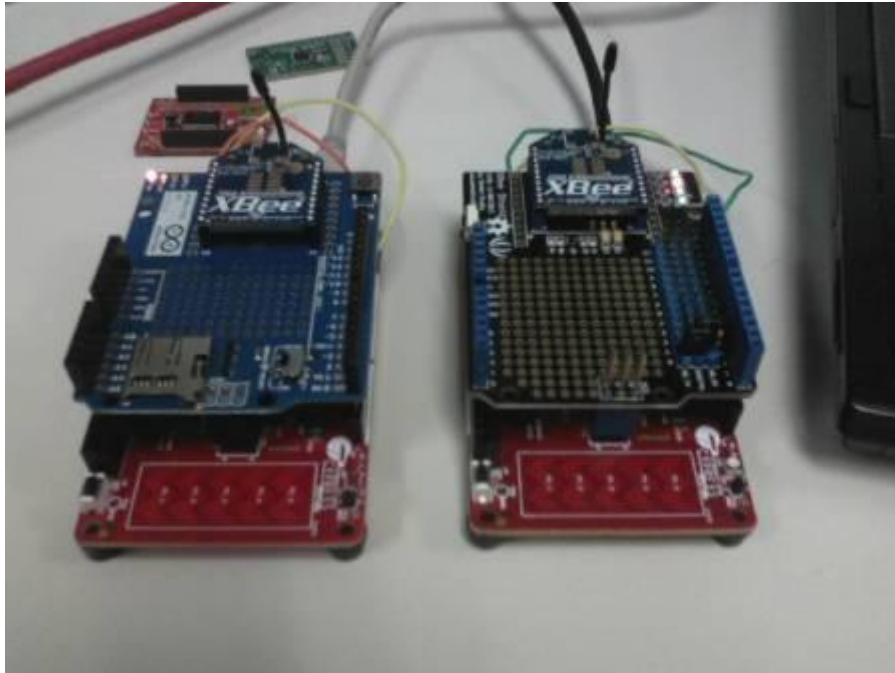


PSoC 4 Pioneer Kit Community Project#026 – XBee API RxTx Example

Today's example demonstrates how to create and receive transmission packets with the XBee modules when they are in API mode. We will set up a network between two Pioneer kits using two Arduino wireless shields and two XBee wireless modules. The following hardware will be used:

- CY8CKit-042 (x2)
- [Arduino wireless shield board](#) (x2)
- [XBee wireless module](#) (x2)



(Note: in the image above I'm using a different wireless board on the right, Seedstudio. No changes needed to the example project)

Forum Post Attachments:

At the bottom of this post we are including the following items:

- Example Project Zip File
- Zip File of Images
 - Project Schematic
 - Component Configurations

Components Used:

The user can download the example project at the bottom of this post. The project uses the following list of Creator Components:

- TX (Coordinator Example)
 - Two UARTs

- XBee_UART
 - PC_COM_UART
- Tinyprintf
- RX (Router Example)
 - Two UARTs
 - XBee_UART
 - PC_COM_UART
 - Tinyprintf

The components are configured by right clicking on the component in your Top Design schematic view and selecting **Configure**. Please enable the following selections in the Configuration windows for the listed components above.

Firmware Description:

The main.c firmware is included in the example project. Please review the commented sections for more details.

This example project requires two separate firmware projects, one for each Pioneer kit. One pioneer kit will serve as the TX module and the second pioneer kit will serve as the RX module. In the attached zip file our example project contains two example projects in the workspace explorer (RX and TX). Each project needs to be built and programmed in their respective unit. It might be helpful when examining this project to mark on your respective kit which functionality it contains (RX or TX).

As in examples #024 and #025 we will need to configure both XBee wireless modules for our example. Again I used the sparkfun explorer module and the X-CTU software to perform the configuration. Please follow the configuration examples detailed in examples #024 and #025, with the following settings:

Coordinator Settings (TX) Module:

PAND ID: 1777 (Any address from 0 to FFFF works, must match router)

Destination Address High: 0013A200

Destination Address Low: (Unique address printed below the 0013A200 on the Router module).

Once the information has been entered into the system, click the Write Button in the X-CTU GUI to configure the module.

