PiKoder/SSC Application Note #3:

XBee Communication

Version 1.2
Dated 09/10/2019

Gregor Schlechtriem

Content

1

Overview

The PiKoder/SSC can be controlled through a transparent serial link such as XBee, which became recently very popular in the IoTS arena. This setup provides for a simple but elegant remote control for e.g. robots.

The XBee RF modules can be programmed and provide for a high level of flexibility. Therefore, the first step would be to verify the parameters of the XBee modules at hand to make sure that communication between the radios can be established automatically upon powering up (called "transparent mode"). This process is laid out in section 2.

Section 3 of this document focusses on the hardware setup for the receiving unit comprised of a PiKoder/SSC and an XBee RF module including the beak-out boards required. The following section 4 describes how to establish the Bluetooth connection between the PiKoder/SSC and a PC running Windows 7 (with an XBee USB adapter).

Please check always for updated information and new software releases on www.pikoder.com before you start.

Also, please share with me any comments, improvement ideas or errors you will find or encounter in working with your PiKoder/SSC. I can be reached at webmaster@pikoder.com. Thank you very much!

Room for your notes

2

Configuring your XBee modules

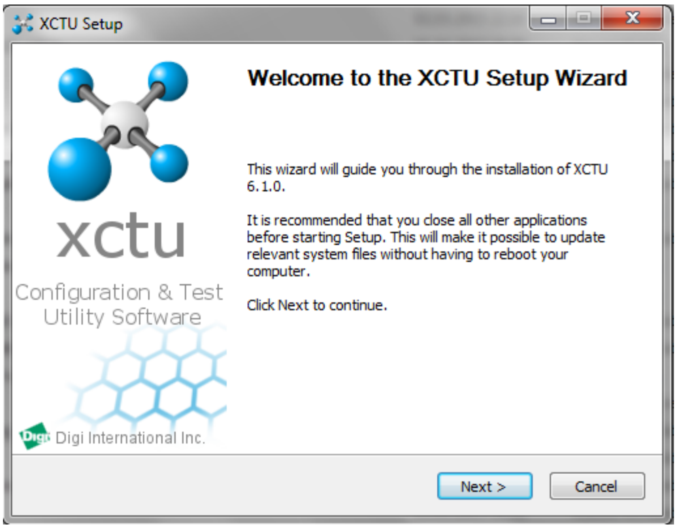
Throughout this application note the PiKoder/SSC will be controlled by a transparent serial communication link. This serial link is established between two XBee modules in the so called "transparent mode" or "AT-mode". This section outlines which parameter settings for the XBee RF modules are required to make sure that the communication is started once the modules are powered up.

According to the XBee specifications each communication link needs a Coordinator setting up the network. This means, that you would have to program one of the radios as a communicator. The second radio will be customized as an end device. To avoid interference with other XBee networks, a specific network id and the destination addresses are set.

For configuring the modules you will need an additional XBee USB adapter to connect the RF modules to your PC. Please note that this application is based on XBee Series 2 modules. Although the XBee modules can be programmed in terminal mode it is recommend to use a readily available tool, the "XCTU" which can be downloaded for free from www.digi.com.

Installing and starting "XCTU"

The "XCTU"-tool downloads as a self-installing application. Start the application and a Setup Wizard will guide you through the process. You can accept the default settings.



Welcome to the XCTU Setup Wizard

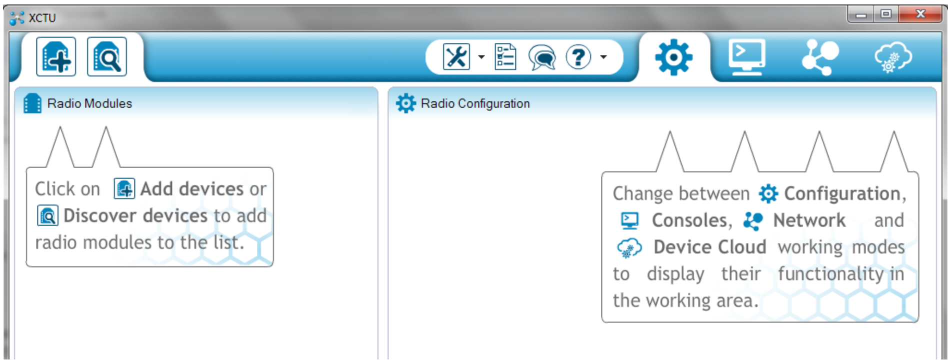
This wizard will guide you through the installation of XCTU 6.1.0.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Programming the XBee radios

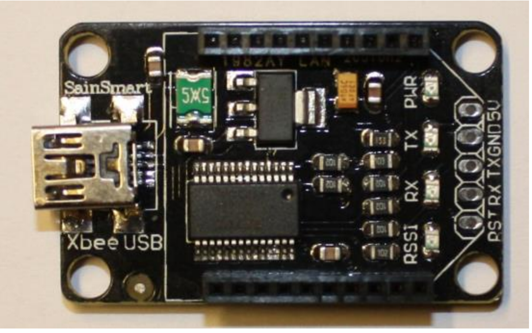
After completing the installation start the application, close the Change Log Window and you will see the following screen.



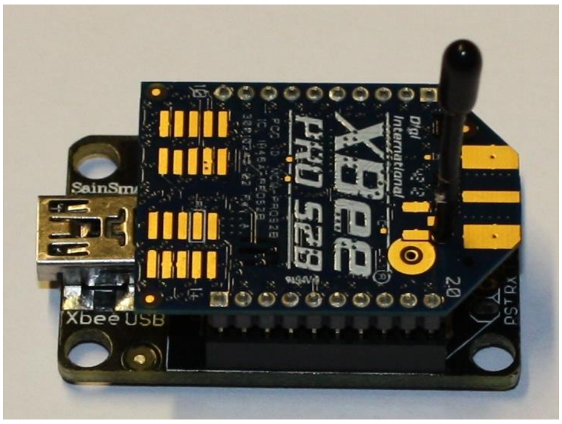
Click on Add devices or Discover devices to add radio modules to the list.

