

How to configure and control remote XBee RF modules

Overview

This tutorial explains how to configure and control a remote XBee RF module for data acquisition and control using API mode commands.

Using API commands we can send AT mode commands to read and write analog and digital I/O pins on a remote XBee RF module.

These examples can be expanded to control multiple remote XBee RF modules.

Requirements

These features are only supported by Digi XBee 802.15.4 RF modules running firmware version 10E6 or later.

Setup

These are the 64-bit addresses of the XBee modules we used in this tutorial:

Base:
SH=0013A200
SL=404BED76

Remote:
SH=0013A200
SL=404BED83

We used a PC running Digi's X-CTU to communicate with the XBee base module.

These are the commands used to enter API mode on the XBee base module. The remote XBee module remained in AT command mode.

```
+++OK      enter AT command mode
ATAP1     enable API mode 1
OK
ATWR      save changes to nonvolatile memory
OK
ATCN      exit AT command mode
```

One useful trick to remember is that even though API mode is enabled, the XBee module still responds to AT commands. This makes it easy to change back to non-API mode without using the more complicated API commands.

A brief explanation of API mode

XBee 802.15.4 modules have two modes of operation; Transparent and API mode. By default, XBee modules start in Transparent mode; all serial data is immediately sent to the remote XBee module, but the base XBee module responds to AT commands.

In API mode, all serial data is ignored unless it is in an API command. But using API commands you can send AT commands to the base and remote XBee module or send and receive serial data from a remote XBee module. This tutorial takes advantage of the fact you can send AT commands to a remote XBee RF module.

The following diagram dissects a typical API command. This is the command used in the first example; “How to configure an I/O pin as an analog input on a remote radio”.

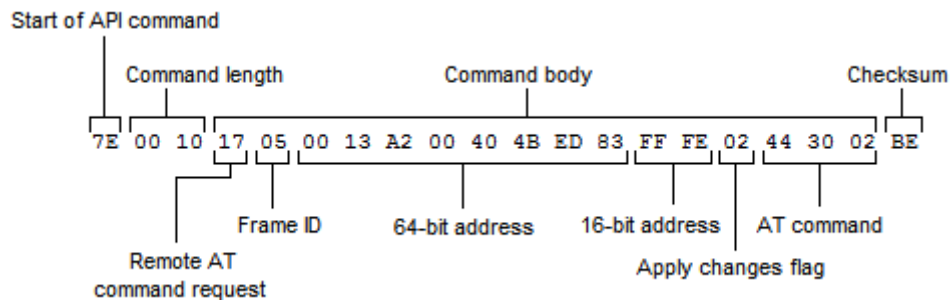


Figure 1: Structure of a typical API command.

The diagram shows the common structure of all API commands above the command.

Start of API command – Every API begins with 0x7E. Everything before it is ignored.

Command length – The next two bytes contain the length of the rest of the command, excluding the checksum.

Command body – The rest of the bytes, excluding the checksum, contain a specific API command.

Checksum – The sum of all the bytes after the command length, including the checksum, should be 0xFF. To calculate the checksum, sum all of the bytes after the command length, keep only the least significant 8 bits and subtract from 0xFF.

And the diagram shows the structure of the “remote AT command request” below the command.

Remote AT command request – All remote AT command requests begin with 0x17.

Frame ID – If the frame ID in an API command is non-zero, then the response will contain the ID.

64-bit address – This is the 64-bit XBee module address

16-bit address – This is the 16-bit XBee module address. By default this address is not used but can be set using the MY command. If 64-bit addressing is being used, then this field should be FF FE.

Apply changes flag – If this byte is 0x02, then changes take effect immediately on the remote XBEE module. If this byte is 0x00, then you must send an AC command for changes to take effect.

AT command – This is the AT command you are sending to the remote XBee module. In Figure 1, 44 30 02 translates to D02, which configures digital pin 0 to be an analog input.

There are 10 API commands in total. In this tutorial we mostly use the “remote AT command request”. The “remote AT command request” sends an AT command to a remote XBee module and causes the remote module to execute it.

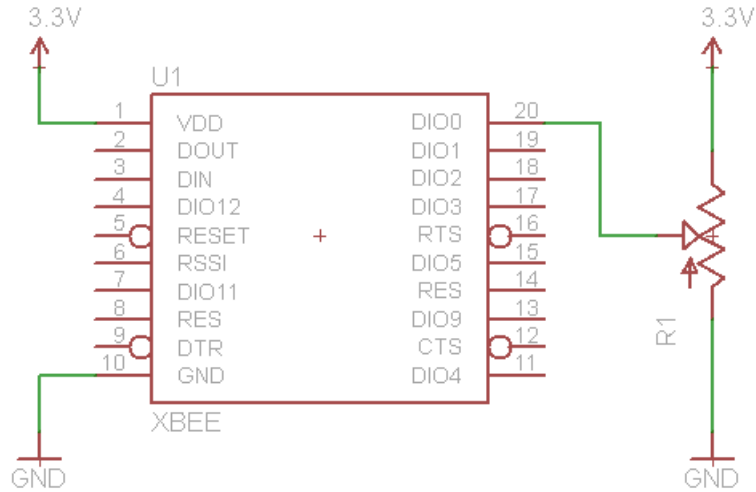
Sending a command to a remote radio

Using the “remote AT command request” we can send any AT command to a remote XBee RF module. Most of the examples in this tutorial are based on sending AT commands to a remote radio.

In the following examples, commands are in blue, responses are in red.

How to configure an I/O pin as an analog input on a remote radio

The XBee RF module has 7 pins that can be used as analog inputs. In this example we remotely configure DIO0 as an analog input on the remote RF module from the base RF module using an API command.