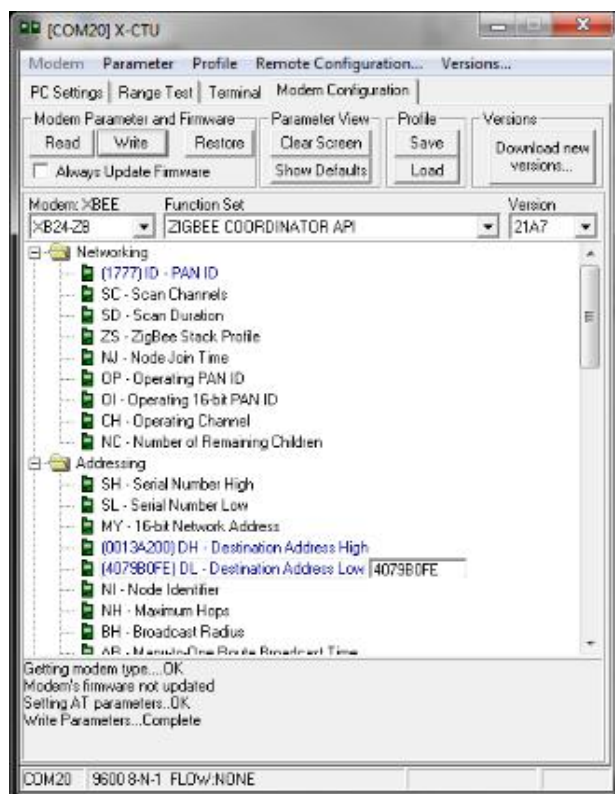
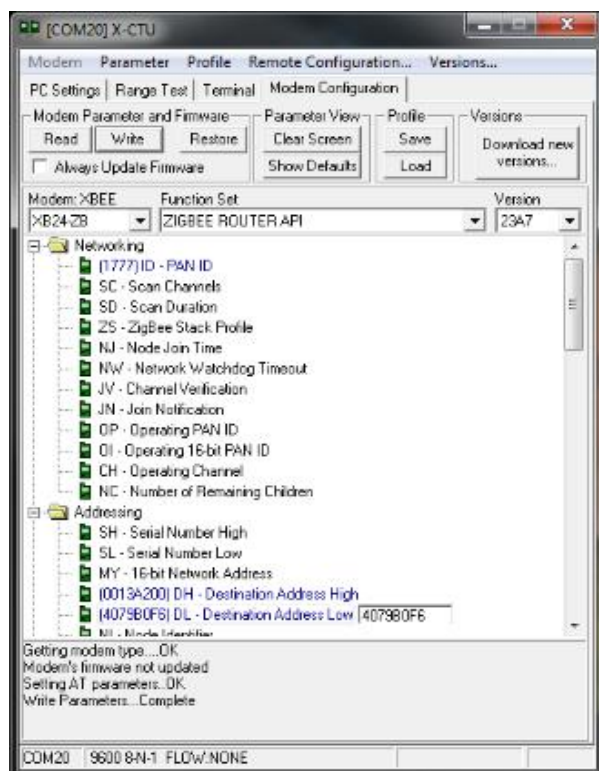
Router Settings (RX) Module:

PAND ID: 1777 (Any address from 0 to FFFF works, must match router)

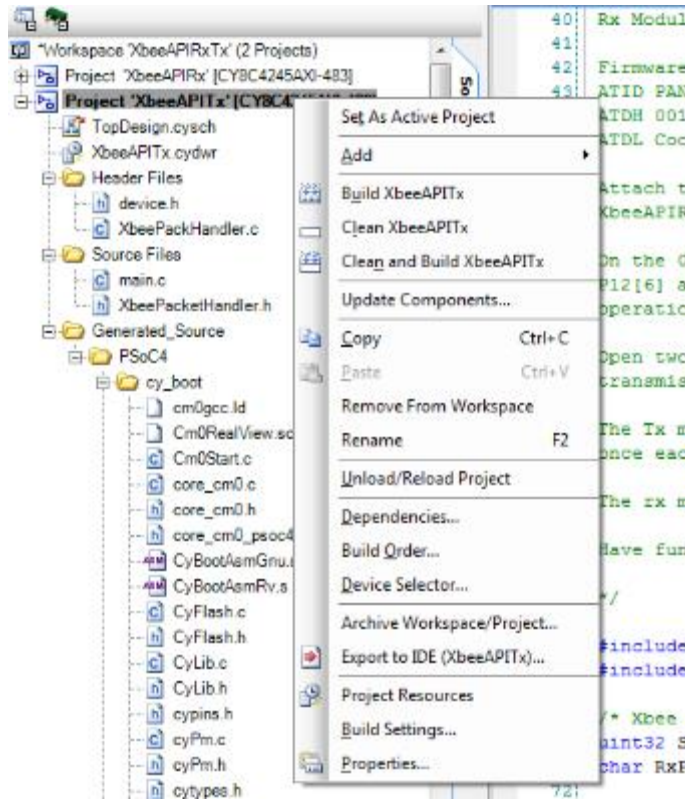
Destination Address High: 0013A200

Destination Address Low: (Unique address printed below the 0013A200 on the Coordinator module).

