

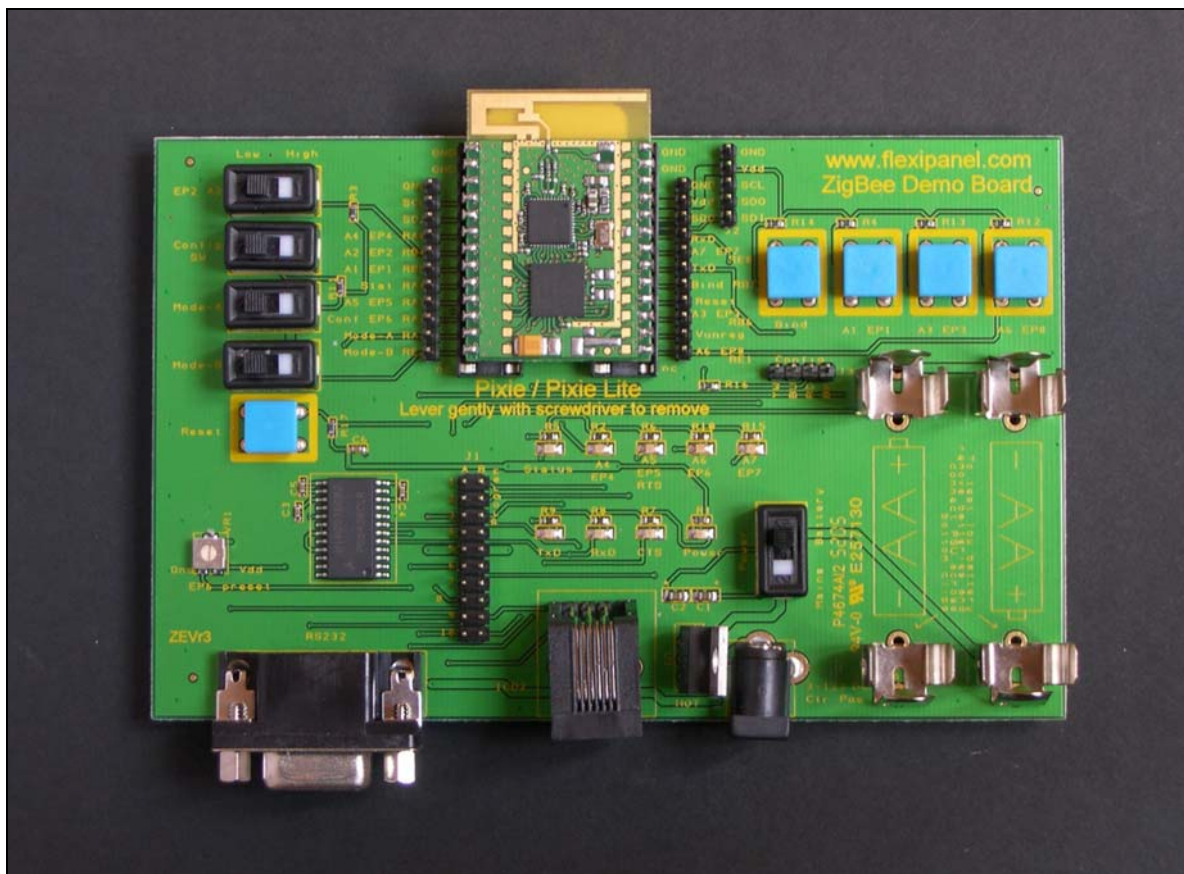
## Summary

The Pixie Evaluation Kit is designed for research and development with FlexiPanel Ltd's Pixie ZigBee modules. It is designed to be compatible with FlexiPanel's firmware solutions, including:

- Switcher (preloaded, tutorials 1-3)
- Packet sniffing (tutorial 4)
- DARC data acquisition & remote control (tutorial 5)
- MailBox Gateway (tutorial 5)
- StarLite and StarLite USB (tutorial 6)

## Board Features

- 4 switch inputs
- 4 pushbutton inputs
- 8 LED outputs
- 1 trimmer input
- RS232 serial port and TTL to RS232 converter
- ICD2 programming socket
- Configuration tool socket
- Power regulator and battery clips.
- Solder pads for custom development