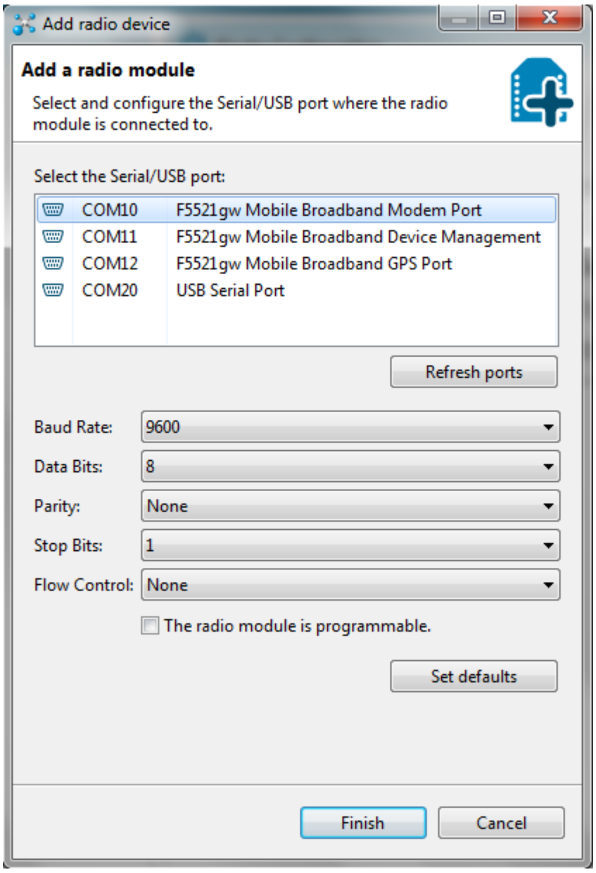
Change between Configuration, Consoles, Network and Device Cloud working modes to display their functionality in the working area.

In order to add devices you need to connect your XBee USB adapter to your computer. Windows will automatically install USB driver software such as an FT243 USB UART driver.



Then you would insert your XBee radio module into the connector of the adapter as shown below. After clicking the left most menu button with a "+" to add the first radio, a new window would appear allowing you to select the serial port to be used (see image on next page – in this instance it would be COM20). Select the port.





Add a radio module

Select and configure the Serial/USB port where the radio module is connected to.

Select the Serial/USB port:

COM10	F5521gw Mobile Broadband Modem Port
COM11	F5521gw Mobile Broadband Device Management
COM12	F5521gw Mobile Broadband GPS Port
COM20	USB Serial Port

Baud Rate: 9600

Data Bits: 8

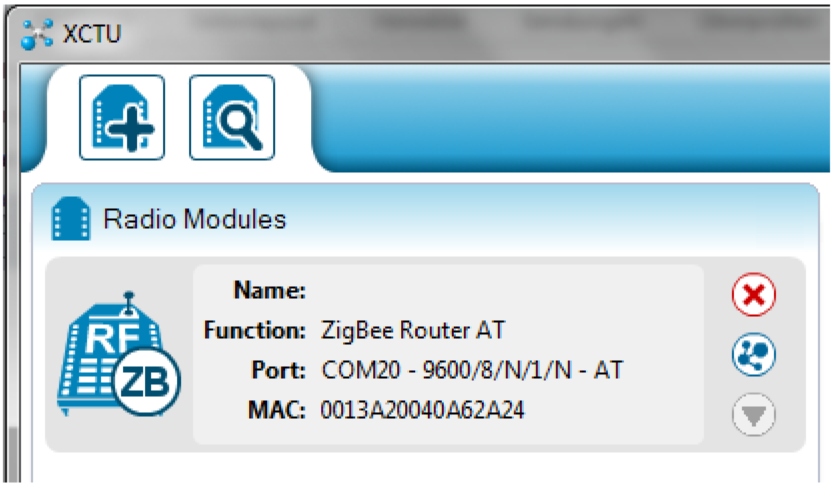
Parity: None

Stop Bits: 1

Flow Control: None

□ The radio module is programmable.

The module plugged into the USB adapter would be detected and added to the list of radios.



Name:

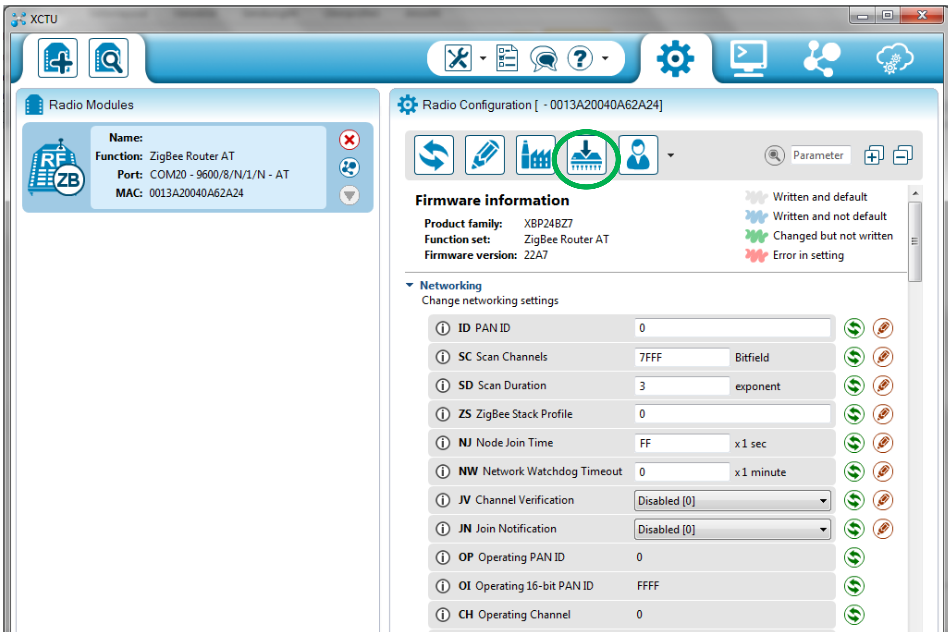
Function: ZigBee Router AT

Port: COM20 - 9600/8/N/1/N - AT

MAC: 0013A20040A62A24

After clicking the module "XCTU" will retrieve the radio parameters and display them in the right hand part of the application screen as shown on the next page. Assuming your XBee radios are freshly out-of-the box you don't have to bother too much with all the parameters.

For setting up the required "transparent-" or "AT-" mode you would have to re-program one of your modules firmware to the Coordinator AT firmware. You would select the firmware update button circled in green on the screen print on the next page.


