

Revisiones	Fecha	Comentarios
0	03/09/08	portado de CTC-054

En este comentario técnico estudiaremos las posibles formas de intercomunicar aplicaciones en forma inalámbrica, mediante módulos XBee ZB (o XBee-PRO ZB) con firmware AT¹, dentro de una red Zigbee.

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 ZB es un módulo de 3V, la interconexión entre estos sistemas y el XBee ZB 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.

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 ZB 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.

Por defecto, el canal de RF y el ID de la PAN son elegidos por el coordinador, el cual puede tener una configuración especial o determinarlos mediante scan. Utilizamos los valores por defecto² y asumimos que el lector dispone de sólo una red Zigbee en su alcance.

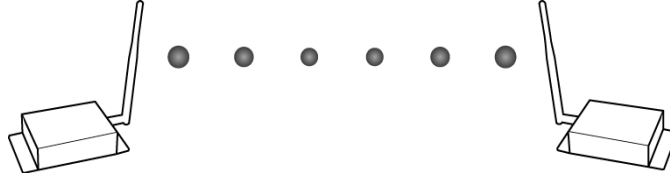
Zigbee es inherentemente una red mesh. Los routers y el coordinador "descubren" la ruta hacia el destinatario del mensaje a enviar; si no hay comunicación directa, entregan sus mensajes al router más cercano, quien a su vez repetirá esta tarea hasta que el mensaje llegue al destinatario. Los end-devices siempre entregan los mensajes a su coordinador (un router o el coordinador de la red Zigbee), quien a su vez almacena los mensajes para éstos hasta tanto despierten y lo contacten, momento en el cual se los entrega. Todo lo relativo al sincronismo y bajo consumo queda entonces manejado por la red Zigbee, sólo debemos tener en cuenta que la aplicación soporte las demoras introducidas por el tiempo de tránsito dentro de la red y el tiempo que un end-device pasa en bajo consumo (con el receptor apagado), si corresponde³.

A continuación, daremos ejemplos de configuración para lograr el efecto de la red cuya topología se plantea, pero como Zigbee es en realidad un stack de protocolo de tres capas, con capa de red, esto es en realidad una comunicación virtual sobre una red física mucho mayor, donde los mensajes de un emisor A a un receptor B en lo que podría verse como una red lógica punto a punto son transportados por varios routers Zigbee dentro de una red mesh. Salvando las distancias, algo así como una red IP. En las explicaciones siguientes nos abstraeremos de esta situación y consideraremos sólo que es visto por la aplicación, es decir, la topología virtual (por llamarla de algún modo para diferenciarla) que estamos analizando.

- 1 El firmware API requiere que se indique la dirección del destinatario en cada mensaje que se envía. El firmware AT permite trabajar con un destino fijo, favoreciendo el tipo de aplicación que describimos en esta nota.
- 2 Es posible cambiar endpoint (E8) y cluster ID (11), tema que no abordamos aquí.
- 3 Un end-device por defecto incluye el parámetro SM=4, lo cual lo configura para dormir periódicamente y reportar a su coordinador al despertar. El tiempo de este ciclo se configura mediante el parámetro SP.

Comunicación 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 ZB con firmware AT 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.

Deberemos configurar en ambos módulos la dirección del otro extremo utilizando los comandos ATDH=<dirección alta> y ATDL=<dirección baja>, es decir, los 32-bits altos y bajos de la dirección de 64-bits que obtenemos mediante ATSH y ATSL (u observamos en X-CTU), respectivamente.

Supongamos que el módulo 1 tiene la dirección 0013A200 12348765 y el módulo 2 la dirección 0013A200 43215678; entonces configuramos:

módulo 1

```
ATDH=13A200
ATDL=43215678
```

módulo 2

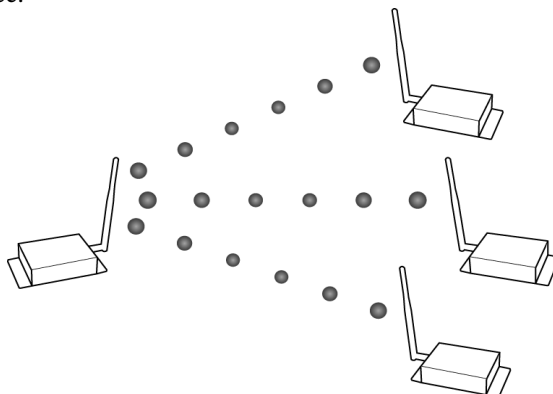
```
ATDH=13A200
ATDL=12348765
```

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

Comunicación punto a multipunto

Caso 1a: comunicación en un sentido (hacia el coordinador), el coordinador es el centro de red

Este es el caso más simple, en el cual tenemos una serie de aplicaciones remotas que reportan información a un punto central, sin que haya comunicación en el sentido inverso, y este punto central coincide con el coordinador de la red Zigbee.



Cada módulo remoto tendrá como dirección de destino la del coordinador, lo cual coincide con la configuración por defecto. De este modo, todo lo que entra por la UART de cada uno de los remotos, sale por la UART del módulo principal y no necesitamos configurar nada.

CTC-059, Utilización de XBee ZB para comunicación de aplicaciones

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 1b: el centro de red es otro módulo cualquiera

La operatoria es idéntica al caso anterior, con la salvedad de que deberemos configurar en los remotos la dirección del módulo central mediante ATDH y ATDL al ponerlos en servicio.

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 ATDH y 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 firmware API, que no desarrollaremos aquí.

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

```
soy 1, hace frio
+++
OK
ATDH=13A200
ATDL=12348765
ATCN
OK
prende el calefactor
soy 2, hace calor
+++
OK
ATDH=13A200
ATDL=43215678
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.

Vemos aquí que el central debe mantener una relación entre el módulo y su dirección, o incluirla en el mensaje. Si utilizamos el modo API, la dirección se entrega en el frame.

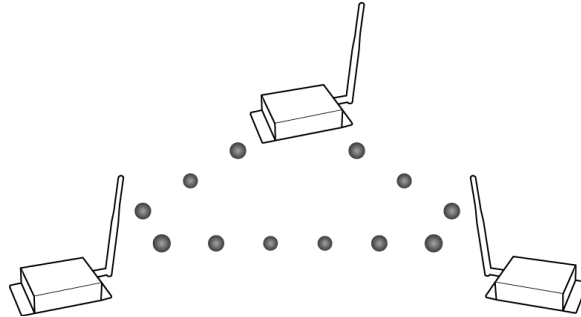
No es necesario volver a configurar ATDH si el valor es el mismo, lo incluimos en el ejemplo por generalidad.

Comunicación 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 ATDH y ATDL antes

CTC-059, Utilización de XBee ZB para comunicación de aplicaciones

de enviar cada mensaje, aunque dependiendo de la frecuencia de envío de mensajes y la aplicación en sí, puede ser preferible utilizar el firmware API.



Broadcasts

Una alternativa cuando la misma información es útil para muchos destinatarios, es realizar un broadcast. Configurando `ATDH=0` y `ATDL=FFFF` se realiza un broadcast, es decir, envía un mensaje que puede ser recibido por todos los miembros de una PAN. Sin embargo, si las "redes" que estuvimos analizando son en realidad parte de una gran red Zigbee, no es aconsejable el uso de broadcasts dado que esto genera mucho tráfico para los routers y provee de información a quien no la solicitó.