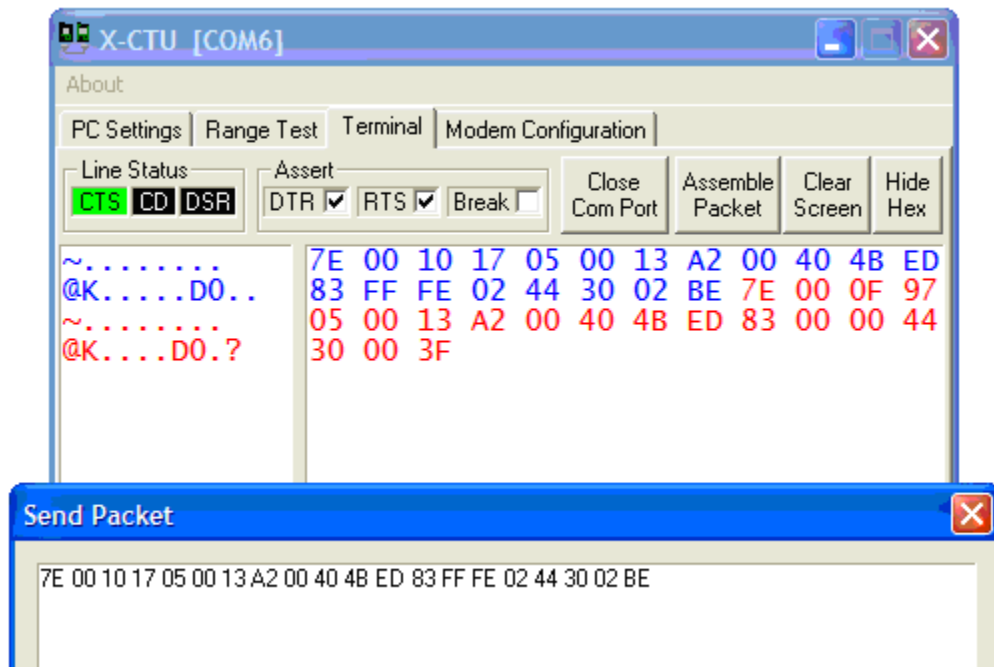
Example circuit showing DIO0 connected to analog source

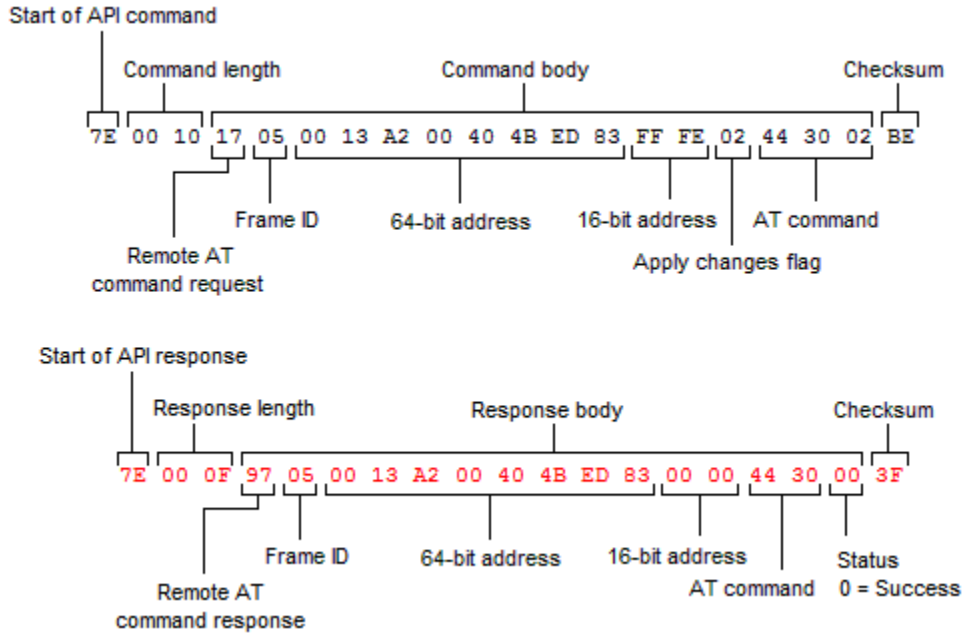
The D02 command configures DIO0 as an analog input; 0x44, 0x30, 0x02 is the D02 command in ASCII (remember the parameter is in binary).

The command to configure DIO0 as analog input is:

```
7E 00 10 17 05 00 13 A2 00 40 4B ED 83 FF FE 02 44 30 02 BE
```

0x44, 0x30 is 'D0' in ASCII, 0x02 is the parameter value.

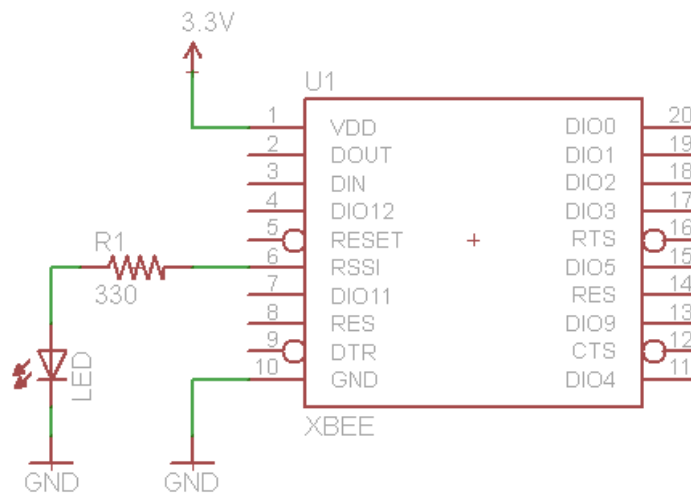




How to configure a PWM pin as an output on a remote radio

Each XBee 802.15.4 RF module has two pins that can be configured as PWM outputs; i.e., RSSI/PWM0 and DIO11/PWM1. This example demonstrates how to configure PWM0 as a PWM output on a remote XBee module.

Like most of these examples, we use the Remote AT Command Request API command. The AT command we use is the P0 (PWM0 Configure) command.