XCTU

Radio Modules

Name:
Function: ZigBee Router AT
Port: COM20 - 9600/8/N/1/N - AT
MAC: 0013A20040A62A24

Radio Configuration [- 0013A20040A62A24]

Parameter

Firmware information

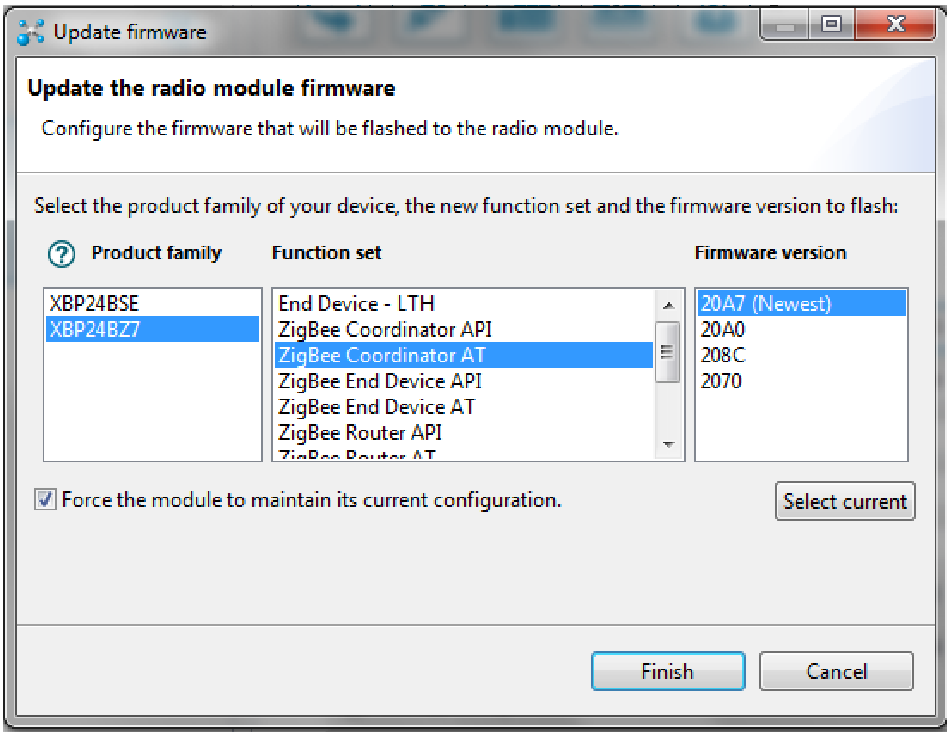
Product family: XBP24BZ7
Function set: ZigBee Router AT
Firmware version: 22A7

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

Networking
Change networking settings

ID PAN ID	0	
SC Scan Channels	7FFF	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	
NJ Node Join Time	FF	x 1 sec
NW Network Watchdog Timeout	0	x 1 minute
JV Channel Verification	Disabled [0]	
JN Join Notification	Disabled [0]	
OP Operating PAN ID	0	
OI Operating 16-bit PAN ID	FFFF	
CH Operating Channel	0	

From the drop box select the firmware for the ZigBee Coordinator AT and program your radio.


Update firmware

Update the radio module firmware

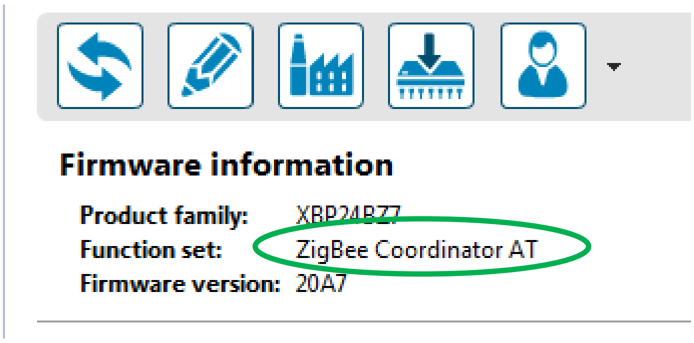
Configure the firmware that will be flashed to the radio module.

Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
XBP24BSE	End Device - LTH	20A7 (Newest)
XBP24BZ7	ZigBee Coordinator API	20A0
	ZigBee Coordinator AT	208C
	ZigBee End Device API	2070
	ZigBee End Device AT	
	ZigBee Router API	
	ZigBee Router AT	

☑ Force the module to maintain its current configuration.

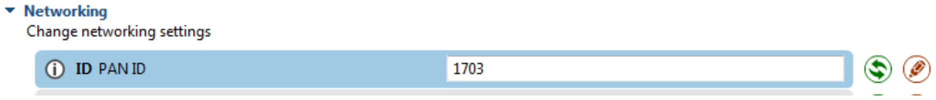
Once the re-programming is completed you will see the revised function set of your controller (see green circle below).



Firmware information

Product family: XBP24BZ7
Function set: ZigBee Coordinator AT
Firmware version: 20A7

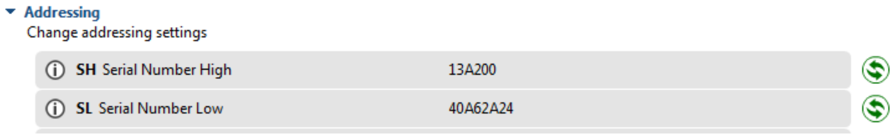
Within the parameter window set a preferable PAN ID, I simply set “1703”. You can also set the Node ID, which I set to “COORDINATOR_01”; this is not critical though. Please note that the parameter is stored into the radio only after clicking the red pen.



Networking
Change networking settings

ID PAN ID	1703

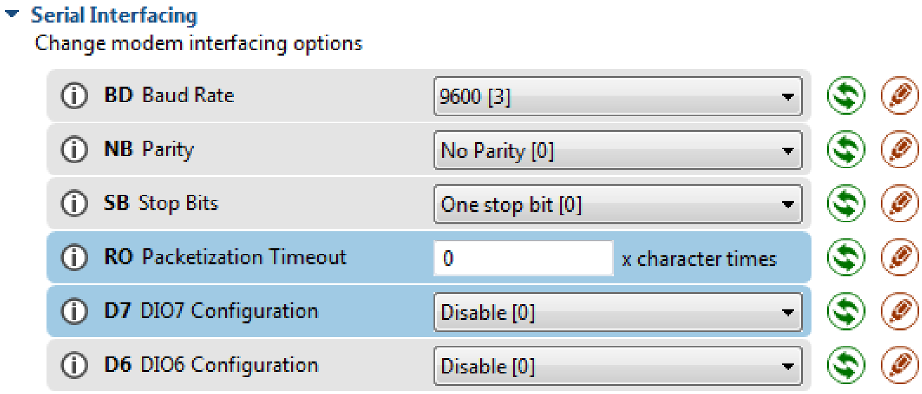
Record the SH and SL ID on this XBee S2 module, you will need it to set up the end device module. In this case, the SH is 13A200, SL is 40A62A24. This is Coordinator Source Address.