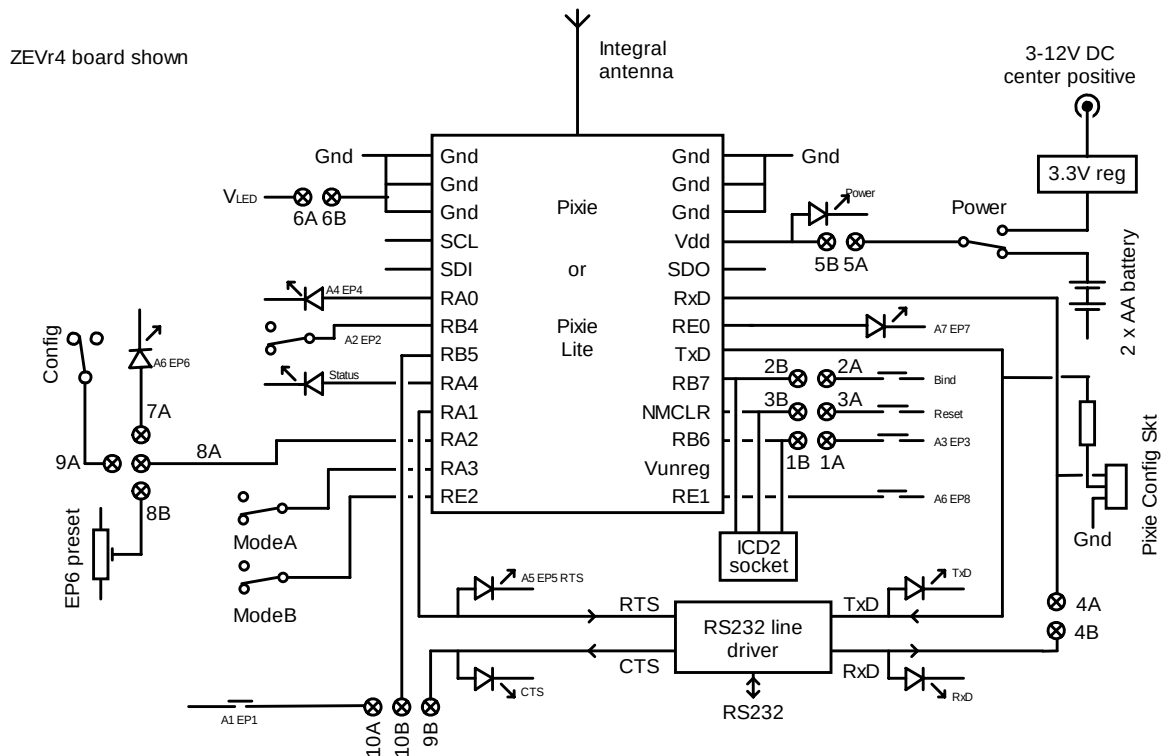
## Ordering Information

Manufactured to ISO9001:2000



| Part No    | Description                                                         |
|------------|---------------------------------------------------------------------|
| EVAL-PIXIE | ZigBee Evaluation Kit: 2 boards, including 1 Pixie and 1 Pixie Lite |

## Schematic diagram



## Firmware

The Pixie Evaluation boards are supplied in sets of two, one with a Pixie module mounted onboard, the other with a Pixie Lite module mounted on-board.

The modules are preloaded with Pixie Switcher firmware. This is simple switching firmware using the ZigBee HC-L profile for turning things on and off. A short tutorial is given on the next page. For detailed instructions, please read the Pixie Switcher data sheet. Contact us regarding the availability of other firmware.

## Power

Power may be provided from a 3-12 VDC supply using the 2.5mm center positive connector. Alternatively, two AA batteries can be connected. The cylindrical wall of the battery must be insulating to be compatible with the metal battery clips supplied.

## Jumper switches

Connect 1A-1B, 2A-2B, 3A-3B during normal use. Remove them to program via the ICD2 connector.

4A-4B connects the RS232 driver to the Pixie RxD input pin, allowing operation. Connect this jumper to use the RS232 port, or remove it to use the Pixie Config Tool connector. (Applies to revision ZEVr4

boards and higher. For ZEVr3 boards, refer to Revision ZEVr3 notes.)

Connect 5A-5B during normal use. Connect an ammeter across these pins for current consumption measurement.

Connect 6A-6B during normal use. Remove to disconnect power from LEDs and trimmer for current measurements.

Connect 8A-7A for an LED on the RA2 pin, or 8A-8B for a trimmer, or 8A-9A for push switch.

Connect 10B-9B for RB5 to connect to CTS flow control input, or 10B-10A for push switch.

