



**Carleton**  
UNIVERSITY

Department of Electronics

## Laboratory 5

# Wireless Networking using ZigBee Protocols

---

### Record of Changes

REV	By	Description	Issue Date
1.0	AB	Uses Freescale K64F development board featuring Kinetis K series microcontroller, built on the ARM® Cortex®-M4 core. Interfacing with I2C temperature sensor. Uses XCTU- NG for setting up ZigBee network using XBee radio modules.	Oct, 2014
2.0	AB	Uses STM NUCLEO F401RE	Nov, 2015

## Purpose

A digital temperature sensor will be interfaced to the microcontroller using the Inter-Integrated Circuit (I2C) protocol. Moreover, it will give some experience with the fundamental aspects of wireless networking using ZigBee compatible devices. The lab will have three major parts:

1. Getting familiar with the microcontroller development board and understanding the provided I2C and UART driver code.
2. Configuring and testing the wireless network.
3. Implementing the code to read the temperature data from the sensor and transmitting it wirelessly.

## Background

Before attempting the lab, it is recommended that you understand what you are attempting to accomplish. Make sure you go through the prelab materials to familiarize yourself with the tools and devices used. Also read the “Industrial Automation” article: “Implementing ZigBee wireless mesh networking” (Implementing ZigBee wireless.pdf); the article provides background information on ZigBee with a description of some network topologies and implementation discussions.

A summary of the reading material:

- Some introduction to the ARM architecture ([http://en.wikipedia.org/wiki/ARM\\_architecture](http://en.wikipedia.org/wiki/ARM_architecture))
- User guide for, the Freescale Freedom K64 hardware, (FRDMK64FUG.pdf).
- I2C protocol (<http://en.wikipedia.org/wiki/I%C2%B2C>)
- Wireless Networking:
  - IEEE-802.15.4-2006 (this year suffix changes as the specification changes) is an IEEE standard that defines OSI Layers 1 and 2 for a low power wireless network system. Remember the 7 layer OSI specification from the lectures; this is about the physical medium and the way to receive and transmit error free frames of data.
  - ZigBee ([www.zigbee.org](http://www.zigbee.org)) defines a network protocol that sits above Layers 1 and 2 of the IEEE-802.15.4 compliant hardware and provides a way to produce low power mesh networks easily.
  - Xbee is an implementation of the ZigBee protocol by the company ‘Digi’ ([www.digi.com](http://www.digi.com)) in a small device suitable for embedded wireless applications. Digi supplies a software for called X-CTU for configuring these modules.

## More about creating ZigBee Networks

The XBee modules in the lab have two operating modes:

- AT (Transparent) mode:
  - In this mode the device acts as a simple wireless modem – Any data that is transmitted to the device via serial UART will be transmitted over the wireless link to the destination node. Also, any data that is received from the network will be made available by the device on the UART RX line.
- ‘API’ mode
  - When operating in this mode, data is transmitted as packets with a well-documented format as described by the manufacturer here (<http://www.digi.com/support/kbase/kbaseresultdetl?id=2184>) . This mode provides more information about the network (e.g. was a packet received?) and easier addressing (the destination address of the packet is included in the packet, much like internet TCP/IP packets) at the expense of extra software complexity.
  - A network can contain both API and AT mode devices. Data transmitted by a device operating in AT mode can be received by a device operating in API mode. But the reverse is not possible.

Xbee modules can also be configured with different roles in the ZigBee network, which you can change by changing the firmware inside the module –

- **ZigBee Coordinator** or ZC (there needs to be at least one in our network),
- **ZigBee Router** or ZR and,
- **ZigBee End Device** or ZED.

For this lab, each station will have two XBee modules. One module will be configured in AT mode with the “Router AT” firmware. Another one will be configured in API mode with the Coordinator API” firmware.