Addressing
Change addressing settings

SH Serial Number High	13A200
SL Serial Number Low	40A62A24

In order to simplify the UART communication you would also set the parameter RO (Packetization Timeout) to “0” to make sure that characters which are received are transmitted right away. Also, disable the flow control for D7. The required configuration is shown below.



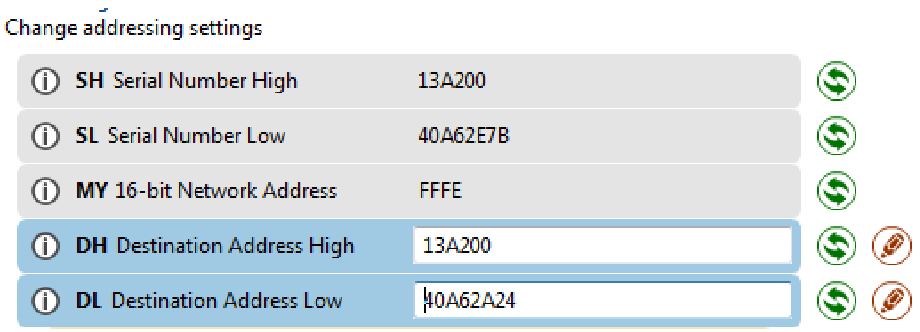
Serial Interfacing
Change modem interfacing options

BD Baud Rate	9600 [3]
NB Parity	No Parity [0]
SB Stop Bits	One stop bit [0]
RO Packetization Timeout	0 x character times
D7 DIO7 Configuration	Disable [0]
D6 DIO6 Configuration	Disable [0]

Now you would have to program the end device. Replace the radio you just programmed with the second unprogrammed radio and follow the steps described earlier to connect to the XBee module.

Reprogram the firmware of this module to an End Device AT and set the PAN ID to the same value assigned to the coordinator module (“1703” in this example) and make sure the serial settings show above are also set for this radio.

Finally, you would have to set the destination address to the Source Address of the coordinator as shown below:



Change addressing settings

ⓘ	SH Serial Number High	13A200
ⓘ	SL Serial Number Low	40A62E7B
ⓘ	MY 16-bit Network Address	FFFE
ⓘ	DH Destination Address High	13A200
ⓘ	DL Destination Address Low	40A62A24

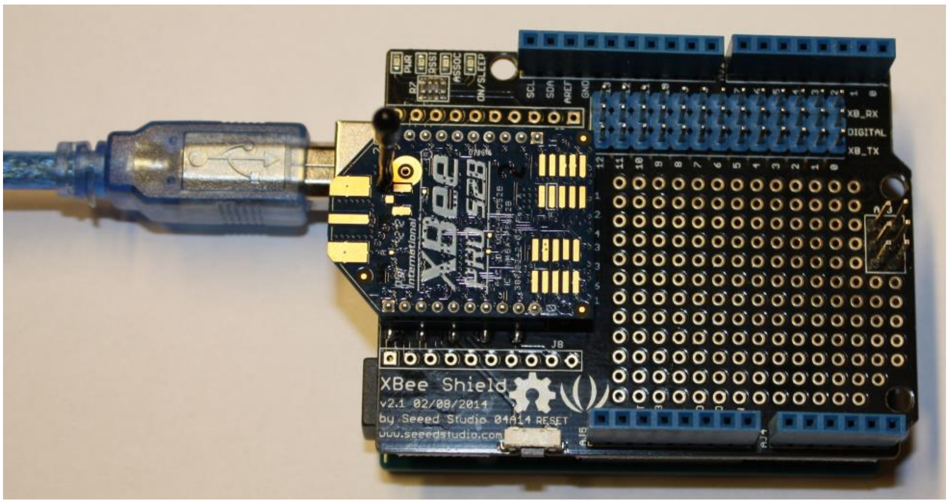
Take a note of the serial number high and low of this module because you have to put these numbers into the respective destination fields of the coordinator.

Once you completed this final step by swapping the radios again and entering the address values into the coordinator radio, the modules are set up and can be tested.

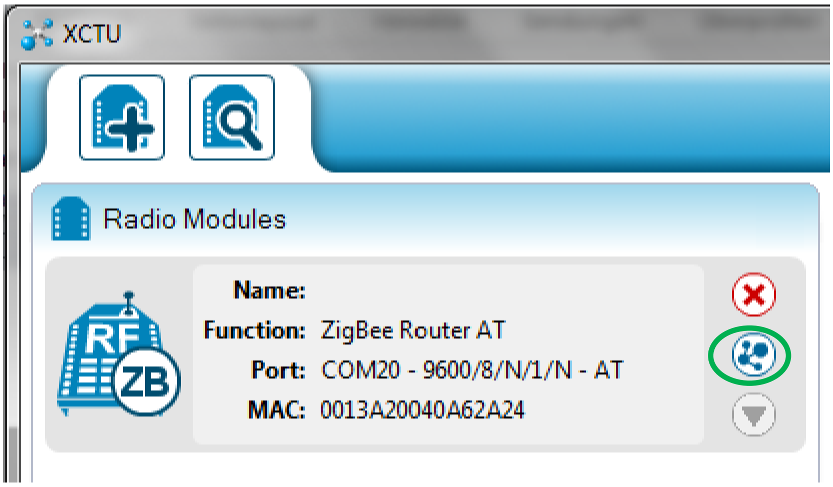
Testing the connection

Now your radios should be programmed to automatically establish a connection once they are powered up. The following section describes an optional verification test of the configuration.

For this test you would use your USB XBee adapter to connect to the first radio and deploy a second adapter to power up and connect your second radio. You could also use an Arduino XBee shield as shown below. Please note that there are no jumpers to connect serial lines – the Arduino is used solely as a power source for this test.



