

Revisiones	Fecha	Comentarios
0	13/02/08	
1	18/12/08	agrega broadcasts
2	01/09/08	diferenciación de otros módulos de la familia XBee

En este comentario técnico estudiaremos las posibles formas de intercomunicar aplicaciones en forma inalámbrica, mediante módulos XBee (o XBee-PRO) 802.15.4.

### Introducción

En los casos que veremos a continuación, cada aplicación es implementada sobre un sistema microprocesado, el cual nos permite disponer de un puerto de comunicaciones. Dado que el XBee 802.15.4 es un módulo de 3V, la interconexión entre estos sistemas y el XBee 802.15.4 deberá respetar dicha tensión. A los fines prácticos, asumimos que se ha conectado a los pines TD y RD la UART de un micro alimentado a 3V.

Si bien nos referimos a comunicación serie como diálogo entre aplicaciones, veremos en otros comentarios técnicos que también es posible transferir el estado lógico de pines y magnitudes analógicas, por lo que si bien elegimos este caso para realizar una explicación de una forma que creemos más didáctica, no es éste un limitante.

Por defecto, los módulos tienen configurado un pin como  $\overline{CTS}$ , lo que significa que en el caso que intentemos transmitir más de lo que el módulo puede enviar al extremo remoto, nos lo indicará poniendo este pin inactivo. En sentido inverso, existe la posibilidad de indicar al XBee que suspenda su comunicación con el micro mediante el pin  $\overline{RTS}$ , el cual por defecto no ha sido configurado para dicha tarea, pero puede hacerse mediante el comando  $ATD6=1$ , dado que  $\overline{RTS}$  comparte el pin con DIO6.

Si bien los módulos soportan un esquema de direccionamiento de 64-bits, por simplicidad utilizaremos el método abreviado de 16-bits; esto requiere que se setee  $ATDH=0$ , lo cual coincide con la configuración por defecto.

No entraremos aquí en consideraciones sobre el canal de RF, el ID de la PAN; utilizamos los valores por defecto y asumimos que los módulos que nos interesan utilizan el mismo canal de radio y pertenecen a la misma PAN.

### Punto a punto

Este es el caso más simple posible, en el cual tenemos dos aplicaciones que requieren dialogar entre sí, y no es factible o deseable utilizar un cable.



Por defecto, el XBee 802.15.4 funciona en modo transparente. En este modo, el módulo envía al remoto configurado como destinatario los mensajes que recibe por su puerto serie, y presenta en éste los mensajes que recibe del módulo remoto. Si sólo tenemos dos módulos, es posible dejar ambos con su configuración de fábrica, en la cual su dirección y el destinatario están en cero, con lo cual queda automáticamente establecida una conexión entre los puertos serie de ambos módulos. Si tenemos más de un módulo dentro de la misma área de cobertura, deberemos configurar las direcciones utilizando el comando  $ATMY=<dirección>$  para la

## CTC-054, Utilización de XBee 802.15.4 para comunicación de aplicaciones

dirección propia y ATDL=<dirección> para la dirección del remoto. Esto lo haremos en ambos módulos, por ejemplo:

módulo 1

ATMY=1234

ATDL=4321

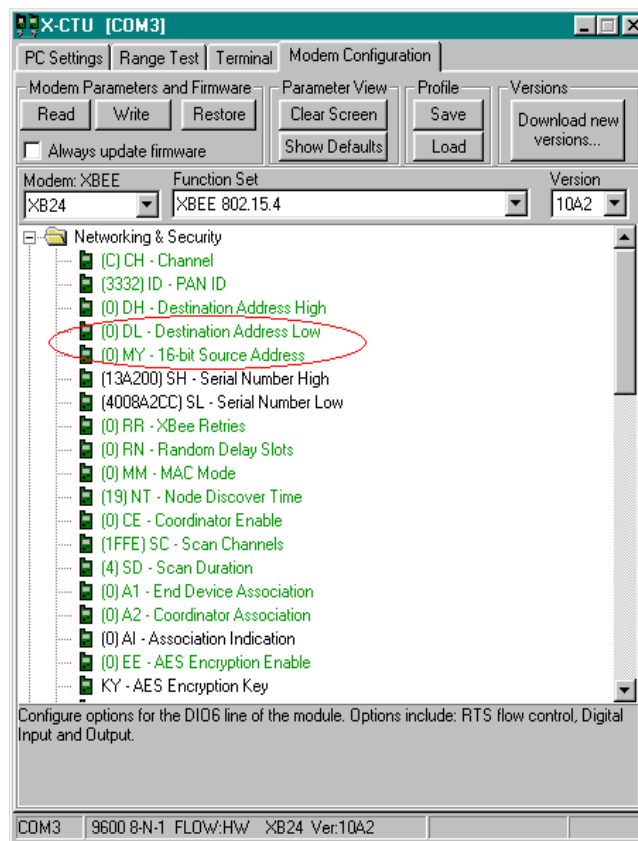
módulo 2

ATMY=4321

ATDL=1234

Dado que cada módulo no sabe en qué momento el otro va a transmitirle algo, es necesario que ambos estén siempre atentos a lo que el otro pueda decirle, lo cual requiere que ninguno de los dos 'duerma', es decir, entre en modo bajo consumo. Esto es así por defecto, por lo que no debemos modificar nada.