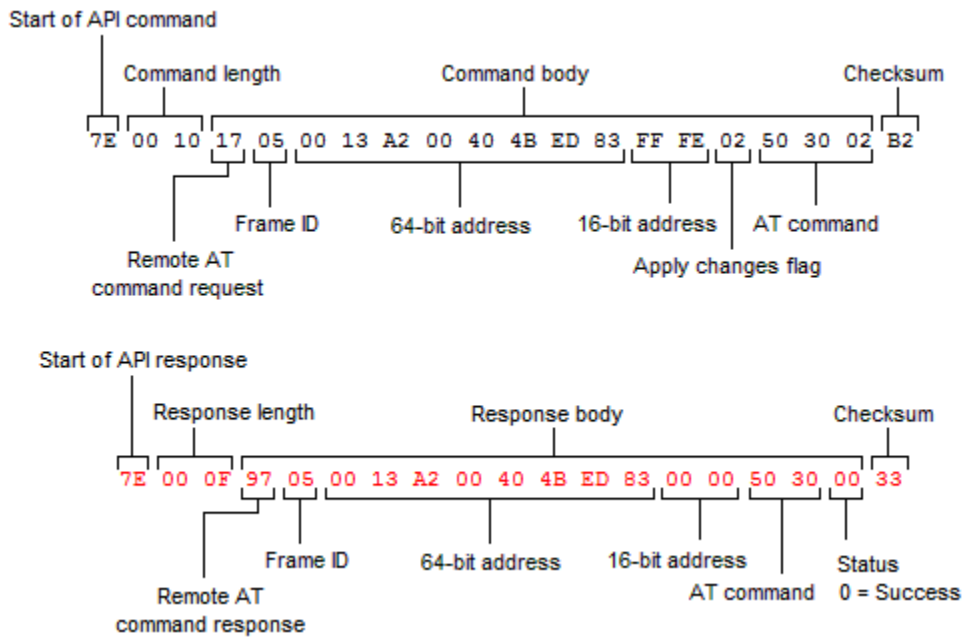
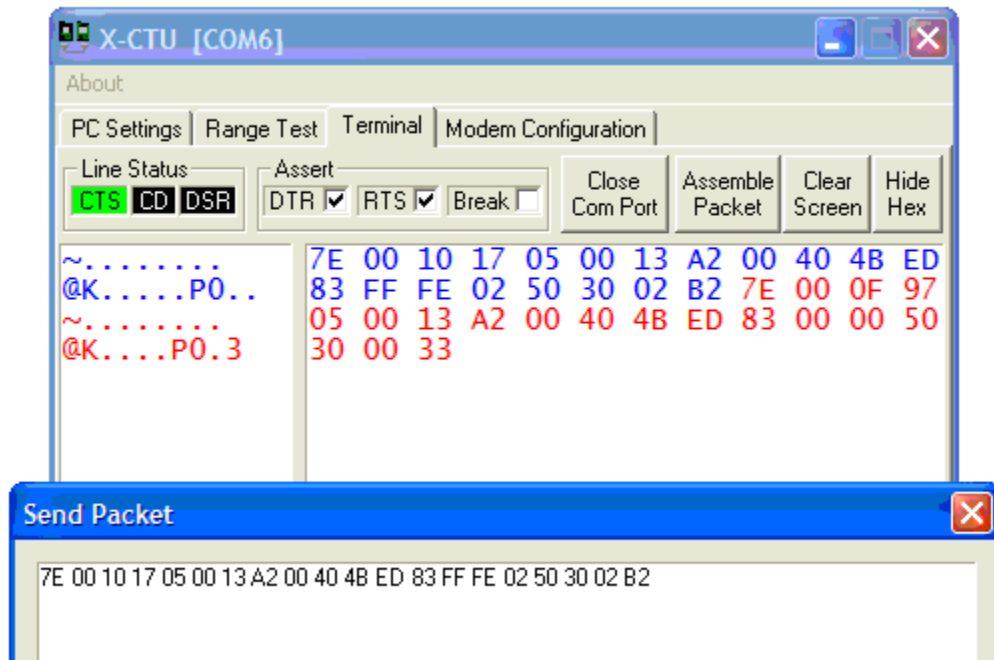


Example circuit showing LED connected to PWM0

The complete command to configure PWM0 as pseudo-analog output is:

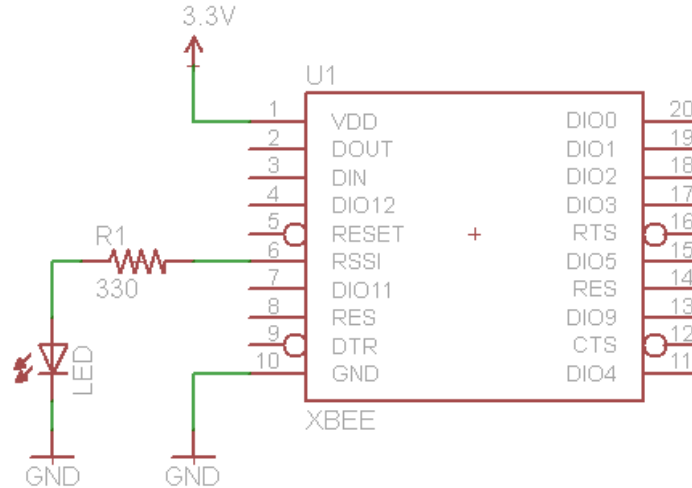
```
7E 00 10 17 05 00 13 A2 00 40 4B ED 83 FF FE 02 50 30 02 B2
```

0x50, 0x30 is 'P0' in ASCII, 0x02 is the parameter value.



How to set a PWM output on a remote radio to 50% duty cycle

This example shows how to set the PWM0 output on a remote XBee module to 50% duty cycle. It assumes you have previously configured PWM0 as a PWM output.



