XBECOM Default Config Profile

The shipped config of each of the COM units is

AP=0

ID = 2513

BAUD = 9600 (Digi factory default)

Parity = N (Digi factory default)

Stopbits = 1 (Digi factory default)

NI = XBEECOMXXXXX

DH = 0 (Digi factory default)

DL = FFFF (Digi factory default)

XXXXX = unique serial number, The NI will be on a sticker, visible on the top of each unit

All other settings as Digi factory default

Communication with host/Pi XBEE units in transparent mode

This post states that you can still send API commands to a remote regardless of the mode it's in, but local XBEE modules in transparent mode need to be changed back into API mode before any commands can be issued.

https://www.digi.com/support/forum/5661/transparent-mode-remote-configuration?show=5662#a5662

So to make any setting changes to a local XBEE it must be in command mode either by sending it (1second pause)+++(1second pause) or it already being in API mode

Sending an XBEE in transparent mode +++ causes it to **temporarily** break out into command mode, where you can issue AT commands to change settings, this mode change times out after about 10 seconds and will then revert back to transparent mode.

Any changes sent to the module in this mode will not take immediate effect until an AC (apply changes, stay in command mode) or CN (apply changes, exit command mode) commands are issued; additionally a WR (write settings command) is also needed if the changes are to persist through module reset or power off/on cycles.

To reconfigure a **local XBEE in transparent mode back to API mode**, I think you will need to use a standard python serial port lib to directly communicate with the XBEE in order to to issue the +++ sequence to change the XBEE into command mode

```
setlocalxbeetoapi –-port=ttyS1 --baud="9600,8,N,1"  --save
```

```
--port=serial port to connect to
--baud="baud,databits, parity, stop bits”
--save (issue a WR to make changes permanent, optional)
```

```
### Reset unit back to API mode ###
open local serial port (connected to XBEE) using standard serial library,
taking command line arguments for serial port and baud rate settings
wait 1second
send +++
wait 1second
expect OK response
send ATAP=2\r
expect OK response
send ATAC\r
expect OK response
if --save option used
      send ATWR\r
      expect OK response
echo "XBEE Now Set to API Mode"
```

We can then use the xbee lib to reconfigure the **local** XBEE module using API mode commands as required.

In the event no/insufficient/–h/--help options are specified display help text above

To make baud rate changes to a **local xbee already in transparent mode** I think it would be more efficient to use the +++ approach

```
setlocalxbeebaud –-port=ttyS1 --oldbaud="9600,8,N,1" --newbaud="19200,8,E,2"  --save

--port=serial port to connect to
--oldbaud="baud,databits, parity, stop bits”
--newdbaud="baud,databits, parity, stop bits”
--save (issue a WR to make changes permanent, optional)

## # Change Baud Rate ###
open local serial port (connected to XBEE) using standard serial library, taking
command line arguments for serial port and baud rate settings
wait 1second
send +++
wait 1second
expect OK response
send ATBD=$newbaud\r
expect OK response
send ATNB=$newparity\r
expect OK response
send ATSD=$newstopbits\r
expect OK response
if –-save option used
       send ATWR\r
       expect OK response
send ATCN\r
close serial port
echo "XBEE BAUD setting now Now set to $newbaud,8,$newparity,$newstopbits"
```

Note databits is ignored and cannot be changed from 8 on the XBEE unit, inclusion only because it’s unusual to not specify it

In the event no/insufficient/–h/--help options are specified display help text above

To change the destination address of **a local xbee already in transparent mode** I think it would be more efficient to use the +++ approach

```
setxbeedest –-port=ttyS1 --baud="9600,8,N,1" -–remote=”XBEECOM1” –-save
```

```
--port=serial port to connect to
--baud="baud,databits, parity, stop bits”
--remote=alphanumeric XBEE node name (NI)
--save (issue a WR to make changes permanent, optional)
```

```
### Change Destination Address ###
Load in network config dictionary with NI vs addresses
Translate the remoteaddress NI to DH/DL address using config
open local serial port (connected to XBEE) using standard serial library,
taking command line arguments for serial port and baud rate settings
wait 1second
send +++
wait 1second
expect OK response
send ATDH=$newdh\r
expect OK response
send ATDL=$newdl\r
expect OK response
if –save option
      send ATWR\r
      expect OK response
echo "XBEE Destination now Set XBEECOM1 (DH=$newdh DL=$newdl)"
send ATCN\r
close serial port
```

In the event no/insufficient/–h/--help options are specified display help text above

Network configuration and discovery

These scripts operate with the XBEE in API mode and use either of the Digi Python libraries as best suits.