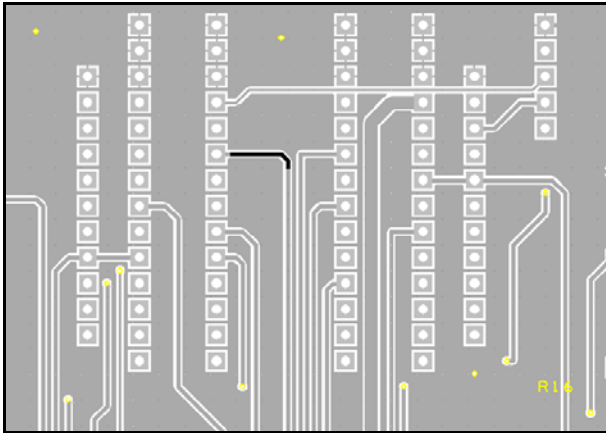
## Programming with ICD2

To program the Pixies using the ICD2 programmer / debugger from Microchip Technology, connect the ICD2 programmer using the RJ11 socket marked ICD2. Jumper switches 1A-1B, 2A-2B, 3A-3B should be removed while programming the Pixies.

## Revision ZEVr3 notes

(1) Config tool port will be TxD only unless the RxD input from the RS232 line driver and the LED is disconnected. To do this (irreversibly), use a sharp knife to cut the track on the board underside

highlighted in black in the diagram below. Take care not to cut any other tracks, not to short the track to the ground plane, and not to damage the Pixie by leaning the board on it. Alternatively, remove the RS232 line driver entirely – see note (2).



(2) RS232 driver is not powered down by removing jumper 6A-6B. For accurate power consumption measurement this component must be removed. (The Config tool can still be used for serial communications.)

## Quick Start Guide – Switcher

The firmware loaded on the evaluation boards is Pixie Switcher. This is simple switching firmware using the ZigBee HC-L profile for turning things on and off. Contact us regarding the availability of other firmware.

This documentation assumes the Switcher firmware you have loaded is version 1.0-3.5-2.0. You can find out which firmware you have loaded in step 6 of tutorial 1.

The Pixie Evaluation Board (marked ZPCrX) is loaded with ZigBee HC-L *coordinator* firmware. The Pixie Lite evaluation board (ZPKrX) is loaded with ZigBee HC-L *sleepy end device* firmware. You might want to label them:

Pixie Board = Coordinator

Pixie Lite Board = Sleepy End Device

In the first tutorial you will set up endpoints for switching between the coordinator and the sleepy end device. You will need a PC and at least one RS232 serial port, preferably two.

In the second tutorial you will set up failsafe alarm which allows battery powered devices to confirm that they are working correctly.

In the third tutorial you will add an additional router and end device. In the fourth tutorial, you will reprogram the router for use as a sniffer so you can watch messages being sent. To perform the third and fourth tutorials you will need:

- A Pixie Switcher Firmware License. This is free on application from FlexiPanel Ltd and enables us to send you the switcher firmware hex files.
- A second Pixie Evaluation Kit
- An ICD2 programmer to program the Pixies. This is available from Microchip Technology.
- MPLAB PIC development environment software for controlling the ICD2. This is a free download from Microchip Technology.
- Pixie Sniffer Development Kit. This is a free download from FlexiPanel Ltd.

## Tutorial 1 – ZigBee Switching Concepts

In this tutorial you will cover basic ZigBee concepts such as joining and binding while setting up a ZigBee switching system.

### Configuring the boards

The purpose of configuring the board is to tell the firmware what hardware is connected to it. This would usually be done at the OEM's factory with the use of a serial port or configuration tool.

1. Connect the serial port to the D9 socket on the Coordinator.
2. Set up a HyperTerminal session on the PC with the serial port set at 19200 baud, 8N1, no flow control.
3. Connect the jumpers on the Coordinator as follows: 1A-1B, 2A-2B, 3A-3B, 5A-5B, 6A-6B, 8A-8B, 10A-10B
4. Power up the Coordinator. Various LEDs may turn on and initialization messages may appear in the HyperTerminal window.
5. If at step 4 you received the message

MAC ->

you must type in a MAC address. (If you don't, go to step 6.) For evaluation purposes, type in any 10-hex-digit value in the range 3841000000 to 384100FFFF. **Note: the MAC address of each device MUST be different!** You will only have to do this once after programming using the ICD2.