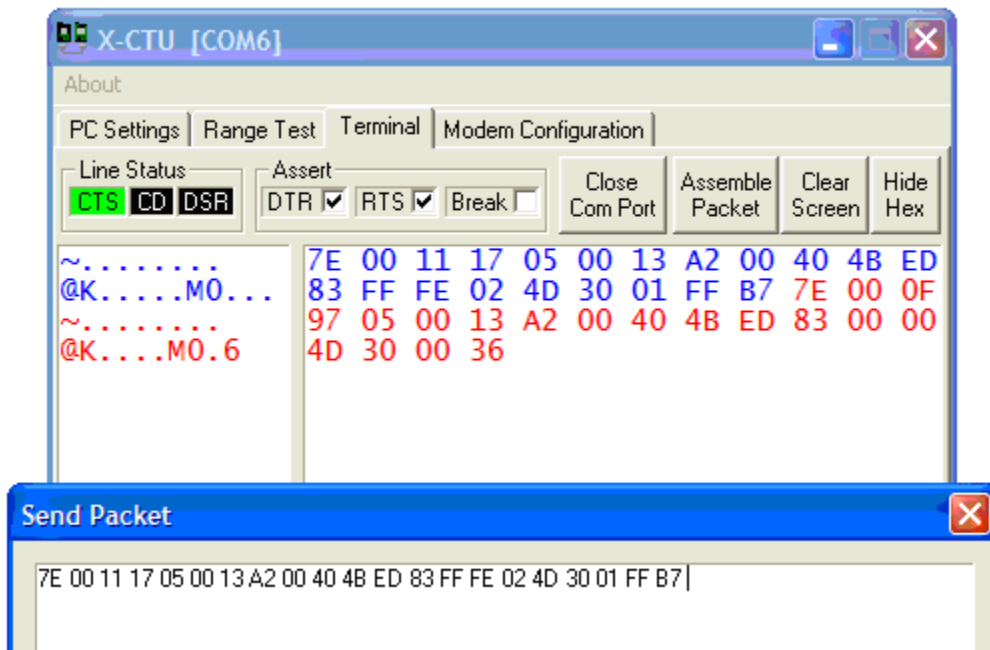
Example circuit showing LED connected to PWM0

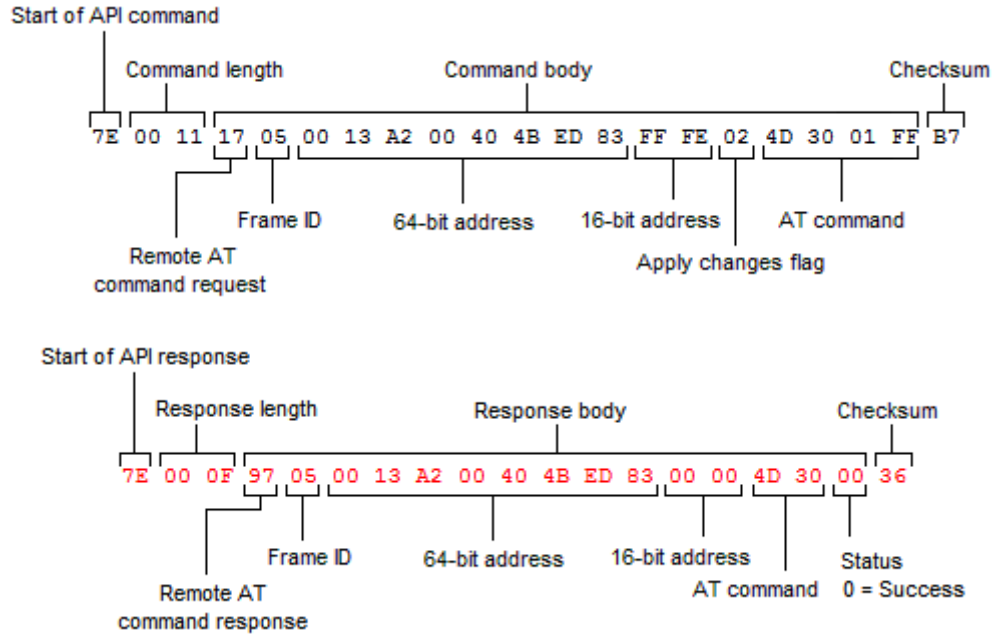
Like most of these examples, we use the Remote AT Command Request API command. The AT command used is the M0 (PWM0 Output Level) command.

The complete command to set PWM0 to 50% duty cycle is:

```
7E 00 11 17 05 00 13 A2 00 40 4B ED 83 FF FE 02 4D 30 01 FF B7
```

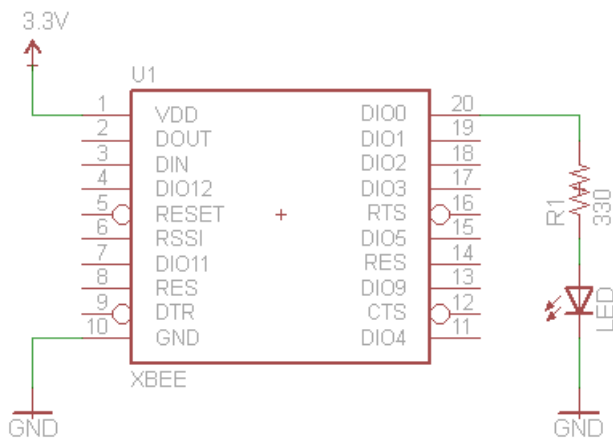
0x4D, 0x30 is 'M0' in ASCII, 0x01, 0xFF is the parameter value.





How to turn-on a digital output on a remote radio

This example demonstrates how to turn-on the DIO0 output on a remote XBee module.



