

Revisiones	Fecha	Comentarios
0	18/02/08	
1	20/02/08	amplía bajo consumo
2	10/09/08	diferencia ADC respecto a otros XBee

En este comentario técnico estudiaremos la forma de configurar los módulos XBee (o XBee-PRO) 802.15.4 para utilización en 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 802.15.4 han sido descritas en CTC-054.

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.

Cada uno de los pines $DIOx$ que puede ser configurado para su operación como entrada analógica, entrada digital, o salida digital, posee un comando $ATDx$ que permite operar sobre esta selección.

Entradas analógicas

Para configurar alguna de las entradas como analógica, es decir, asociada al convertor 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. Debe conectarse una referencia de tensión al pin VREF; es posible utilizar la alimentación del módulo como referencia.

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).

Lectura 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. Si no estamos utilizando el modo API, dicha información estará en ASCII:

```
+++OK
ATIS
1
1415
011
1DC
1FD
```

Caso contrario, la respuesta al comando será un 'OK', y recibiremos una trama de datos en modo API. 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, el siguiente es el estado

de las entradas y salidas habilitadas, y sólo está presente si las hay. A continuación, los valores de todos los canales AD habilitados.

Envío 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=64* configura al módulo para tomar muestras cada 100ms (recordemos que los valores son en hexadecimal).

Para evitar enviar constantemente información, podemos tomar una determinada cantidad de muestras, y luego enviarla. Dado que disponemos de una capacidad limitada de buffer debemos administrarla, por ejemplo disponemos de espacio para cuarenta y seis muestras, si habilitamos dos canales de AD, podremos almacenar un total de veintitres muestras de ambos. Esto se configura mediante el comando *ATIT*, por ejemplo *ATIT=4* toma cuatro muestras de todos los canales, espaciadas según indica *ATIR*, y luego las envía al módulo remoto cuya dirección hayamos configurado.

Bajo consumo

Si el módulo se configura para bajo consumo, al salir de su letargo toma una muestra y la envía. Esto sucede siempre, aún cuando no haya una frecuencia de muestreo configurada (*ATIR=0*). En este caso, 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, esto permite que el conversor espere un tiempo al despertar.

La forma de operar en bajo consumo puede configurarse de modo similar a como hiciéramos en CTC-054, aunque en el caso de un sensor remoto sin ninguna otra aplicación, podemos aprovechar la facilidad de *cyclic sleep* del módulo. El comando *ATSM=4* habilita el modo *cyclic sleep*, en el cual el módulo duerme durante un tiempo, indicado por el comando *ATSP*. Al despertar, tomará tantas muestras como indique *ATIT*, espaciadas el tiempo que indique *ATIR*. A continuación, permanecerá un cierto tiempo en espera de actividad en la interfaz serie o la RF, y volverá a dormir. Este tiempo se configura mediante el comando *ATST*. Por ejemplo, seteando *ATSM=4*, *ATSP=64*, *ATIT=1*, *ATIR=14* y *ATST=1*, el módulo permanece en bajo consumo por 1 segundo, luego espera 20ms, toma una muestra y la transmite, espera 1ms por actividad, y repite el ciclo. El tiempo que el módulo permanece encendido es: $t_{on} = t_{IR} \cdot IT + t_{wakeup} + t_{Tx} + t_{ST} \approx 25 \text{ ms}$, y el que permanece apagado es $t_{off} = t_{SP}$.

En estas condiciones, la corriente consumida es: $I = \frac{I_{Rx} \cdot t_{on} + I_{sleep} \cdot t_{off}}{t_{on} + t_{off}} \approx 1,3 \text{ mA}$, despreciando la diferencia de consumo en transmisión (similar en Xbee y mucho mayor en XBee Pro).

Recepción de la información

En el módulo remoto, esté o no configurado el modo API, la información de I/O proveniente del otro módulo se envía por la UART en formato API. A menos que explícitamente estemos trabajando en el modo con caracteres de escape, tendremos algo como lo siguiente:

```
7E 00 14 83 65 87 49 00 02 14 01 00 01 01 23 02 34 00 00 01 24 02 35 79
```

La descripción completa del formato API se encuentra en el manual del usuario, sin embargo daremos una pequeña introducción para poder tomar un pequeño atajo:

7E: inicio de trama
 00 14: longitud de datos (20 bytes a continuación, más uno de checksum)
 83: recepción de datos de I/O
 65 87: dirección de quien envía
 49: RSSI (nivel de recepción)
 00: opciones
 02: cantidad de muestras
 14 01: canales habilitados (AN3, AN1, D0)

CTC-055, Utilización de XBee 802.15.4 para sensores remotos

00 01: estado de I/O, muestra 1
01 23: valor de AN1, muestra 1
02 34: valor de AN3, muestra 1
00 00: estado de I/O, muestra 2
01 24: valor de AN1, muestra 2
02 35: valor de AN3, muestra 2
79: checksum

Como podemos apreciar, más allá de implementar el modo API, una vez configurado el módulo transmitirá siempre el mismo formato, por lo que podemos directamente tomar los bytes que nos interesan para acceder a la información. Obsérvese que los valores de 16-bits se transmiten de modo que el byte más significativo va primero.

En el caso visto al analizar la lectura de la información desde el mismo módulo, la información recuperada en un módulo remoto sería la siguiente:

```
7E 00 0E 83 xx xx rr oo 01 14 15 00 11 01 DC 01 FD ch
```

donde *xxxx* es la dirección del módulo remoto, *rr* el nivel de recepción, *oo* las opciones, *ch* el checksum