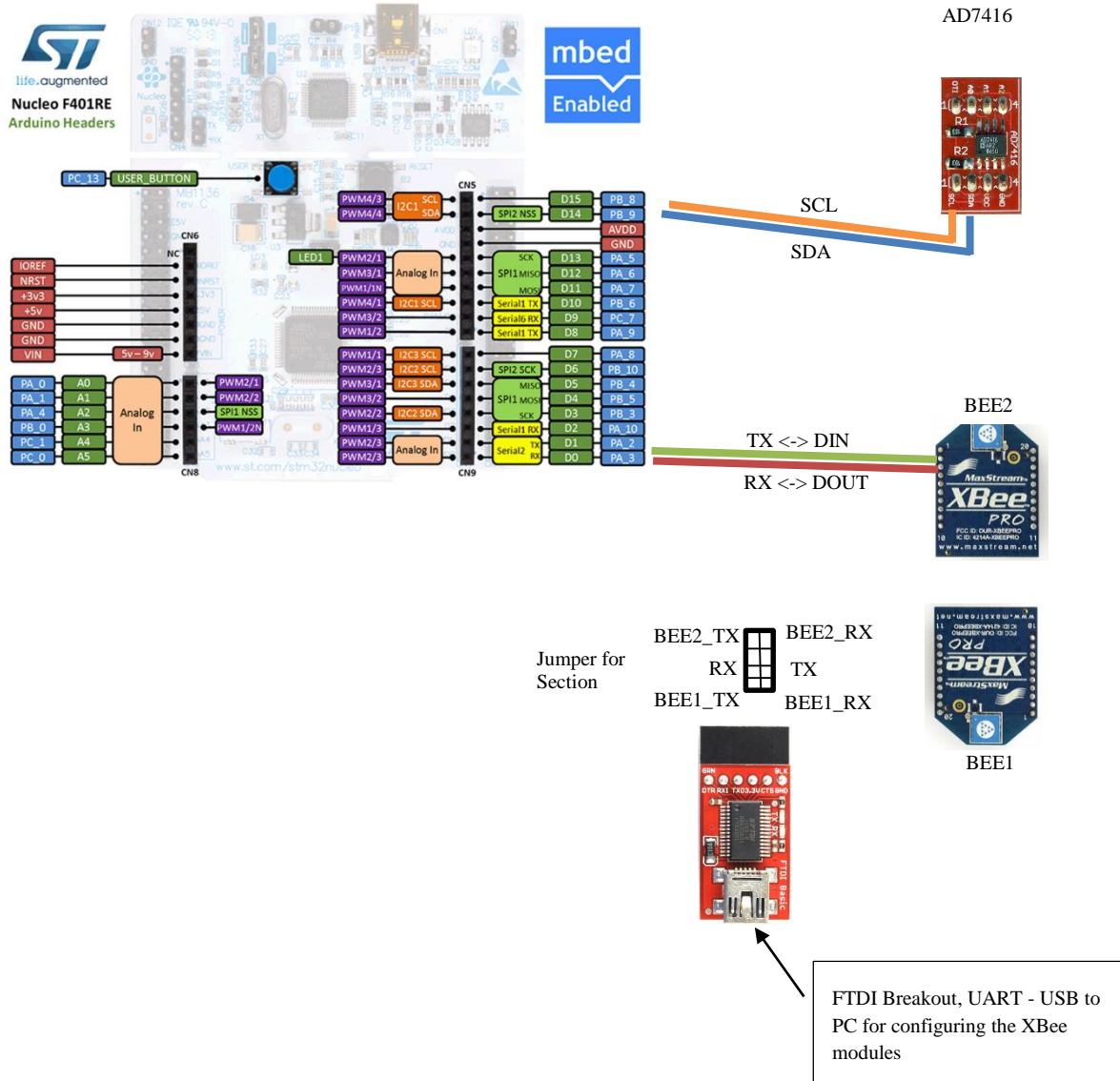
## Equipment

You will be provided with the following:

- STM NUCLEO FR401 development board connected with two XBee Pro Series-2 ZigBee devices
- A temperature sensor (AD7416) interfaced to the development board on I2C bus.
- A Windows PC with the following software installed
  - Keil uVision 5
  - Digi XCTU-NG software for configuring the XBee modules.

