 330mV0.0806mW/dia \approx 4095dias \quad (3)$$

Este análisis tiene la intención únicamente de brindar una idea clara de cómo la frecuencia de operación en la transmisión de datos puede afectar la duración de la batería. Este dato del consumo es simplemente una parte del consumo que tendrá el dispositivo Periférico, por lo que se debería obtener un consumo más completo del sistema para poder determinar la frecuencia de transmisión de datos. Con el flujo de datos elegido para este proyecto se puede decir que si bien será más conveniente transmitir una única vez en el día, no es crítico, podría por otros motivos considerarse el envío de datos por hora, y no impactar significativamente el consumo del sistema.

References

- [1] nRF Connect <https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-desktop/Download#infotabs>
- [2] Logic 2, de Saleae <https://www.saleae.com/pages/downloads>
- [3] Smart Commander, de Wurth Elektronik <https://www.we-online.com/en/products/components/service/wireless-connectivity-sensors#i8117>
- [4] BLE Fundamentals <https://academy.nordicsemi.com/courses/bluetooth-low-energy-fundamentals/lessons/lesson-1-bluetooth-low-energy-introduction/topic/phy-radio-modes/>