

Revisiones	Fecha	Comentarios
0	10/09/08	

En este comentario técnico estudiaremos la forma de configurar los módulos XBee (o XBee-PRO) ZB para utilización en redes Zigbee con aplicaciones de sensores remotos.

Introducción

En los casos que veremos a continuación, el módulo transmitirá la información de estado de sus entradas y salidas a uno o varios módulos remotos. Las formas de configuración para envío de información a módulos remotos en las diversas topologías han sido descritas en CTC-059.

Cada uno de los pines DIO x que puede ser configurado para su operación como entrada analógica, entrada digital, o salida digital, posee un comando *ATD x* que permite operar sobre esta selección. Los pines DIO10 a DIO12 se configuran mediante los comandos *ATP0* a *ATP2*, respectivamente.

Entradas analógicas

Para configurar alguna de las entradas como analógica, es decir, asociada al conversor analógico a digital, utilizamos el comando que le corresponde, por ejemplo *ATD2=2* setea el pin AD2/DIO2 como AD2, es decir, asociado al ADC. Los módulos poseen una referencia interna de 1,2V. No debe conectarse el pin VREF dado que no se lo utiliza.

Entradas digitales

Para configurar alguna de las entradas como digital, utilizamos el comando que le corresponde, por ejemplo *ATD0=3* setea el pin AD0/DIO0 como entrada (DI). El comando *ATP1=3* setea el pin DIO11 como entrada (DI).

Lectura local de la información

Es posible leer el estado de los pines desde la UART, mediante el comando *ATIS*. Este comando fuerza una muestra inmediata, y devuelve por la UART la información. En el firmware AT, dicha información estará en ASCII:

```
+++OK
ATIS
01
081C
03
0818
03FF
03FF
```

Este método en realidad sólo es confiable si no tomamos muestras automáticamente, dado que si es así, en realidad estamos tomando muestras de ese proceso de muestreo. El primer número que observamos es la cantidad de muestras que hay en el buffer, el segundo es la configuración de canales digitales y el tercero la de canales analógicos; el siguiente es el estado de las entradas y salidas digitales habilitadas, y sólo está presente si las hay. A continuación, los valores de todos los canales AD habilitados.

Si utilizamos el firmware API, deberemos enviar el comando dentro de una trama de comando y recibiremos la respuesta dentro de una trama de respuesta. El formato de las tramas se describe en el manual del usuario, un ejemplo de como generar tramas API ha sido descrito en CAN-088. Por ejemplo, si no estamos trabajando en el modo con caracteres de escape, el siguiente mensaje:

```
7E 00 04 08 12 49 53 49
```

solicita una muestra.

7E: inicio de trama
 00 04: longitud de datos (4 bytes a continuación, más uno de checksum)
 08: comando
 12: identificador de trama (cualquier número distinto de cero)
 49 53: IS, el comando
 49: checksum

Ejecutada la acción, recibiremos una trama conteniendo el resultado:

```
7E 00 0F 88 12 49 53 00 01 08 1C 03 00 18 03 FF 03 FF 85
```

7E: inicio de trama
 00 0F: longitud de datos (15 bytes a continuación, más uno de checksum)
 88: respuesta a comando
 12: identificador de trama (el valor que enviamos en el comando)
 49 53: IS, el comando enviado
 00: resultado, OK
 01: cantidad de muestras
 08 1C: I/O habilitados (DIO11, DIO4, DIO3, DIO2)
 03: canales analógicos habilitados (AN1, AN0)
 00 18: estado de I/O
 03 FF: valor de AN0
 03 FF: valor de AN1
 85: checksum

Lectura remota de la información

Es posible solicitar la ejecución del comando *ATIS* a un módulo remoto. Para esto, el módulo XBee ZB (o XBee-PRO ZB) que controla debe tener el firmware API, dado que la solicitud de comandos remotos se realiza mediante una trama especial en la que se indica la dirección del módulo sobre el que se opera, y el comando en sí. El módulo controlado puede tener cualquier firmware.