- Template code

A block diagram of the setup is provided below:



## Procedures and measurements

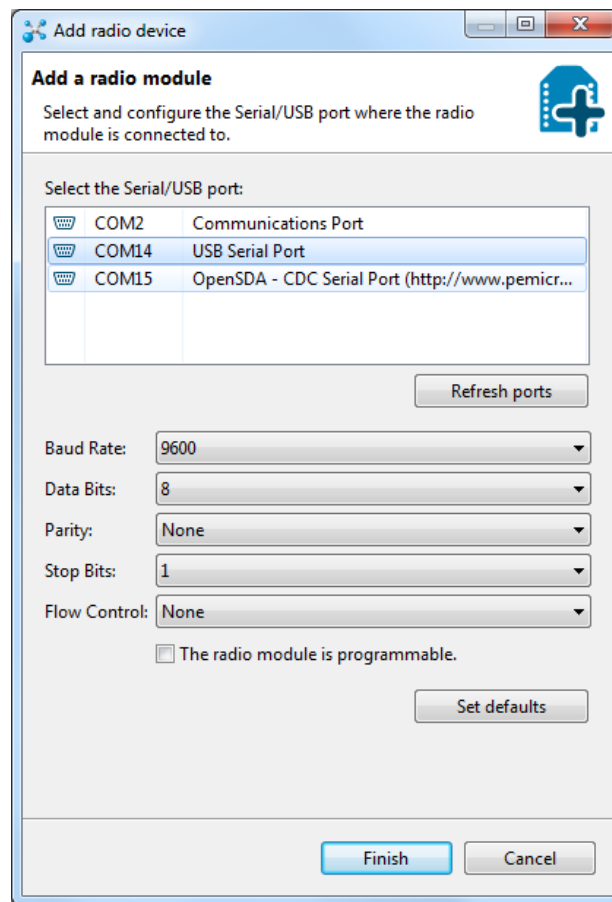
### Part 1: Setting up the ZigBee Network

#### Procedure

Step1: Make sure the jumpers on the development board are set to BEE1.

Find and Open XCTU 

Step2: Click on the “Add Radio module” button, in the selection wizard choose “USB Serial Port”, then click “Finish”.



Step3: Once the radio device is added successfully. Select the device from the list and click on the settings icon on the top right corner.