Once the information has been entered into the system, click the Write Button in the X-CTU GUI to configure the module.



Once the XBee modules have been configured program the respective kits with the correct project. A key point here will be to make sure you are connected to the correct 'Active Project'. You will notice in the Workspace Explorer window that there are two examples, only one **bolded**. The bold text indicates the active project. When programming the target Pioneer kit you will need to make sure you have the correct active project selected. You can do this by simply right clicking on the project and setting to 'Active'.



When programming the respective Pioneer kit from inside PSoC Creator it is easiest to ensure that only one Pioneer kit is connected to the PC at a time.

In both example projects you will notice two additional files "XbeePacketHandler.c" and "XbeePacketHandler.h". These two files are additional code modules included in this example. They expose a number of additional APIs for your use. Please review the .h file for more information on the available features. This code module is analogous to the RGB LED code module shipped with example #014.

Once the user has connected the TX kit to the hyperterminal software you will see that the TX unit is displaying two values "Delivery Status" and "DiscoveryStatus". These values are detailed in the "XbeePacketHandler.h" file. The returned value corresponds to a certain state of the radio network. Please review those values for more details.

Hardware Connections:

For this example you will need the following hardware:

1. Two Pioneer Kits
2. Two Wireless Shields
3. Two XBee wireless modules

Connect the wireless modules to the wireless shield boards. Then connect the shield board to the Pioneer kits.

Enable the PSoC 4 UART connections to the PSoC 5LP on both Pioneer kits. Using wires make the following connections on each Pioneer kit:

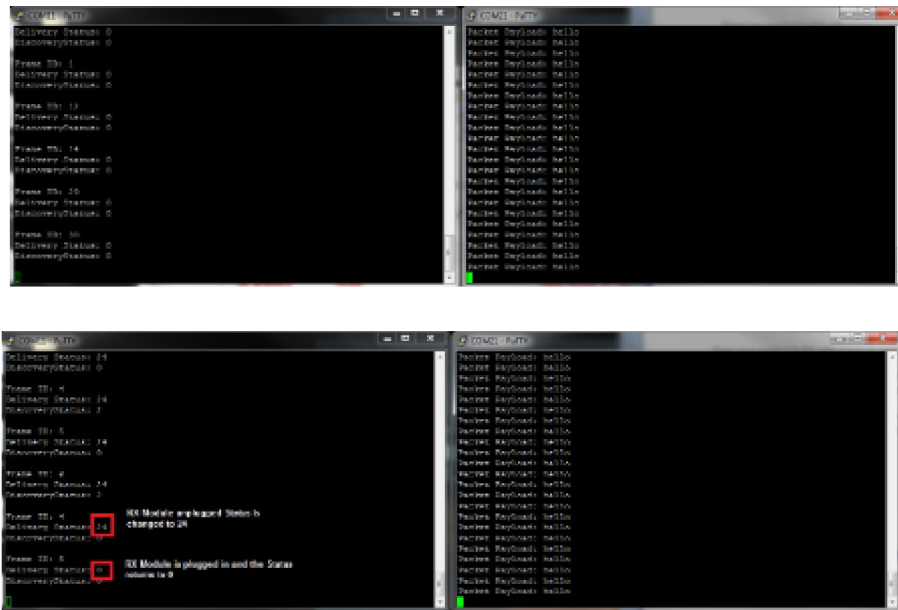
PSoC 4 **P4[1]** -> P5LP **P12[6]**

PSoC 4 **P4[0]** -> P5LP **P12[7]**

Test Your Project:

Once you have configured the wireless modules, programmed your PSoC 4s, and connect through the COM port to the Pioneer Board for each kit. You will see the hyperterminal software updating the values every second.

You can see that the 'Payload' from the TX kit has been delivered to the RX kit and then displayed on the COM port display from the RX kit.



As a final note, this is probably our most complex example. A lot of information was covered so please feel free to ask any questions in the comments below. We'll be happy to answer and clarify any aspect of the example.

I hope this example can help you out in your design.

<http://www.element14.com/community/message/79051>