- Verify that you get a startup message containing the version number 1.0-3.5-2.0, e.g.:

DIFC PXSC=1.0-3.5-2.0 (Coordinator)

DIFC PLFE=1.0-3.5-2.0 (Sleepy End Device)

This confirms you are running the correct firmware for this tutorial. If you don't see 1.0-3.5-2.0 or higher, contact us to obtain the correct firmware.

- Enter the letter **F** into HyperTerminal to perform a factory reset. This ensures that the firmware is in the configuration this tutorial expects it to be in. You will need to press **Y** to confirm.

Note that when you power up any Switcher unit except the Coordinator, it will look for a network. If it can't find one, it will enter a sleep state. So if you get no response when you type something, the device may be sleeping. Press the bind button once to wake it up again.

- When the device has reset, enter **M** to turn on diagnostic messages. These are very useful if you wish to understand the internal workings of the Switcher devices. The messages are explained in the document *Pixie Switcher Message Reference DS493*.

Note that messages are not normally available for Pixie Switcher Lite. However, an alternate hex file is included in the development kit with diagnostic messages enabled and failsafe disabled for the Fast End Device. You will use this in Tutorials 3 and 4.

- You will now specify the function of the input and output pins. Press type **E** into the HyperTerminal window, and when prompted, the following characters:

TLD00M0T

(That's three zeroes, not the letter O.) You have specified the following I/O:

| Endpoint (EP) | Function                      |
|---------------|-------------------------------|
| 1             | Toggle switch (T)             |
| 2             | Latching switch (L)           |
| 3             | Time delay input (D)          |
| 4             | Output, initially off (O)     |
| 5             | Output, initially off (O)     |
| 6             | Time delay setpoint input (M) |
| 7             | Output, initially off (O)     |
| 8             | Toggle switch (T)             |

- On the Sleepy End Device, connect jumpers as follows: 1A-1B, 2A-2B, 3A-3B, 5A-5B, 6A-6B, 7A-

8A, 10A-10B. Note this is slightly different from the coordinator board.

- Power up the Sleepy End Device. If you have two serial ports, connect the second to the board; if not, just swap over the connector from the first as needed.

- Repeat steps 2 to 8 for the Sleepy End Device.

- You will now specify the function of the input and output pins. Enter **E** and, when prompted, the following characters:

TLT01TUU

(That's zero-one.) Remember you may need to wake it by pressing the Bind button first. You have specified the following I/O:

| Endpoint (EP) | Function                  |
|---------------|---------------------------|
| 1             | Toggle switch (T)         |
| 2             | Latching switch (L)       |
| 3             | Toggle switch (T)         |
| 4             | Output, initially off (O) |
| 5             | Output, initially on (1)  |
| 6             | Toggle switch (T)         |
| 7             | Unassigned (U)            |
| 8             | Unassigned (U)            |

- Enter **W** repeatedly until you get the message:

DWKI WkUp=04

You have specified that you want the device to sleep and wake up every 2 seconds to check for messages. If any of the EP1 – EP3 or the Bind switch change state, the device will wake immediately. (If you have only inputs connected to EP1 – EP3, no wakeup would be required at all.)

Note that input and output settings must correspond to the circuit that the Pixie Switcher is connected to. All the steps to this point are product specific and you would normally execute them at the factory during product manufacture.

## Joining

The rest of the tutorial does not require the serial connections, but you may wish to leave them connected to read the diagnostic messages out of interest.

The following steps are one-time only and would normally be done on-site at the time the equipment is installed by the customer or technician.

15. Power up and/or reset the Coordinator with the Bind button held down. Once the Status LED lights, you may release the button. This performs an erase reset. The settings programmed in steps 1-14 will not be changed, but any network membership information will be erased. This ensures that the firmware is in the configuration this tutorial expects it to be in.
  16. After initializing, the Status LED will flash the letter **L** in Morse code (•-••) while it is looking for a frequency to operate on. When it is ready, it will flash the letter **Y** (-•-•-).
  17. Press the Bind button on the Coordinator once to tell it to permit another device to join. The LED will flash the Morse letter **J** (•-•-•) to indicate that it is allowing a device to join. You have one minute within which to join another device.
  18. Power up and/or reset the Sleepy End Device. Then it will look for a network to join. The LED will flash the letter **L** in Morse code (•-••). When it finds the network, it will flash the Morse letter **Y** (-•-•-). It has successfully joined. If the letter **N** (-•) flashes, it failed to find the network and you should repeat this step again.
  19. The Sleepy End Device is now a member of the Coordinators network.
  20. Press the Bind button on the Sleepy End Device once. The Status LED will flash a number as a series of 0-10 long pulses (tens) followed by series of 0-9 short pulses (units). This is the signal strength, an indication of the quality of the link between the end device and its parent. This signal strength indication is only available for Fast and Sleepy End Devices. (Routers and coordinators use a single press of the bind button instead to permit joining.)  
  