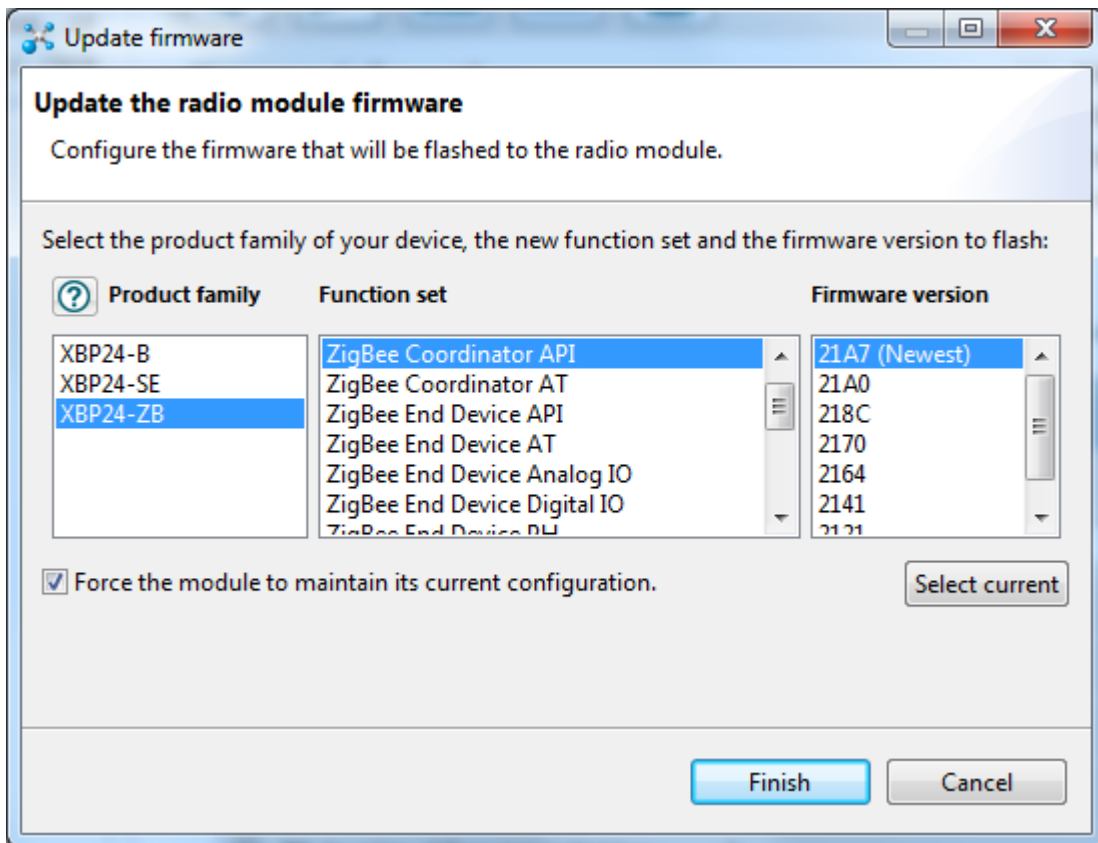


Now, you should be able to see the configuration parameters for the device. Check to see if the firmware on the device is “Zigbee Coordinator API” (You should see the function listed under the device name in “Radio List” window) . If it is, skip to Step5. If not, follow the procedure on Step4 to load the Coordinator API firmware.

Step4: Click on the “Update Firmware” Icon.










Select the settings shown below carefully. And then click “Finish” Wait for few seconds for the new firmware to be uploaded on the device.



Step5: In this step you will need to set the PAN (Personal Area Network) ID for the network. Use the student number of one of your group members.





Radio Configuration [COORD\_P15 - 0013A2004060C6C0]










Parameter  

**Firmware information**

Product family: XBP24-ZB  
 Function set: ZigBee Coordinator API  
 Firmware version: 21A7

 Written and default  
 Written and not default  
 Changed but not written  
 Error in setting

▼ **Networking**  
Change networking settings

 ID PAN ID	100874332		
 SC Scan Channels	1FFE Bitfield		
 SD Scan Duration	3 exponent		

Click on the red button on the right to write the settings on the device.

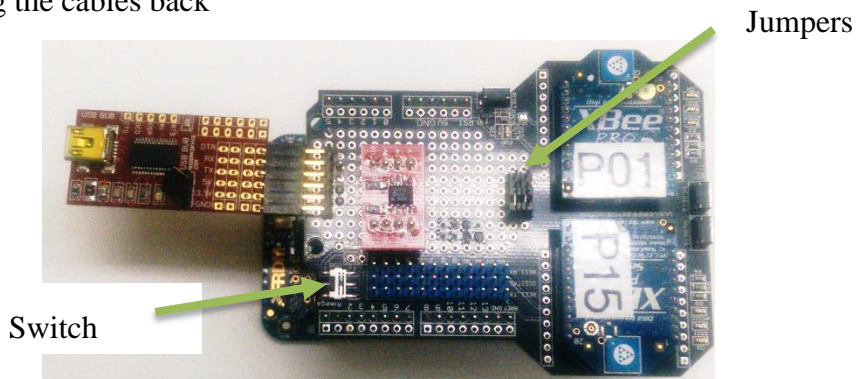
Step6: Note down the SH and SL parameters for the device.

▼ **Addressing**  
Change addressing settings

 SH Serial Number High	13A200	
 SL Serial Number Low	4060C6C0	

Step7: If you made it to this step, by now you have set up the “Coordinator” on the ZigBee network. The next step is to setup the other XBee (BEE2) which is connected to the MCU to join the network. Before we continue, you need to make some adjustment to the circuit. This is very important.