After hooking up the hardware and connecting to the first radio by selecting the appropriate COM port, click the Search function (circled green below) to test the wireless connectivity.



Radio Modules

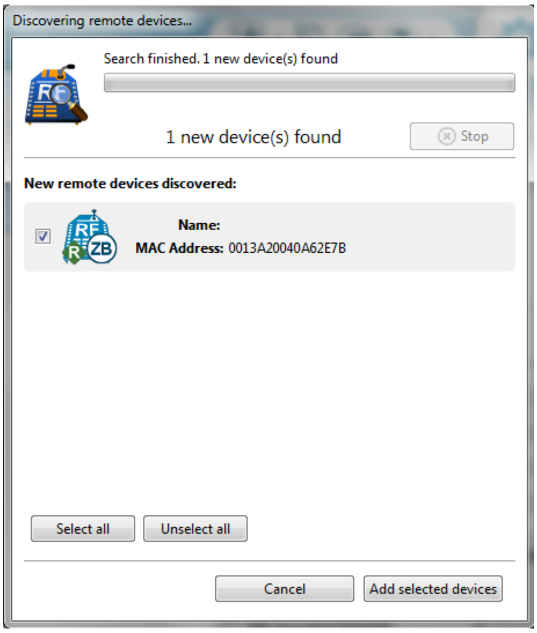
Name:

Function: ZigBee Router AT

Port: COM20 - 9600/8/N/1/N - AT

MAC: 0013A20040A62A24

If your programming was successful then the second XBee module will be found indicating that both modules are in the same network and did recognize each other (see next page).


Discovering remote devices...

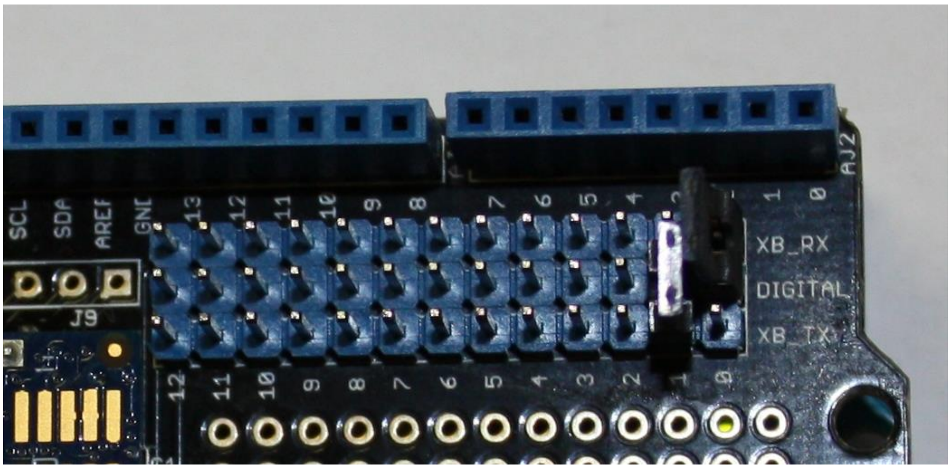
Search finished. 1 new device(s) found

1 new device(s) found

New remote devices discovered:

☑ Name:
MAC Address: 0013A20040A62E7B

Finally, you can test the data transmission between the two radios. First you would have to set the jumpers of the XBee shield according to the scheme below. This would allow you to communicate directly with the radio's UART without employing the Arduino.



To avoid that any Arduino serial activity would interfere with the XBee radio communication you would have to connect your Arduino's RESET pin with GND.

For the final testing the connection would be established through two Tera Term windows. Start Tera Term twice and connect one instance to the port for the XBee coordinator and the other instance to the end device port. Please activate the local echo function also (check respective box in "Setup\Terminal").

Then go to to the terminal connected to the coordinator and start to type in text. This text should appear right away in the terminal connected to the end device.

After completing this test, you are ready to build the hardware for the actual receiver.

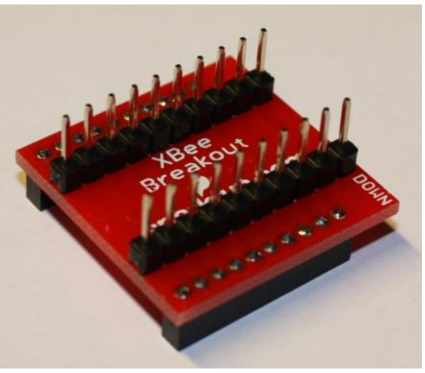
3

Receiver unit

For the serial link receiving unit shown below you will need a PiKoder/SSC, a break out board for the XBee radio, an adapter-pcb, and the XBee radio end device, which you programmed according to the previous section.

Building the break out board for the Xbee radio

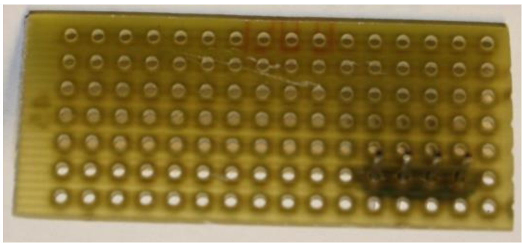
The XBee radios have 2 mm sockets and for interfacing with a standard prototype pcb you will need a break out board breaking out all 20 pins of the XBee to a 0.1" standard spacing dual row header. For my receiver unit I used the sparkfun “Breakout Board for XBee Module”. I used female sockets to avoid having to solder the XBee permanently to the breakout board. The completed board is shown below.