The signal strength can be used to establish the effect of various obstacles during network installation. The value reported is the signal strength of the last packet of data received from the parent. Signal strength is on a scale of 0 (-110dBm, weakest) to 110 (0dBm, strongest).
- ## Binding and Switching
- You will now bind individual endpoints together. This involves multiple presses of the Bind button on both devices. Note that the number of times the button is pressed is always the endpoint number plus one, e.g. to bind EP2, press the button 3 times.
21. Press the bind button on the coordinator twice in reasonably quick succession, then on the end device 6 times. After a few seconds, both devices will flash the letter **Y** (-•-•-) indicating that the bind was successful.
  22. You have bound pushbutton EP1 on the coordinator to LED EP5 on the end device. Press the button EP1 on the coordinator and it will toggle the state of the EP5 LED on the end device. Note it can take up to two seconds to respond, because the end device spends much of its time sleeping.
  23. Press the bind button on the coordinator 9 times, then on the end device 6 times. After a few seconds, both devices will flash the letter **Y** in Morse code (-•-•-) indicating that the bind was successful.
  24. You have also bound pushbutton EP8 on the coordinator to LED EP5 on the end device. Pressing either button EP1 or EP8 on the coordinator will toggle the state of the EP5 LED on the end device. This is how multiple inputs can control one output.
  25. Press the bind button on the end device 3 times, then on the coordinator 5 times. After a few seconds, both devices will flash the letter **Y** in Morse code (-•-•-) indicating that the bind was successful.
  26. You have bound latching switch EP2 on the end device to LED EP4 on the coordinator. Throwing switch EP2 on the end device will set the state of the EP4 LED on the coordinator. Note the response is immediate, since the end device is woken by interrupt and sends the message immediately. If this doesn't work, check the jumper settings.
  27. Press the bind button on the end device 3 times, then on the coordinator 6 times. After a few seconds, both devices will flash the letter **Y** in Morse code (-•-•-) indicating that the bind was successful.
  28. You have also bound latching switch EP2 on the end device to LED EP5 on the coordinator. Throwing switch EP2 on the end device will set the state of both EP4 and EP5 LEDs on the coordinator. This is how one input can control multiple outputs. You could, for example, have one pushbutton light switch turn off all the lights in a house.

29. Set the position of trimmer EP6 to the position shown, so the wiper is approximately 0.48V.



30. Press the bind button on the coordinator 4 times, then on the end device 5 times. After a few seconds, both devices will flash the letter **Y** in Morse code (– • –) indicating that the bind was successful.
31. You have bound timer switch EP3 on the coordinator to LED EP5 on the end device. Pressing EP3 on the coordinator will light EP5 on the end device and then extinguish it after about 8 seconds. The time delay can be set from 1 second to 18 hours using the trimmer EP6.

| Voltage range | Time delay                            |
|---------------|---------------------------------------|
| 0.00V – 0.67V | From 1 sec to 10 sec in 1 sec steps   |
| 0.67V – 1.33V | Varies linearly from 10 sec to 60 sec |
| 1.33V – 2.00V | Varies linearly from 1 min to 10 min  |
| 2.00V – 2.67V | Varies linearly from 10 min to 60 min |
| 2.67V – 3.33V | Varies linearly from 1 hr to 18 hrs   |

Not all endpoints specified in steps 6 and 10 have been bound. So experiment with binding them. Also try un-binding, which is simply to repeat the binding process on an already bound input / output. Remember, to bind endpoint EPx, you will need to press the Bind button (x+1) times.

## Tutorial 2

This tutorial illustrates the use of the failsafe input and output. A failsafe input regularly sends messages to the failsafe output. If the failsafe output does not receive the messages, it raises an alarm.

### Configuring the boards

1. The serial connections, HyperTerminal sessions and jumper switch connections may stay the same as for Tutorial 1.
2. On the coordinator board, enter the letter **F** into HyperTerminal to perform a factory reset. This ensures that the firmware is in the configuration this tutorial expects it to be in. You will need to press **Y** to confirm.