- First unplug the two USB cables to power off the board
- Change the position of the two jumpers to “BEE2”
- Make sure the switch at the bottom left corner of the board is set to middle position.
- Plug the cables back

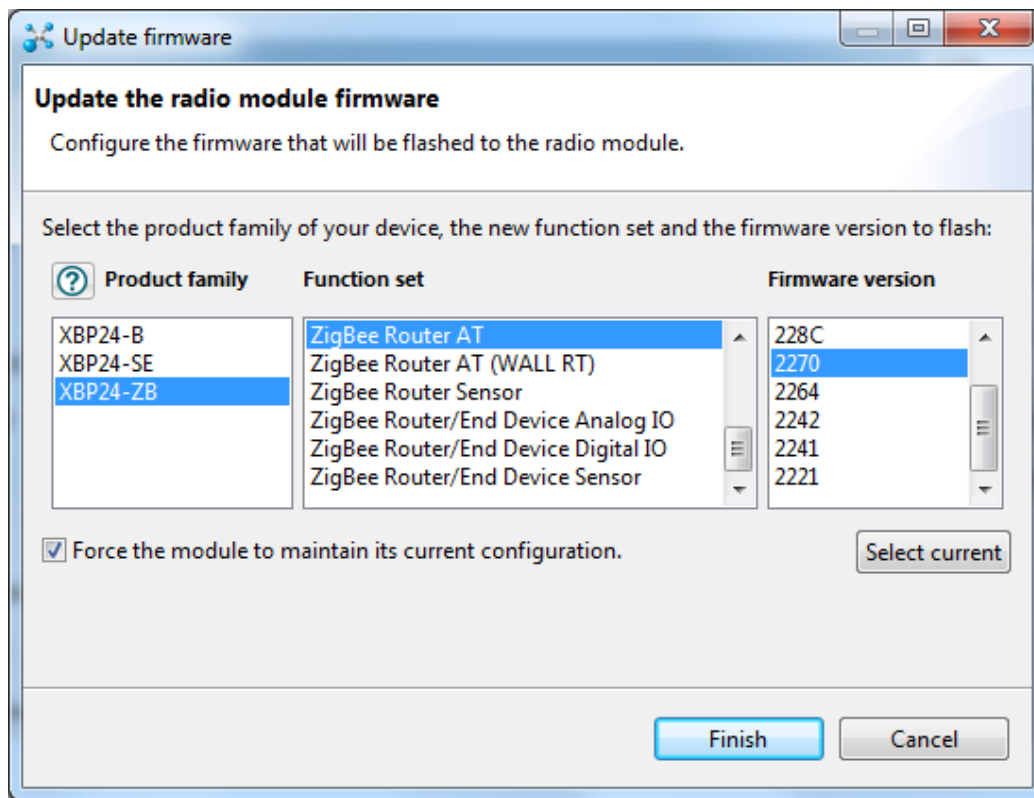


Step8: In XCTU, click on the “x” icon to remove the first XBee that you configured. Follow “Step2” to add the new device.

Step9: Once the new radio device is added successfully. Select the device from the list and click on the settings icon on the top right corner.

Now, you should be able to see the configuration parameters for the device. Check to see if the firmware on the device is “Zigbee Router AT”. If it is, skip to Step11. If not, follow the procedure on Step10 to load the “Router AT” firmware.

Step10: Click on the “Update Firmware” Icon. Select the settings shown below carefully. And then click “Finish”. Wait for few seconds for the new firmware to be uploaded on the device.



Step11: You have now put the “Router AT” firmware on the second XBee. Now, you will need to set the PAN ID parameter to make it join the network setup by the “Coordinator”. (Hint: Make sure you use the same PAN ID). Also set the destination address high (DH) and destination address low (DL) parameters to similar to SH and SL parameters of the coordinator you recorded in Step6. What happens if you leave these to their default value?