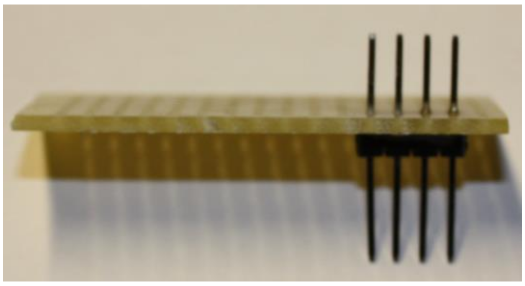


Building the adapter board for the PiKoder

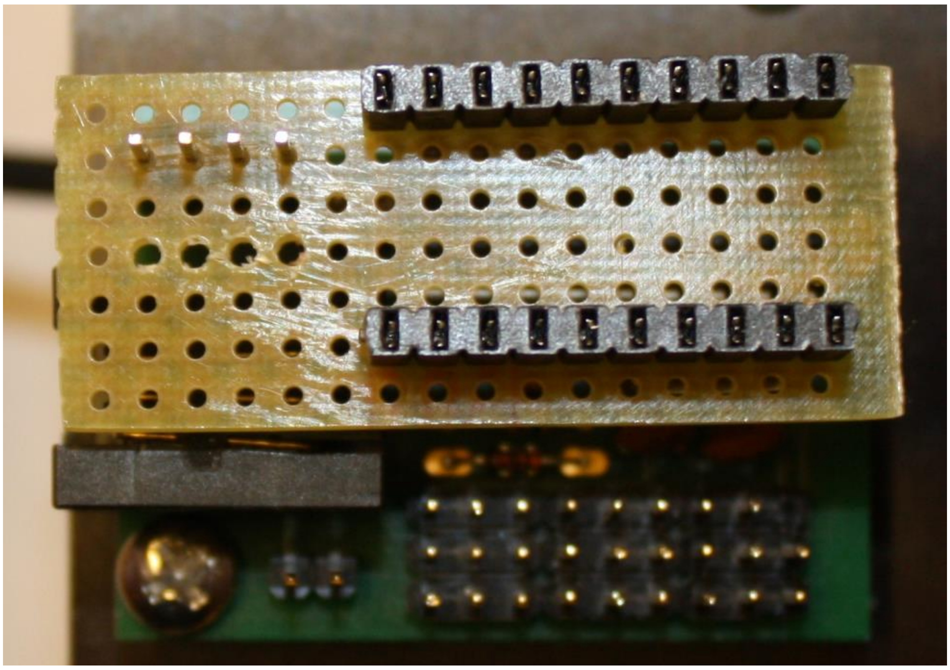
Another adapter PCB is needed to interface the xbee breakout board to the PiKoder/SSC. A standard prototype PCB will suffice for this adapter.

Start of building with cutting a smaller piece with 7 x 16 holes from a larger PCB. Then you have to glue in a four pin male – male header to connect to the PiKoder's female input connector. Please refer to the following photos for the correct position.

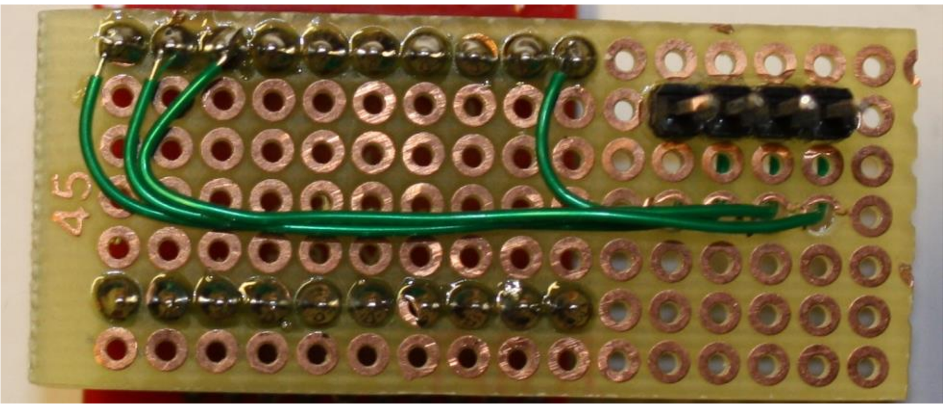




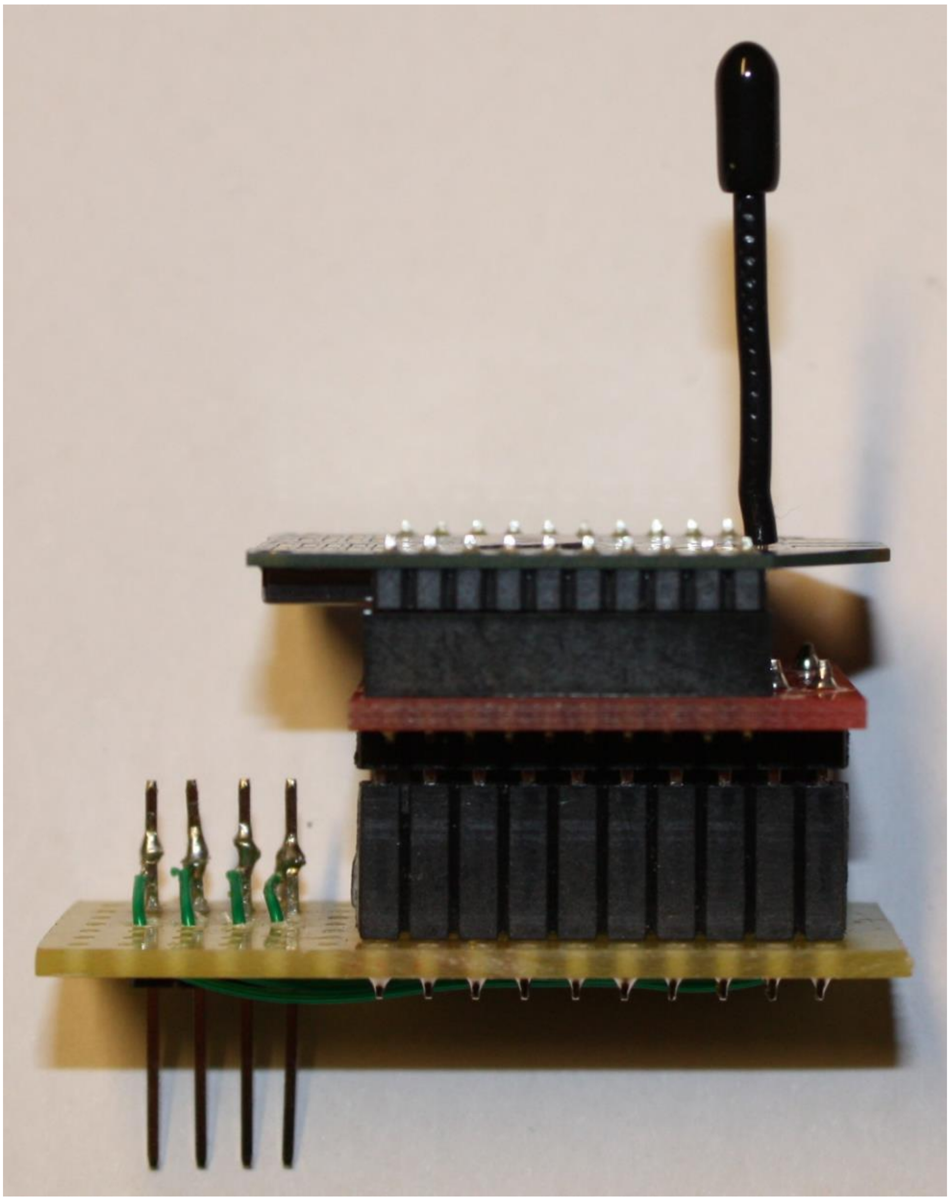
The next step would be to place the 10 pin female headers for accepting the XBee breakout male connectors in the final setup. Please refer to the next photo for correct positioning.