El ingreso de comandos AT lo podemos hacer mediante el programa X-CTU antes de poner los módulos en servicio.

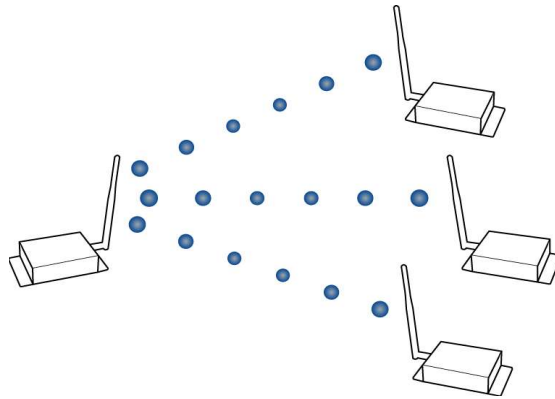


### Punto a multipunto

#### Caso 1: recolección de datos remotos

Este es el caso más simple de los dos, en el cual tenemos una serie de aplicaciones remotas que reportan información a un punto central, sin que haya comunicación en el sentido inverso.

## CTC-054, Utilización de XBee 802.15.4 para comunicación de aplicaciones



Cada módulo remoto tendrá como dirección de destino la del módulo central, y su dirección propia será única. En el módulo central no configuramos dirección de destino, dado que éste no transmite información a ninguno de los remotos. De este modo, todo lo que entra por la UART de cada uno de los remotos, sale por la UART del módulo principal.

módulo 1

ATMY=1234

ATDL=0

módulo 2

ATMY=1235

ATDL=0

módulo 3

ATMY=1236

ATDL=0

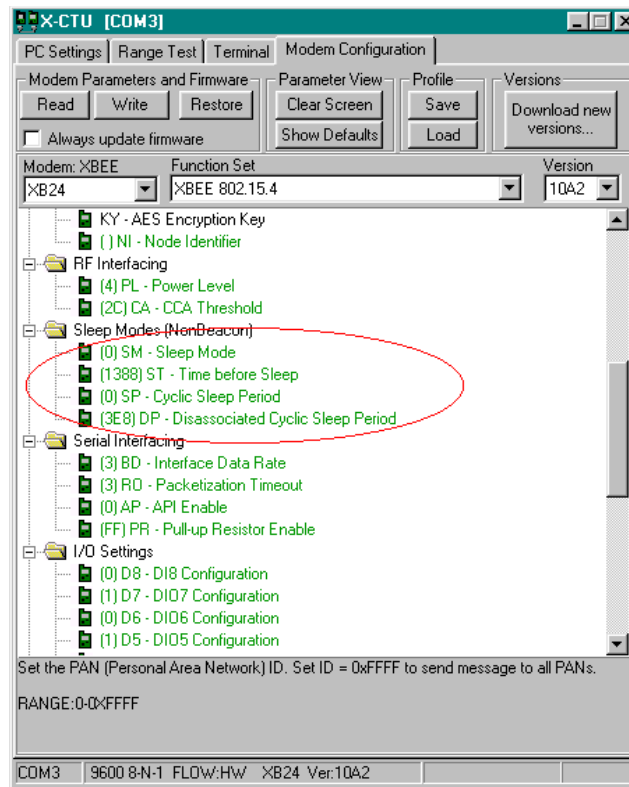
módulo central

ATMY=0

El ingreso de comandos AT lo podemos hacer también mediante el programa X-CTU, antes de poner los módulos en servicio.

En este caso, dado que los remotos reportan periódicamente y no reciben nada, cada uno de ellos puede ponerse a dormir por un tiempo arbitrario, dependiente de la aplicación, mientras no tiene nada que reportar. Esto se configura mediante el comando ATSM y demás asociados, como puede apreciarse en X-CTU:

## CTC-054, Utilización de XBee 802.15.4 para comunicación de aplicaciones



Dado que es la aplicación la que controla el envío de mensajes, tal vez lo más indicado sea configurar ATSM=1 (pin hibernate) o ATSM=2 (pin doze), con lo cual el módulo duerme hasta que la aplicación (el micro conectado) lo despierta activando el pin *DTR*. El modo a elegir dependerá del tiempo disponible para esperar a que el módulo despierte (13,2ms y 2ms, respectivamente), y los requerimientos de consumo (10uA y 50uA, respectivamente).