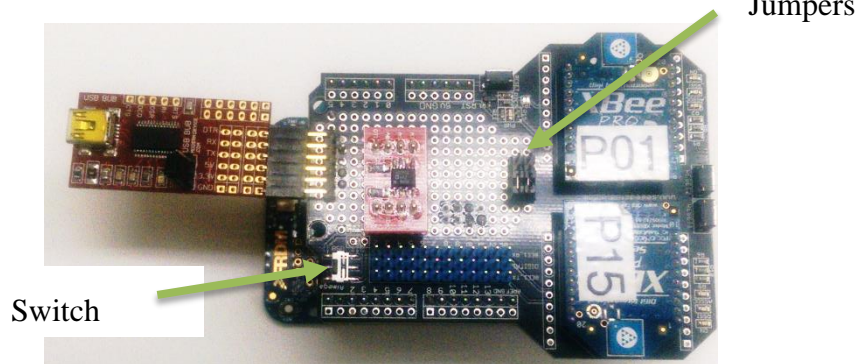
Step12: Click on the “Write Settings” icon to save the parameters.





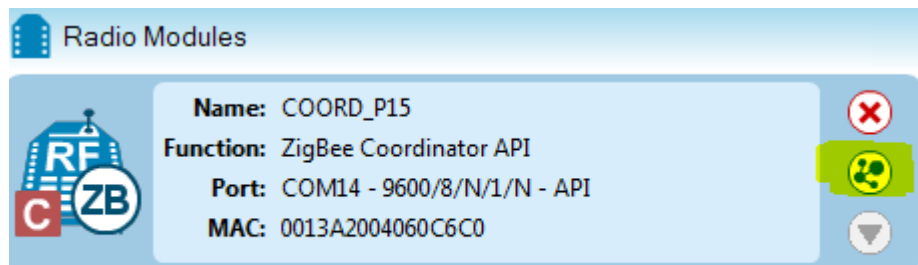
Step 13: If you made it to this step, by now you have set up the “Router” to join the ZigBee network set up by the “Coordinator”. Before we continue, put the circuit back to the original configuration:

- First unplug the two USB cables to power off the board
- Change the position of the two jumpers to “BEE1”
- Make sure the switch at the bottom left corner of the board is set to “Atmega” position. (Ignore the name for the position of the switch, it has nothing to do with this lab)
- Plug the cables back



Step14: In XCTU remove the currently listed device. And then Follow Step2 and Step 3 to add the “Coordinator” XBee.

Step15: Click on the “Discover radio node in the same network” icon.



Step16: If you can see the “Router” XBee listed under discovered devices, congratulations, you have successfully setup the ZigBee network. If not, go back to Step 2 to try this part again.

Note: After you click on “Add selected devices to add the discovered XBee to the network, you should now be able to configure the device “Over the air” without connecting to it directly through the USB.

## Part 2: Acquiring and transmitting data from the temperature sensor

### Procedure

In this part, you will need to implement the code to read the data from the temperature sensor using I2C and send it to the “XBee” (BEE2) using UART. Hint: Use the mbed I2C and UART libraries.

- Set the jumper positions to “BEE2”
- Switch position to “Atmega”

PIN	Function
PB_9	SDA
PB_8	SCL
PA_2	TX
PA_3	RX

## Part 3: Printing the temperature on the LCD (Bonus)

Here is the LCD wiring:

PIN	LCD Function
PA_10	D7
PB_3	D6
PB_5	D5
PB_4	D4
PC_7	RS
PA_9	EN
GND	R/W

## Prelab

1. Read the materials presented in the Background section.
2. Briefly explain how the I2C protocol works. How many wires are needed? What is the function of each signal? How does a master device communicate with a slave device? Use a timing diagram.
3. How many I2C modules are supported by the microcontroller?
4. How many UART modules are supported by the microcontroller?
5. Read the datasheet for the temperature sensor “AD7416.pdf”. In the lab board, the address bits A0-A2 are all pulled high to VCC. What will be the 8 bit I2C slave address for the sensor? What is the register address for reading the temperature data?