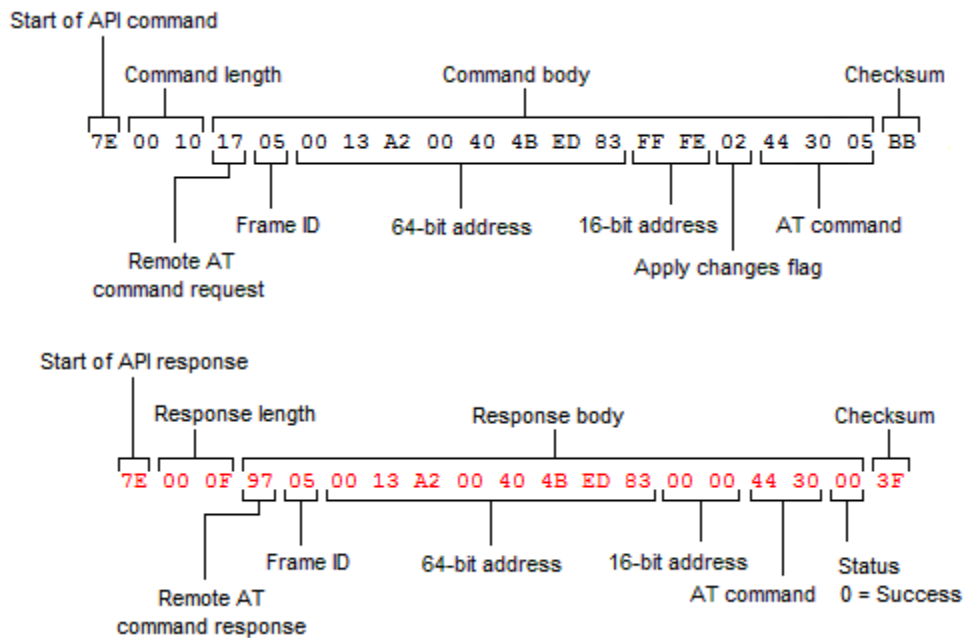
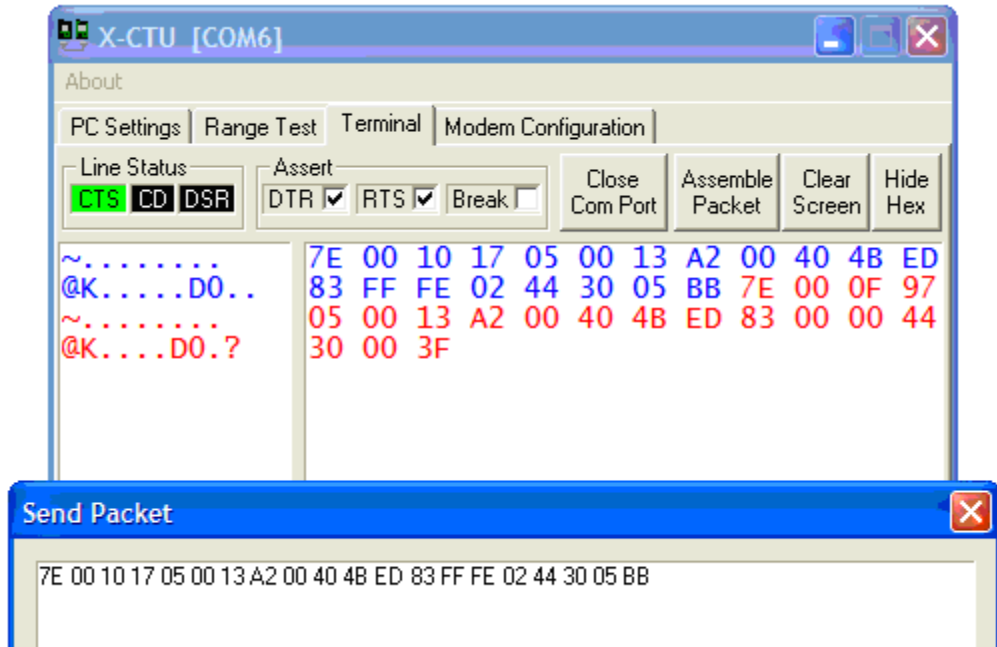
Example circuit showing LED connected to DIO0

Like most of these examples, we use the Remote AT Command Request API command. The AT command we use is the D0 (DIO_n Configuration) command.

The complete command to turn-on DIO0 is:

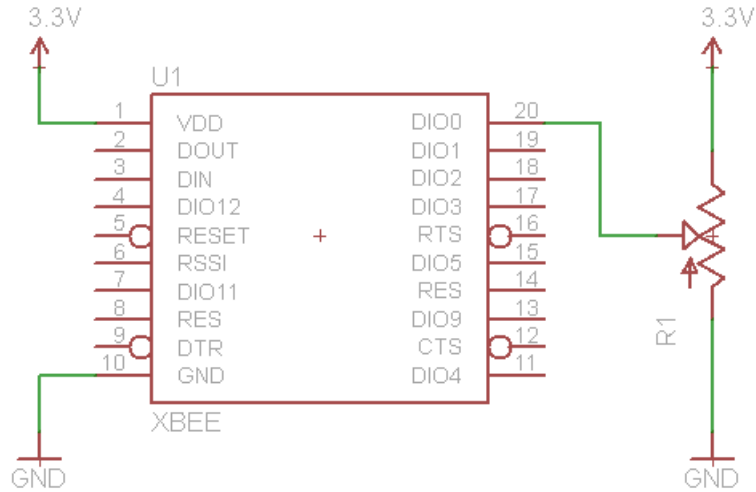
```
7E 00 10 17 05 00 13 A2 00 40 4B ED 83 FF FE 02 44 30 05 BB
```

0x44, 0x30 is 'D0' in ASCII, 0x05 is the parameter value.



How to poll a remote XBee module

This example demonstrates how to poll a remote XBee module. It assumes you have previously configured one or more pins as analog or digital inputs. In this example we have configured DIO0 as an analog input.