En el sistema conectado al módulo central, recibiremos los mensajes que nos envíen los remotos. Para determinar quién es el que nos envía el mensaje, deberemos insertar algún tipo de identificador dentro del mismo mensaje, o utilizar el modo API, tema que dejaremos para otra oportunidad.

A fin de mantener la integridad de un mensaje, todos los bytes que lo conforman deberán ser enviados uno a continuación del otro, sin una demora mayor a lo que indica el comando ATRO y con una longitud de mensaje menor a 100 bytes.

### Caso 2: comunicación en ambos sentidos

En este caso, el sistema central debe enviar información a los remotos. Como pudimos observar, la información se envía a aquel remoto cuya dirección coincide con lo que colocamos en ATDL; para direccionar diferentes módulos, deberemos alterar periódicamente este parámetro (cada vez que deseemos transmitir a un remoto diferente).

Dependiendo de la frecuencia de transmisiones, esto puede llegar a resultar molesto o incluso tedioso, dado que los cambios de configuración se realizan escapando a modo comando, lo cual consiste en dejar un tiempo de guarda, ingresar una secuencia de escape, y esperar otro tiempo de guarda y luego la respuesta 'OK' del módulo. En un caso como éste, suele ser preferible utilizar el modo API, que no desarrollaremos aquí.

El siguiente es un ejemplo de una comunicación como la descrita, vista desde el sistema conectado al módulo central. Para mayor claridad, hemos resaltado *lo que envía este sistema y las respuestas del módulo*:

```
soy 1234, hace frio
+++
OK
ATDL=1234
```

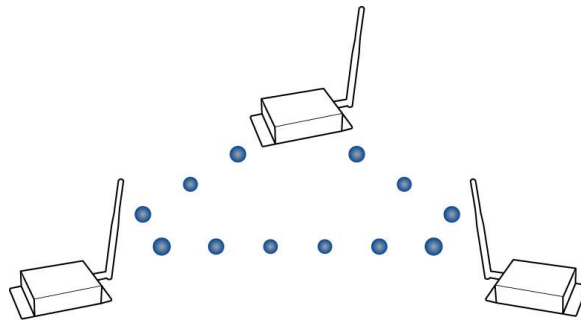
```
ATCN
OK
prende el calefactor
soy 1235, hace calor
+++
OK
ATDL=1235
ATCN
OK
apaga el calefactor
```

Nótese que tanto antes como después de enviar la secuencia de escape, debe dejarse transcurrir un tiempo de guarda que se configura mediante ATGT, y que por defecto es de un segundo.

En este caso, al haber comunicación en ambos sentidos, los remotos no saben en qué momento el central va a transmitirles algo; esto nos impide utilizar el modo bajo consumo, a menos que hagamos algunas restricciones. Si el central puede esperar a que el remoto en cuestión le hable primero, entonces envía el mensaje que tenga en ese momento; esto permite nuevamente que los remotos puedan entrar en modo bajo consumo hasta tanto tengan algo que transmitir y allí recibir lo que haya para ellos. Si bien en este caso no es la intención, es posible configurar los módulos con el central como coordinador, en cuyo caso éste almacena hasta dos mensajes en espera de detectar que los remotos están despiertos y poder transmitirlos. La configuración de una red multipunto con coordinador es algo más complicada, y no la abordaremos aquí.

### Peer to peer

Lo que tenemos aquí es una determinada cantidad de aplicaciones (más de dos) que deben dialogar entre sí. Desde el punto de vista de la configuración de los módulos, cada uno puede considerarse como un módulo central en el caso punto a multipunto, dado que recibe mensajes de cualquiera de los remotos y envía también a cualquiera de ellos. Como comentáramos en dicho caso, es posible cambiar el valor de ATDL antes de enviar cada mensaje, aunque dependiendo de la frecuencia de envío de mensajes y la aplicación en sí, puede ser preferible utilizar el modo API.



Una vez más, dado que cada módulo no sabe en qué momento el otro va a transmitirle algo, es necesario que todos estén siempre atentos a lo que otro pueda decirles, lo cual requiere que ninguno entre en modo bajo consumo. Esto es así por defecto, por lo que no debemos modificar nada.

### Broadcasts

Una alternativa cuando la misma información es útil para muchos destinatarios, es realizar un broadcast. Configurando ATDL=FFFF realiza un broadcast, es decir, envía un mensaje que puede ser recibido por todos los miembros de una PAN.