Comentario Técnico: CTC-071
Título: **Utilización de XBee Wi-Fi para comunicación de aplicaciones**
Autor: Sergio R. Caprile, Senior Engineer

Revisiones	Fecha	Comentarios
0	21/09/11	

En este comentario técnico estudiaremos las posibles formas de intercomunicar aplicaciones en forma inalámbrica, mediante módulos XBee Wi-Fi.

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

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.

El transporte de datos puede realizarse por UDP o por TCP, configurado mediante el comando *ATIP*. Por ejemplo *ATIP=0*, configura el uso de UDP, lo cual corresponde al valor por defecto.

La comunicación puede ser entre módulos XBee Wi-Fi o con hosts IP. En este último caso, en CAN-096 se describe la forma de utilizar los XBee IP Services, en particular el Serial Communication Service que es el que provee esta facilidad. De igual modo, también es posible operar con o sin un Access Point (modo ad-hoc o infrastructure). Los ejemplos consideran el caso más común de utilizar una red existente, con un Access Point.

Si bien es posible obtener direcciones IP por DHCP, para poder comunicarnos con un módulo necesitamos conocer su dirección. Si bien podemos configurar al servidor DHCP para que asigne una en particular según la MAC o recurrir a otras estrategias, utilizaremos mayormente direcciones estáticas, configurando *ATMA=1*. Si ambos corresponsales están en diferentes redes, deberá configurarse la dirección del router con *ATGW*, por ejemplo: *ATGW192.168.1.1*

Punto a punto

Este es el caso más simple posible, en el cual tenemos dos aplicaciones que requieren dialogar entre sí.

Por defecto, el XBee Wi-Fi 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 las direcciones utilizando el comando *ATMY<dirección>* para la dirección propia y *ATDL<dirección>* para la dirección del remoto. Esto lo haremos en ambos módulos, por ejemplo:

módulo 1
ATMY192.168.1.1
ATDL192.168.1.2
ATMK255.255.255.0
módulo 2
ATMY=192.168.1.2
ATDL=192.168.1.1
ATMK255.255.255.0

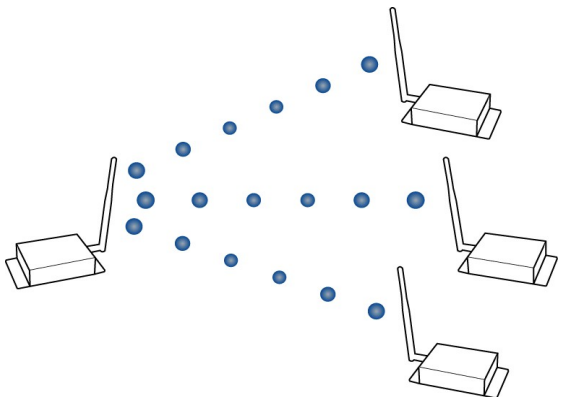
Dado que cada módulo no sabe en qué momento el otro va a transmitirle algo, es necesario que ambos estén siempre atentos a lo que el otro pueda decirle, o haya un tercero que arbitre. Esto requiere que ninguno de los dos 'duerma', es decir, entre en modo bajo consumo o si lo hacen sea en el modo AP Associated Sleep, en el que el Access Point provee el almacenamiento temporario mientras duermen. Con la configuración por defecto, los módulos no dormirán.

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.



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=192.168.1.11
ATDL=192.168.1.1
ATMK255.255.255.0

módulo 2

ATMY=192.168.1.12
ATDL=192.168.1.1
ATMK255.255.255.0

módulo 3

ATMY=192.168.1.13
ATDL=192.168.1.1
ATMK255.255.255.0

módulo central

ATMY=192.168.1.10
ATMK255.255.255.0

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.

Dado que es la aplicación la que controla el envío de mensajes, tal vez lo más indicado sea configurar ATSO=0 (deep sleep) y *ATSM=1* (pin sleep), con lo cual el módulo duerme hasta que la aplicación (el micro conectado) lo despierta activando el pin $\overline{DTR}$. Deberá tenerse en consideración que se deberá esperar a que el módulo se asocie al Access Point.

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 1400 bytes.

Caso 2a: comunicación en ambos sentidos con un módulo como centro de red

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 descripta, vista desde el sistema conectado al módulo central. Para mayor claridad, hemos resaltado *lo que envía este sistema* y las respuestas del módulo:

```
soy 192.168.1.11, hace frio
+++
OK
ATDL192.168.1.11
ATCN
OK
prende el calefactor
soy 192.168.1.12, hace calor
+++
OK
ATDL192.168.1.12
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.

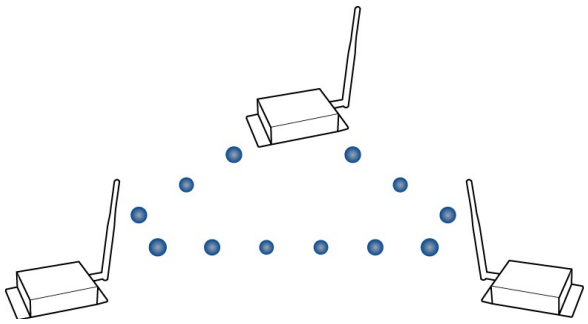
En este caso, al haber comunicación en ambos sentidos, los remotos no saben en qué momento el central va a transmitirles algo; esto nos impide utilizar el modo deep sleep, a menos que hagamos algunas restricciones. Si el central puede esperar a que el remoto en cuestión le hable primero, entonces envía 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.

Caso 2b: comunicación en ambos sentidos con un host IP como centro de red

Dado que un host IP puede abrir y cerrar un socket a gusto, puede comunicarse con el remoto utilizando el esquema descripto en CAN-096. Asimismo, determinará la dirección de quien envía observando el socket. Las consideraciones para el uso de deep sleep siguen siendo las mismas.

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 o haya un mediador, lo cual requiere que ninguno entre en modo deep sleep. 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 en *ATDL* la dirección *255.255.255.255* realiza un broadcast, es decir, envía un mensaje que puede ser recibido por todos los miembros de una red. Esto requiere además que el transporte utilizado sea UDP.