3. When the device has reset, enter **M** to turn on diagnostic messages.
4. Press type **E** into the HyperTerminal window, and when prompted, the following characters:

UUU0BUUU

(That's a zero, not the letter O.) You have specified the following I/O:

| Endpoint (EP) | Function                          |
|---------------|-----------------------------------|
| 1, 2, 3       | Unassigned (U)                    |
| 4             | Output, initially off (0)         |
| 5             | Failsafe output, initially on (B) |
| 6, 7, 8       | Unassigned (U)                    |

5. Enter **A** and, when prompted, the following settings:

Retransmit = 0000  
Quiet = 000A  
Undervolt = 0000

The Quiet = 000A setting indicates that if a failsafe output does not hear from the input for 10 seconds (000A in hex), it should turn on automatically.

6. Repeat step 2 for the Sleepy End Device.
7. Enter **E** and, when prompted, the following characters:

TSUUUUUU

Remember you may need to wake it by pressing the Bind button first. You have specified the following I/O:

| Endpoint (EP)    | Function           |
|------------------|--------------------|
| 1                | Toggle switch (T)  |
| 2                | Failsafe input (S) |
| 3, 4, 5, 6, 7, 8 | Unassigned (U)     |

8. Enter **A** and, when prompted, the following settings:

Retransmit = 0004  
Quiet = 0000  
Undervolt = 0898

The Retransmit = 0004 setting indicates that the failsafe input should transmit an "I'm OK" message every 4 seconds. Likewise, the Undervolt = 0898 indicates that if the battery voltage falls below 2300mV (0898 in hex), it should transmit an "I'm not OK" message.

9. Enter **W** repeatedly until you get the message:

DWKI WkUp=04

You have specified that you want the device to sleep when not busy and wake up every 2 seconds to check for messages

### Joining

10. Power up and/or reset the Coordinator with the Bind button held down. Once the Status LED lights, you may release the button.
11. After initializing, the Status LED will flash the letter **L** in Morse code (•–••) while it is selecting a frequency to operate on. When it is ready, it will flash the letter **Y** (–•–) indicating that a failsafe alarm condition exists – it will not extinguish until it receives a message from the failsafe input.
12. Press the Bind button on the Coordinator once to tell it to permit another device to join. The LED will flash the Morse letter **J** (•––) to indicate that it is allowing a device to join. You have one minute within which to join another device.
13. Power up and/or reset the Sleepy End Device. It will look for a network to join. The LED will flash the letter **L** in Morse code (•–••). When it finds the network, it will flash the Morse letter **Y** (–•–). It has successfully joined. If the letter **N** (–•) flashes, it failed to find the network and you should repeat this step again.
14. The Sleepy End Device is now a member of the Coordinator's network.
17. Ensure the Sleepy End Device's EP2 button is in the off ('high') position.
18. Press the bind button on the end device 3 times, then on the coordinator 6 times. After a few seconds, both devices will flash the letter **Y** in Morse code (–•–) indicating that the bind was successful.
19. You have bound pushbutton EP2 on the end device to LED EP5 on the coordinator. This input / output relationship performs the failsafe alarm function between the two devices. Note how the LED on EP5 extinguished once it started receiving messages from the failsafe input every four seconds. (Normally the delays would be hours, not seconds, to conserve battery life.)
20. There are four ways to trigger the failsafe alarm EP5 on the coordinator:
  - If you turn on the failsafe input switch EP2, an alarm is signaled immediately.
  - If the input device cease to function (e.g. try turning it off) the alarm triggers after the failsafe quiet period.
  - If the power supply to the end device falls below the undervolt level, 2300mV. This is not easy to simulate unless you connect a laboratory power supply to the battery clips. (Make sure the voltage does not fall below 2000mV. This is the brownout reset voltage.)

### Binding and Switching

You will now bind individual endpoints together. This involves multiple presses of the Bind button on both devices. Note that the number of times the button is pressed is always the endpoint number plus one, e.g. to bind EP2, press the button 3 times.

15. Press the bind button on the end device twice, then on the coordinator 5 times. After a few seconds, both devices will flash the Morse letter **Y** (–•–) indicating that the bind was successful.
16. You have bound pushbutton EP1 on the coordinator to LED EP4 on the end device. Press the button EP1 on the coordinator and it will toggle the state of the EP4 LED on the end device. This input / output relationship performs the normal switching function of the devices.