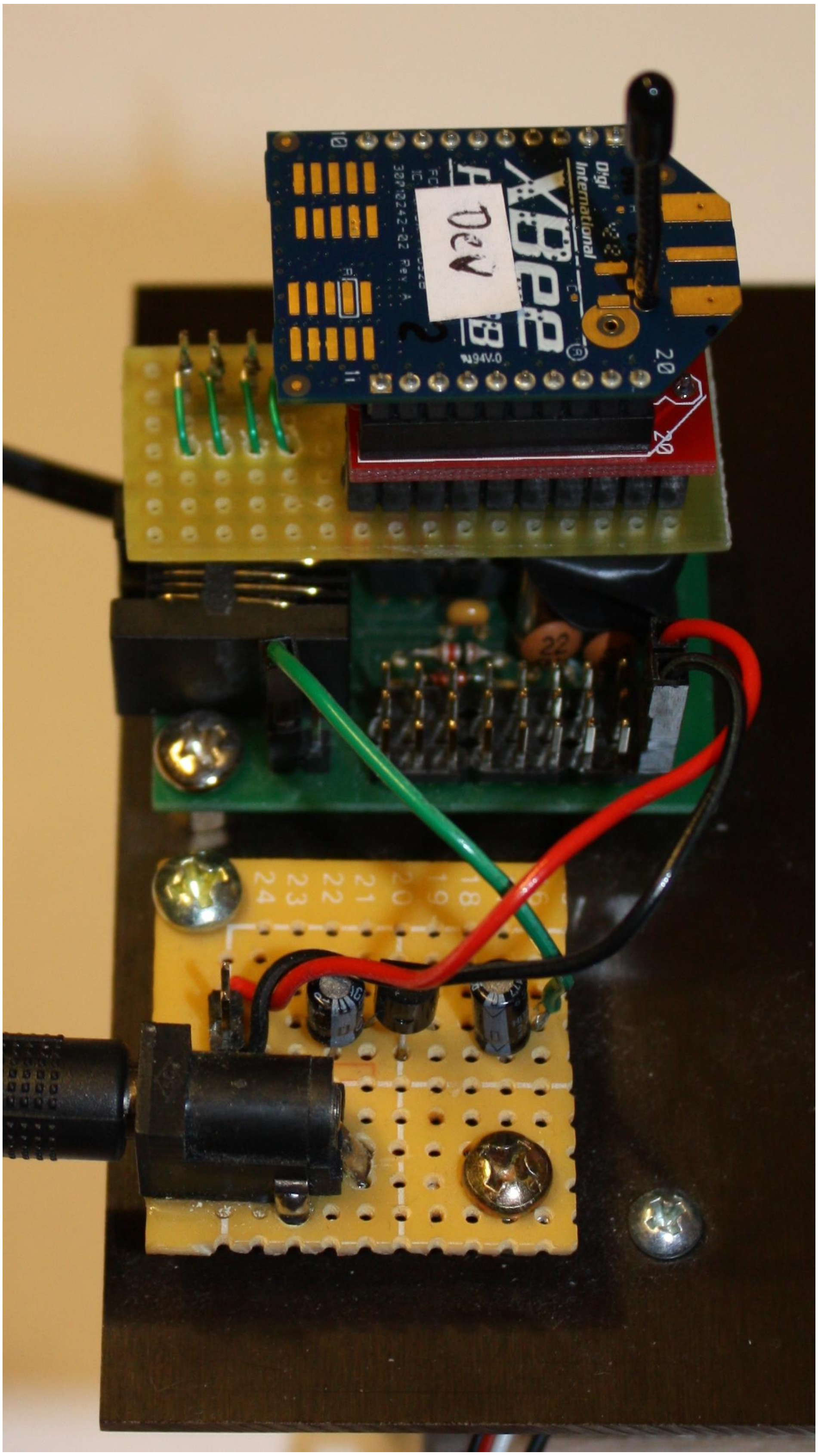
Finally, for completing this adapter, you would have to wire the four male pins to the respective female connectors as shown on the next page.



Now the boards can be stacked up as shown below.

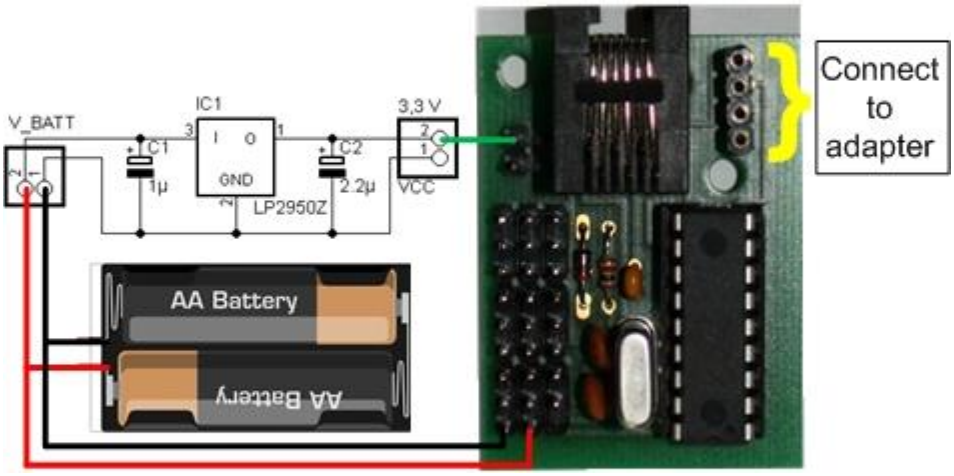


This unit then can be stacked onto the PiKoder/SSC as shown below to form the complete receiver unit. Please note that the radio which has been programmed as a device is used for the receiving unit.