Por ejemplo, si no estamos trabajando en el modo con caracteres de escape, el siguiente mensaje:

```
7E 00 0F 17 12 00 13 A2 00 40 0A 5E 3B FF FE 00 49 53 A5
```

solicita al módulo con dirección 00 13 A2 00 40 0A 5E 3B que tome una muestra.

7E: inicio de trama
 00 0F: longitud de datos (15 bytes a continuación, más uno de checksum)
 17: comando remoto
 12: identificador de trama (cualquier número distinto de cero)
 00 13 A2 00 40 0A 5E 3B: dirección IEEE del destinatario
 FF FE: dirección corta del destinatario, FFFE indica que no se la conoce
 00: opciones
 49 53: IS

A5: checksum

Ejecutada la acción, recibiremos una trama conteniendo el resultado:

```
7E 00 19 97 12 00 13 A2 00 40 0A 5E 3B 5E 6A 49 53 00 01 08 1C 03 00 18 03
FF 03 FF 16
```

7E: inicio de trama

00 19: longitud de datos (25 bytes a continuación, más uno de checksum)

97: respuesta a comando remoto

12: identificador de trama (el valor que enviamos en el comando)

00 13 A2 00 40 0A 5E 3B: dirección IEEE de quien responde

5E 6A: dirección corta de quien responde

49 53: IS, el comando enviado

00: resultado, OK

01: cantidad de muestras

08 1C: I/O habilitados (DIO11, DIO4, DIO3, DIO2)

03: canales analógicos habilitados (AN1, AN0)

00 18: estado de I/O

03 FF: valor de AN0

03 FF: valor de AN1

16: checksum

Envío periódico de la información

En el caso más común, nos interesa enviar la información de forma periódica a algún módulo remoto. El comando *ATIR* nos permite setear la frecuencia de muestreo en milisegundos, por ejemplo *ATIR=3E8* configura al módulo para tomar muestras cada 1s (recordemos que los valores son en hexadecimal).

Recepción de la información

En el módulo que recibe la información de I/O, deberemos tener el firmware API. La información proveniente del otro módulo se envía por la UART en una trama de I/O¹. A menos que explícitamente estemos trabajando en el modo con caracteres de escape, tendremos algo como lo siguiente:

```
7E 00 16 92 00 13 A2 00 40 0A 5E 3A F2 64 01 01 08 1C 03 00 18 02 B1 02 DF
AB
```

7E: inicio de trama

00 16: longitud de datos (22 bytes a continuación, más uno de checksum)

92: recepción de datos de I/O

00 13 A2 00 40 0A 5E 3A: dirección IEEE de quien envía

F2 64: dirección corta de quien envía

01: se ha enviado ACK

01: cantidad de muestras

08 1C: I/O habilitados (DIO11, DIO4, DIO3, DIO2)

03: canales analógicos habilitados (AN1, AN0)

00 18: estado de I/O

02 B1: valor de AN0

02 DF: valor de AN1

AB: checksum

¹ Si se habilita el modo *explícito* el formato es el de una trama explícita, es decir, incluye información de direccionamiento Zigbee (endpoint, cluster-ID, profile-ID). Asumimos que esto no es así.

Bajo consumo

Si el módulo es un end-device, al salir de su letargo, si tiene configurado algún valor de *ATIR*, toma una muestra y la envía. Si la señal a medir no está lista al momento de despertar, la medición puede ser errónea, por ejemplo si apagamos una referencia de tensión externa para minimizar aún más el consumo. En este caso, es conveniente configurar una frecuencia de muestreo que permita tomar más de una muestra y descartar la primera. El módulo sigue tomando muestras mientras está despierto; el tiempo que permanece en actividad se configura mediante el comando *ATST*. El tiempo que no está activo se indica mediante el comando *ATSP*. Por ejemplo, seteando *ATSM=4*, *ATSP=100*, *ATIR=32* y *ATST=70*, el módulo permanece en bajo consumo por 2,56 segundos, despierta, luego toma una muestra y la transmite, espera 50ms, luego toma otra muestra y la transmite, espera unos 12ms por actividad en la interfaz, y si no la hay se va a dormir, repitiendo el ciclo.

Si se configura *ATIR > ATST*, el módulo toma y transmite sólo una muestra.