### Tutorial 3

To perform tutorials 3 and 4 you will need:

- A Pixie Switcher Firmware License. This is free on application from FlexiPanel Ltd and enables us to send you the switcher firmware hex files.
- A second Pixie Evaluation Kit
- An ICD2 programmer to program the Pixies. This is available from Microchip Technology.
- MPLAB PIC development environment software for controlling the ICD2. This is a free download from Microchip Technology.
- Pixie Sniffer Development Kit. This is a free download from FlexiPanel Ltd.

Tutorial 3 illustrates how routers may be used to send messages longer distances. The network will have one coordinator, one router and two end devices.

### Reprogramming the Pixies

You will need to reprogram some of the Pixies for this tutorial.



1. The Coordinator you have been using does not need reprogramming, but you should type **F** to make it perform a factory reset.
2. Both Pixie Lite boards will be programmed as Fast End Devices. They are called “fast” because they do not sleep. Normally, diagnostic messages are not available for Pixie Lite products, but there is an alternate build which provides them in place of failsafe alarms. You are about to load this firmware so you can see the diagnostic messages for the end devices.
3. On your Windows PC, start up MPLAB and connect the ICD2 In-Circuit Debugger.
4. Choose *Programmer > Select Programmer > MPLAB ICD2* from the MPLAB menu. You should see the message “...Connected”. (Don’t worry if you get an “Invalid Target ID” message.) If you get the message “Failed to open port”, check the USB connection and choose *Programmer > Connect* from the menu. You may have to do this several times.
5. Choose *Programmer > Settings > Power tab*. Ensure the check box “Power target circuit from MPLAB ICD2” is **not** checked.
6. Remove the jumper pins A1-B1, A2-B2, A3-B3 from one of the Pixie Lite boards. This is necessary for programming. Connect the RJ11 programming connector from ICD2 to the RJ11 socket marked IDC2 on the board. Power up the board.
7. Select *File > Import* in the MPLAB menu. Open the file PLFV-1.0-3.5-2.0.hex which is in the Pixie Switcher Development Kit. This is the firmware for the Fast End Device with diagnostic messages.
8. Choose *Programmer > Program* to reprogram the Pixie. After a few moments you should get the message “...Programming Succeeded”.

Note that if you buy Pixies in volume from us, they can be preloaded with the firmware of your choice.

9. Turn off the power to the board and replace jumper pins A1-B1, A2-B2, A3-B3. Reprogramming of this board is complete.
10. Repeat for the other Pixie Lite board.
11. The second Pixie Board is to be reprogrammed as a router. Remove the jumper pins A1-B1, A2-B2, A3-B3 from one of the Pixie board connect the RJ11 programming connector and apply power as before.

12. Choose *File > Import* in the MPLAB menu. Open the file PXS-1.0-3.5-2.0.hex which is in the Pixie Switcher Development Kit. This is the firmware for the Router.
13. Select *Programmer > Program* to reprogram the Pixie. After a few moments you should get the message “...Programming Succeeded”.
14. Turn off the power to the board, replace jumper pins A1-B1, A2-B2, A3-B3. Reprogramming of this board is complete.

## Configuring the boards

All boards will be configured with two input endpoints and two output endpoints.

15. Power up each board in turn with the serial connector connected. Set a unique MAC address if prompted to do so. Use the **F** command to perform a full factory reset on the board and an **M** command to switch on diagnostic messages.
16. Use the **E** command to set the following endpoints:

TLU00UUU

You have specified the following I/O:

| Endpoint (EP) | Function                  |
|---------------|---------------------------|
| 1             | Toggle switch (T)         |
| 2             | Latching switch (L)       |
| 3             | Unassigned (U)            |
| 4             | Output, initially off (O) |
| 5             | Output, initially off (O) |
| 6             | Unassigned (U)            |
| 7             | Unassigned (U)            |
| 8             | Unassigned (U)            |

## Joining

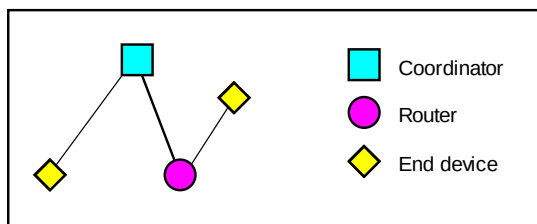
If, as you are performing the following steps, you need to restart the network, turn off all devices, then turn on the coordinator and wait for it to initialize. Then turn on the router and let it find the network. Finally turn on the children. This is the quickest way to return them to the initial state.