Since the XBee module requires an operating voltage of 3.3 V max. You would have to use a voltage converter to reduce your local (robot-) battery voltage to 3.3 V. Since the PiKoder/SSC would also operate on this lower voltage the complete receiving unit would be connected to the converter. The wiring scheme is shown below.

The voltage converter uses an integrated low drop voltage converter LP2950 and two capacitors. This simple setup can be built easily on a prototype board as shown above.



PiKoder/SSC voltage converter and wiring for XBee receiving unit

Room for your notes

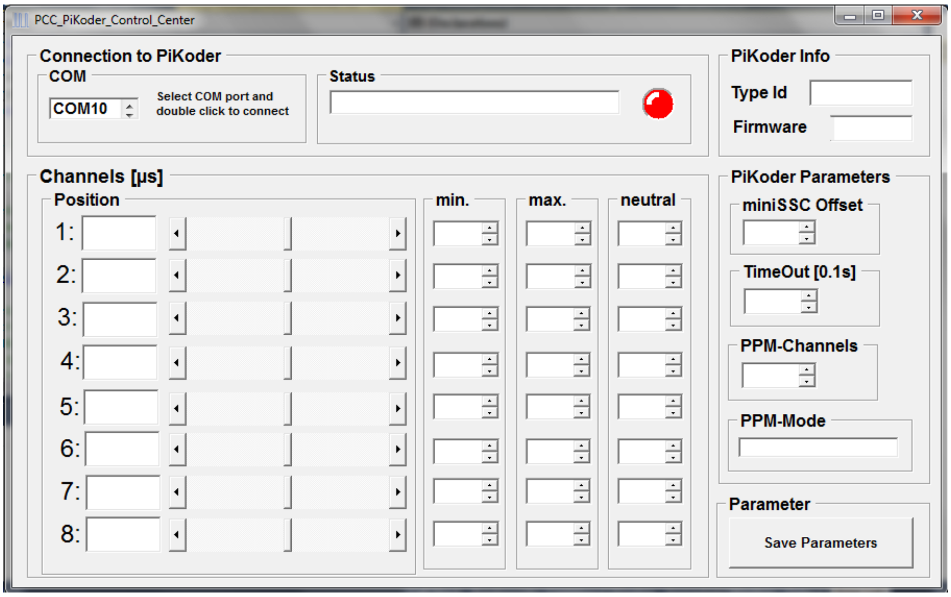
4

Connecting the PiKoder/SSC to a Windows 7 PC

This section describes how to establish the XBee based communication between your PC respectively the SSC Control Center Application for controlling and programming the PiKoder/SSC itself. From a hardware standpoint you will need an XBee USB adapter and the receiving unit described in the previous section. It is assumed that your radios are programmed in line with section 2.

Begin the setup by placing the radio which has been programmed as the coordinator into the XBee USB adapter. Connect the adapter to a free USB port. If you had programmed the radios earlier all required Windows 7 drivers should be set up already as needed.

Connecting the PiKoder/SSC to the PiKoder Control Center (PCC)



COM: COM10

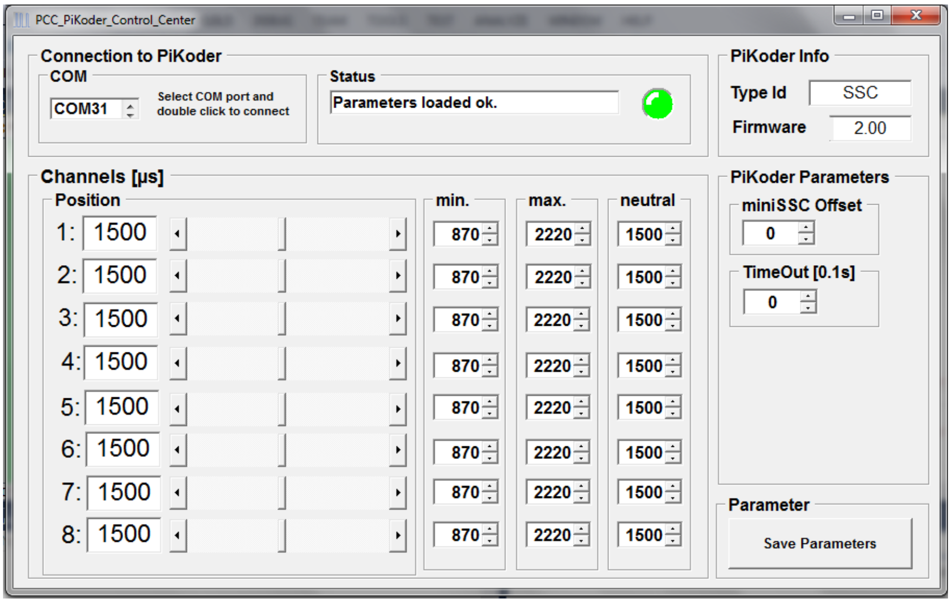
Select COM port and double click to connect

Please download the latest executable version of the PCC application from www.pikoder.de/PiKoder_PCC_EN.html and unzip the downloaded file or go to github.com/Pikoder/PCC_PiKoder_Control_Center to download the souce code and build the application yourself.

Start the PiKoder Control Center application and a screen similar to the one shown on the previous page would appear.

The PiKoder Control Center application would show COM10 to be the first available port in your computer. Select the COM port your PiKoder/SSC is connected to (in this example COM31, see below) and then click on the box to indicate your selection and communication to the controller via XBee will be established.

The LED color would change to green for an online indication and the current parameters and settings of the PiKoder/SSC would be loaded and displayed as shown on the following page.


PCC_PiKoder_Control_Center
Select COM port and double click to connect
Parameter
Save Parameters

Connection to PiKoder

COM: COM31

Status: Parameters loaded ok.

PiKoder Info

Type Id: SSC

Firmware: 2.00

Channels [µs]

Position	min.	max.	neutral
1: 1500	870	2220	1500
2: 1500	870	2220	1500
3: 1500	870	2220	1500
4: 1500	870	2220	1500
5: 1500	870	2220	1500
6: 1500	870	2220	1500
7: 1500	870	2220	1500
8: 1500	870	2220	1500

PiKoder Parameters

miniSSC Offset: 0

TimeOut [0.1s]: 0

You would now have full control of your SSC: either for real-time control by the sliders or for changing the settings and saving the new parameters.