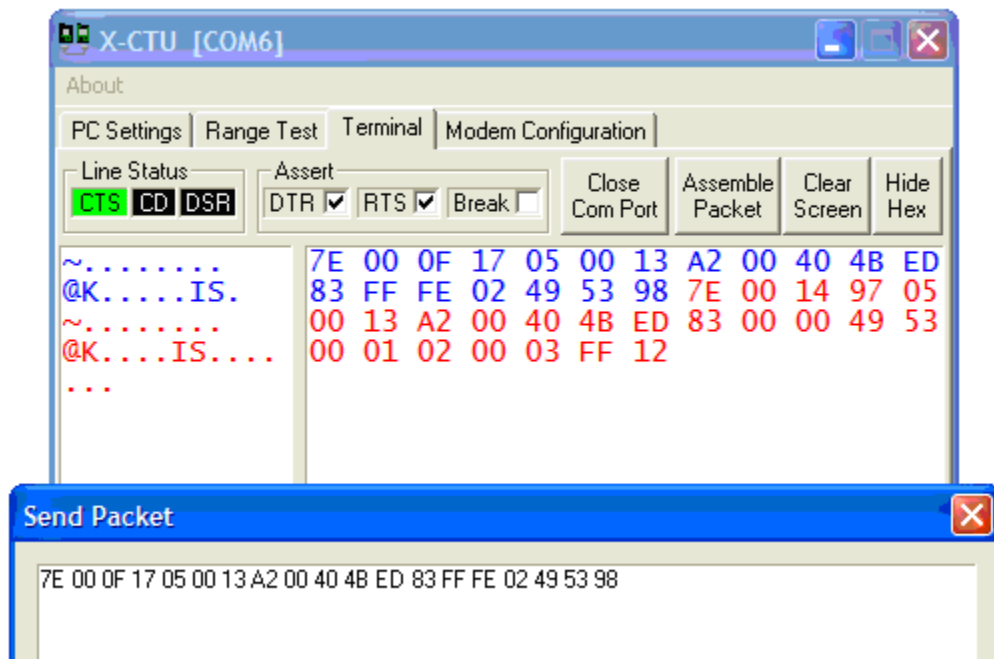


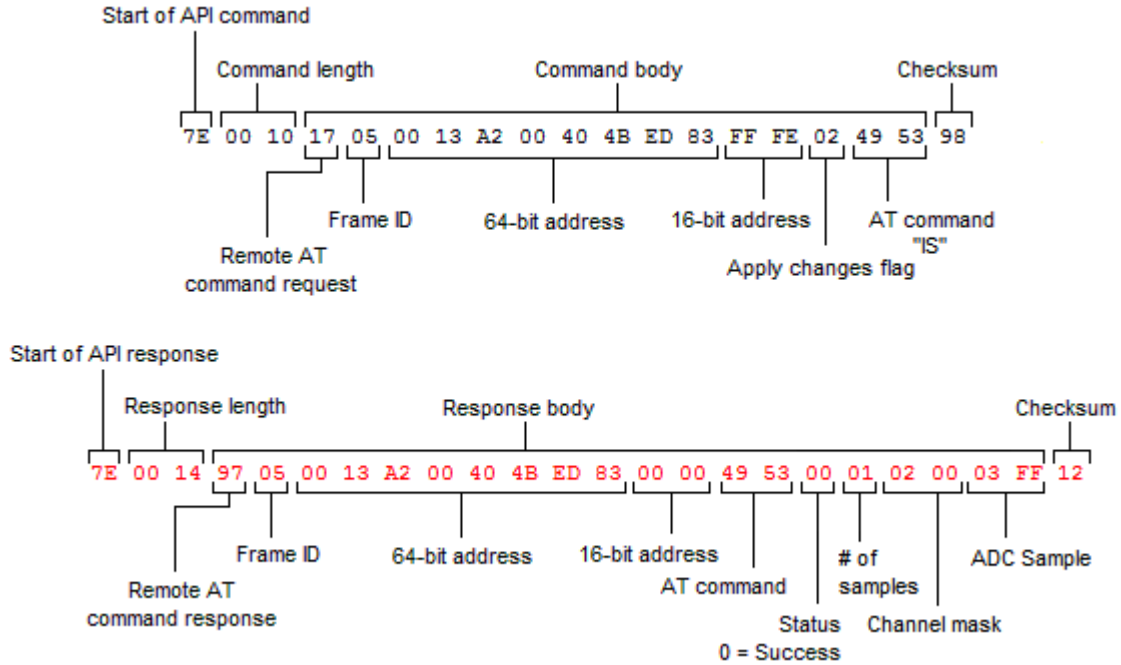
Example circuit showing DIO0 connected to analog source

Like most of these examples, we use the Remote AT Command Request API command. The AT command we use is the IS (Force Sample) command.

The API command to poll a remote XBee module is:

```
7E 00 0F 17 05 00 13 A2 00 40 4B ED 83 FF FE 02 49 53 98
```