Each script runs on the Pi unit and should perform the tasks shown

The first script does the initial configuration of the local/host XBEE unit fitted to the Pi board and configures all the remote COM outstations in transparent mode to send their (response) data back to this module/node.

```
configurenetwork –-port=ttyS1 --baud="9600,8,N,1"

--port=serial port to connect to
--baud="baud,databits, parity, stop bits”
--address=address to use instead of FFFF for AG command
```

Open local serial port (connected to XBEE) using either digi library, taking command line arguments for serial port and baud rate settings

Set local network ID to 2513 (ID)
Set local Node ID to “XBEEPI”

Apply changes

Scan network for nodes, writing out a config file detailing network names vs addresses for use by other programs. If file already exists overwrite it. The format of this file should be the same as used by the IO scripts.

Issue AG command using broadcast address (00000000000FFFF) to configure the DH/DL setting on each of the remote units to the host unit’s address and force nodes to build network mesh routes

Note: As it is not clear from the documentation if the result of the AG command’s setting the DH/DL on the remote units is then saved/written or if we have to manually issue “WR” commands to each of the remote units, let’s assume it isn’t saved so we need to trigger this manually.

For each of the the discovered network nodes that have a node name containing “COM” issue a remote WR command

Print out a list of the nodes on the network

In the event no/insufficient/–h/--help options are specified display help text above

Next we need a script to manually configure the baud rate setting in a given remote

```
xbeesetremotebaud –-port=ttyS1 --baud="9600,8,N,1" -–remote=”XBEECOM1” -
remotebaud=”19200,8,N,1” --save
```

--port=serial port to connect to
--baud="baud,databits, parity, stop bits”
--remotebaud="baud,**databits**, parity, stop bits”

--remote=alphanumeric remote XBEE node name (NI)
--agonly (skip section setting DH/DL directly and only execute AG section, optional)
--address=address to use instead of FFFE for AG command (optional)
--save (issue a WR to make changes permanent, optional)

Open local serial port (connected to XBEE) using either digi library, taking command line arguments for serial port and baud rate settings
Load in network config dictionary with NI vs addresses

Translate the remote address NI to DH/DL address using config
If lookup fails, exit suggesting network rescan (should already have a script for this)

Set the BD,NB,SD settings on the remote COM unit

Issue a remote WR command to the target remote unit

Issue a remote AC command (if required) to apply changes

Print confirmation message if script was successful stating new baud rate set on remote unit

In the event no/insufficient/–h/--help options are specified display help text above

Finally we need an additional script to manually configure the DH/DL setting on a given remote in the event of a host node swap out or other situation where the remote node's DH/DL setting is incorrect.

```
xbeesetremotedhdl –-port=ttyS1 --baud="9600,8,N,1" -–remote=”XBEECOM1” --save

--port=serial port to connect to
--baud="baud,databits, parity, stop bits”
--remote=alphanumeric remote XBEE node name (NI)
--agonly (skip section setting DH/DL directly and only execute AG section,
optional)
--address=address to use instead of FFFE for AG command (optional)
--save (issue a WR to make changes permanent, optional)
```

```
Open local serial port (connected to XBEE) using either digi library, taking
command line arguments for serial port and baud rate settings
Load in network config dictionary with NI vs addresses

Translate the remote address NI to DH/DL address using config
If lookup fails, exit suggesting network rescan (should already have a script
for this)

Set the DH&DL to the SH/SL address of the host XBEEPI unit

Issue AG command using invalid broadcast address (000000000000FFFE)to force
nodes to build/update network mesh routes without affecting DH/DL on nodes

Issue a remote WR command to the target remote unit

Print confirmation message if script was successful
```

Note databits is ignored and cannot be changed from 8 on the XBEE unit, inclusion only because it's unusual to not specify it

In the event no/insufficient/–h/--help options are specified display help text above