How to configure a digital input to trigger on a state change

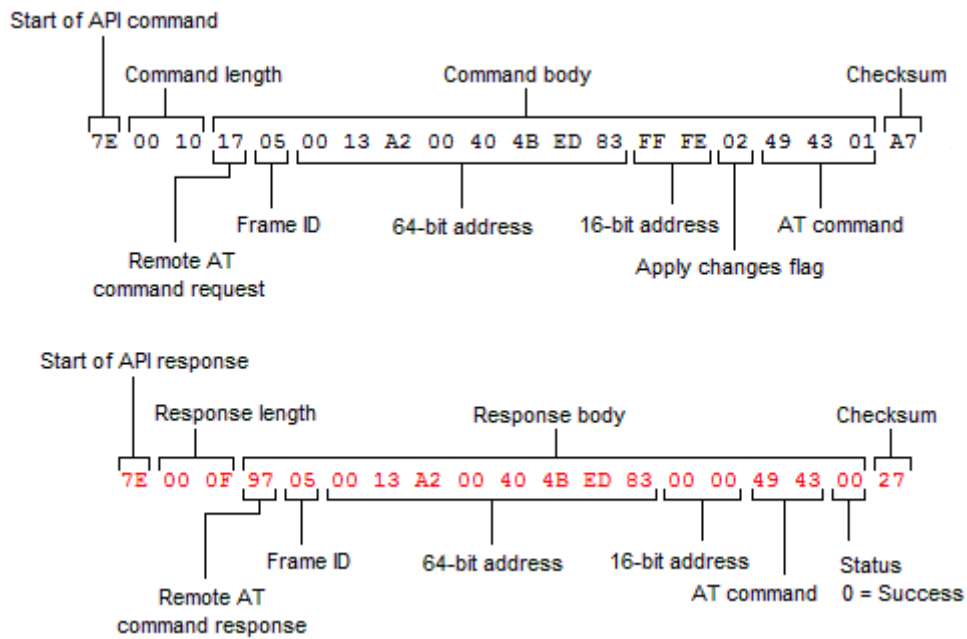
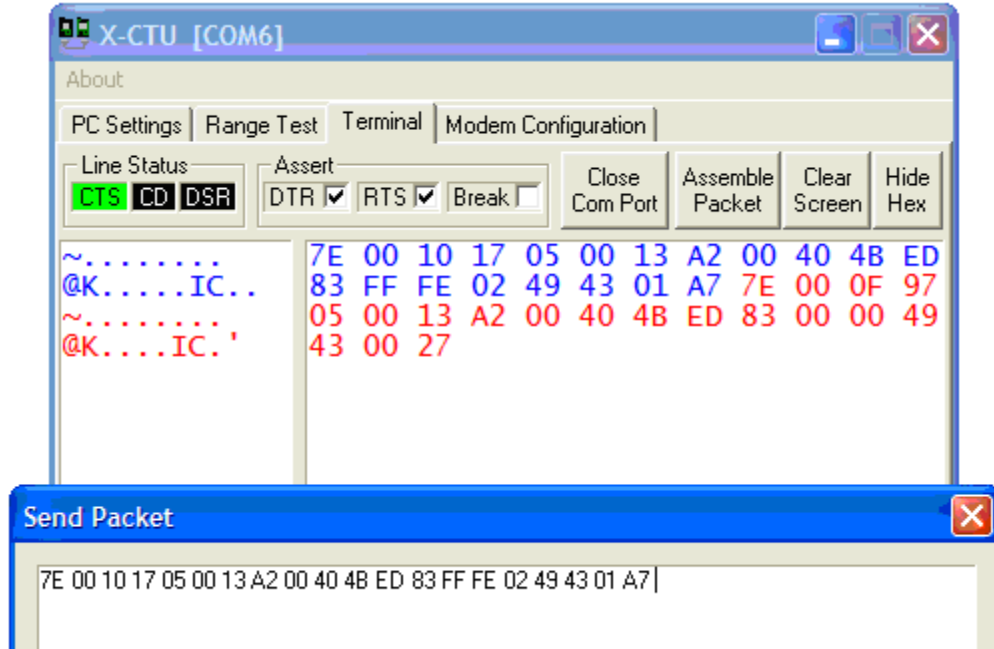
This example demonstrates how to configure a digital input on a remote XBee module to trigger a transmission on a change of state of a digital input pin using the IC command.

The parameter of the IC command is a bitfield indicating which input pin(s) to monitor for a change of state. In this example we are monitoring bit 0 or the DIO0 pin.

The API command to configure a remote XBee module to detect changes on DIO0 is:

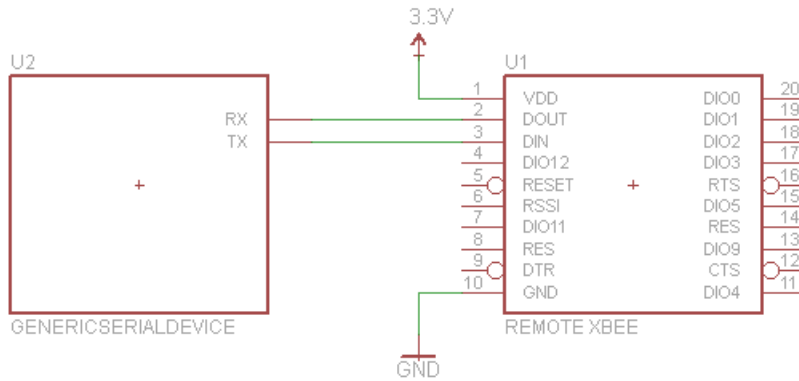
```
7E 00 10 17 05 00 13 A2 00 40 4B ED 83 FF FE 02 49 43 01 A7
```

0x49, 0x43 is 'IC' in ASCII, 0x01 is the parameter value for DIO0.



How to send serial data to an external device attached to a remote XBee module

This example demonstrates how to send serial data to an external device attached to a remote XBee module from a base module.



Example circuit showing XBee connected to generic serial device

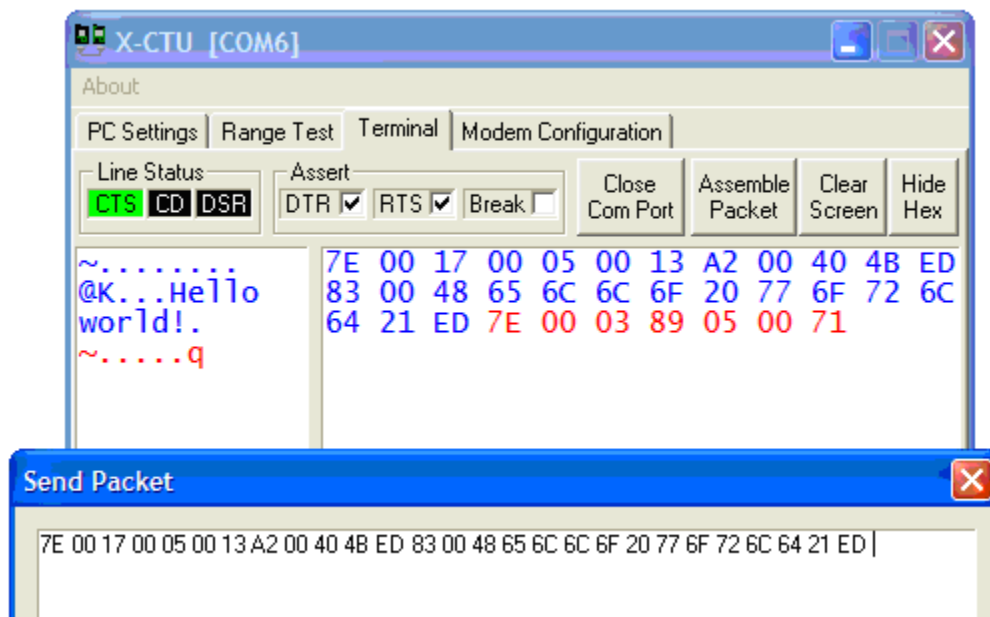
When both base and remote modules are in AT command mode, all serial data send to the base module from the host (except for AT commands) is sent to the remote module. Things are more difficult when the base module is in API mode.

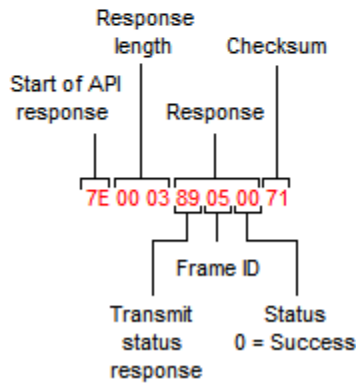
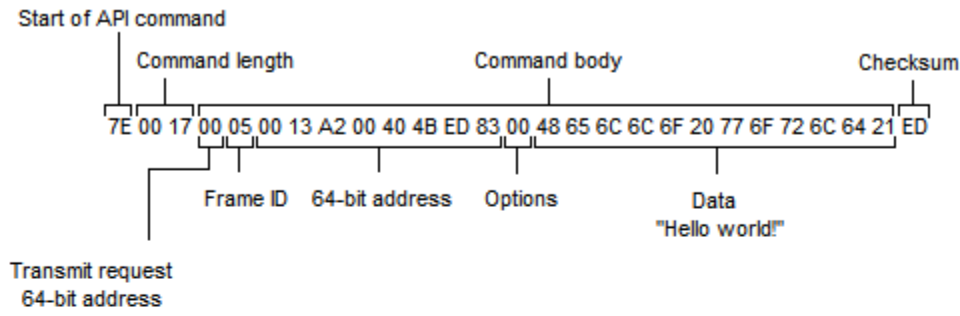
In API mode, you must use either the 64-bit address transmit request or 16-bit address transmit request to send serial data to a remote XBee module.

Notice that we are using the 0x00 API command in this example instead of the 0x17 API command used in the previous examples.

This command sends the ASCII string “Hello world!” to an external device attached to a remote XBee module using the 64-bit transmit request command:

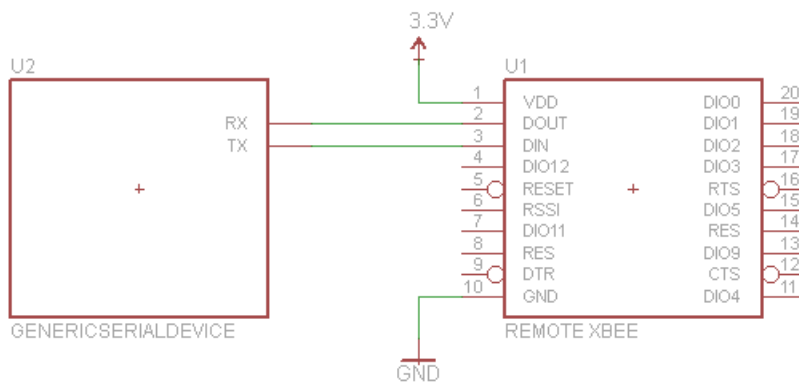
```
7E 00 17 00 05 00 13 A2 00 40 4B ED 83 00 48 65 6C 6C 6F 20 77 6F
72 6C 64 21 ED
```





Receiving serial data from an external device connected to a remote XBee module

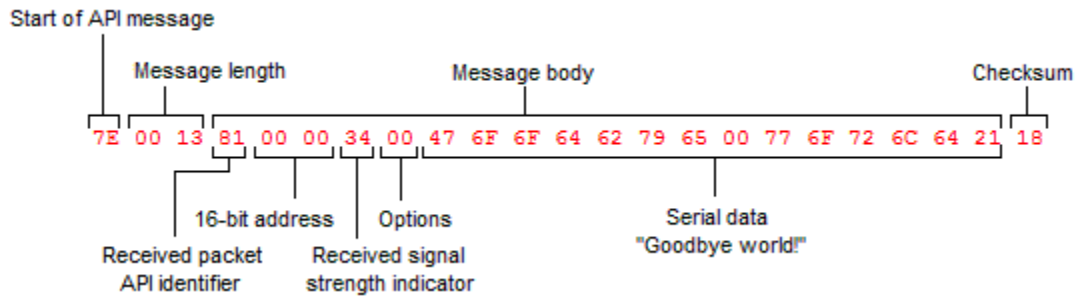
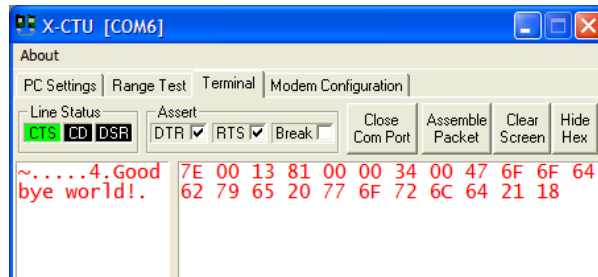
This example shows the format of a serial data message received from a remote XBee module.



Example circuit showing XBee connected to generic serial device

When the base module is in API mode, serial data received from a remote module is received as an API message.

The following example shows a message received by a base from a remote module.



References

XBee®/XBee-PRO® OEM RF Modules

http://ftp1.digi.com/support/documentation/90000982_A.pdf

“What is API (Application Programming Interface) Mode and how does it work?”

<http://www.digi.com/support/kbase/kbaseresultdetl.jsp?kb=184>

“XBee Digital Input/Output Line Passing”

<http://www.digi.com/support/kbase/kbaseresultdetl.jsp?kb=188>

“Analog to Digital Conversion on the XBee 802.15.4”

<http://www.digi.com/support/kbase/kbaseresultdetl.jsp?id=2180>

“Converting the XBee PWM to an analog voltage for DAC (Digital to Analog Conversion)”

<http://www.digi.com/support/kbase/kbaseresultdetl.jsp?id=2202>