17. Power up and/or reset the Coordinator with the Bind button held down. Once the Status LED lights, you may release the button. When it has completed setting up a network, press the Bind button once so it will allow another device to join.
18. Power up one of the Fast End Devices with the Bind button held down. Once the Status LED



lights, you may release the button. The LED will flash the letter **L** in Morse code (•-••). When it finds the network, it will flash the Morse letter **Y** (-•-•-). It has successfully joined. If the letter **N** (-•) flashes, reset the board and try again.

19. Press the Bind button once on the coordinator so it will allow another device to join.
20. Power up the Router. The LED will flash the letter **L** in Morse code (•-••). When it finds the network, it will flash the Morse letter **Y** (-•-•-). It has successfully joined. If the letter **N** (-•) flashes, reset the board and try again.
21. Press the Bind button once on the Router so it will allow another device to join.
22. Power up the second Fast End Device. The LED will flash the letter **L** in Morse code (•-••). When it finds the network, it will flash the Morse letter **Y** (-•-•-). It has successfully joined. If the letter **N** (-•) flashes, reset the board and try again.
23. You now have a network with the following architecture:



24. Press the bind button on one of the End Devices twice and on the other End Device five times. After a few seconds, both devices will flash the letter **Y** in Morse code (-•-•-) indicating that the bind was successful. (If one registers success and the other doesn't, try resetting both devices and continuing. This is an occasional bug in version 1.0-3.5-2.0.)
25. You have bound pushbutton EP1 on first end device to EP4 on the second end device. Press EP1 on the first end device and the LED EP4 should toggle on the second end device, routing through the coordinator and the router.

26. Each unit has an input on EP1 and EP2, and an output endpoint on EP4 and EP5. Try binding further inputs and outputs. Remember, to bind endpoint EPx, you will need to press the Bind button (x+1) times.

## Tutorial 4: Sniffer

Tutorial 4 illustrates the use of a sniffer. The network will use the same coordinator and two end devices as tutorial 3, but the router will be reprogrammed and used as a sniffer.

### Reprogramming the Router

To reprogram the router as a sniffer:

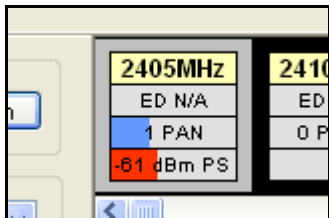
1. On your Windows PC, start up MPLAB and connect the ICD2 In-Circuit Debugger.
2. Remove the jumper pins A1-B1, A2-B2, A3-B3 from the router board. Connect the RJ11 programming connector from ICD2 to the RJ11 socket marked IDC2 on the board. Power up the board.
3. Choose *File > Import* in the MPLAB menu. Open the file `PXMF.hex` which is in the PixieMAC Development Kit. This is the firmware for the Sniffer.
4. Choose *Programmer > Program* to reprogram the Pixie. After a few moments you should get the message "...Programming Succeeded".
5. Turn off the power to the board, replace jumper pins A1-B1, A2-B2, A3-B3. Reprogramming of this board is complete.

### Starting the sniffer

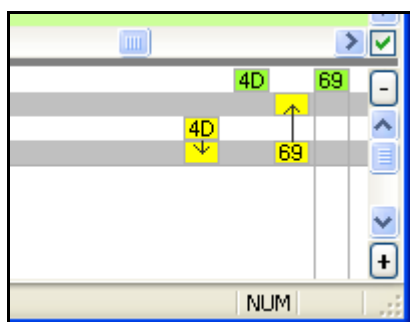
6. Turn off the End Devices. Turn on the Coordinator with the Bind button pressed down so that it establishes a new network.
7. Ensure a serial cable is connected to the Sniffer board but that HyperTerminal is not connected to the COM port.
8. Start the Sniffer application. This can be downloaded from [www.flexipanel.com](http://www.flexipanel.com).
9. In the Port / Channel Select box, select the COM port connected to the Sniffer. Press the Scan button.

Verify that the status bar in the sniffer says "Scanning" or "Sniffing". This confirms that it is communicating with the Eval Board.

10. When the scan of all frequencies is complete, the sniffer should detect the coordinator on one of the operating frequencies (see graphic below). The sniffer will then continue to listen on this frequency and display any messages which are transmitted. If it fails to find the network, try again.



11. Press the Bind button on the Coordinator so it allows another device to connect to the network. Then power up one of the end devices with the Bind button held down so it erases and joins the new network.
12. Repeat the previous step for the second end device. Exhibit A shows the typical exchange of messages seen when new devices join, which you should see in the sniffer.
13. Reset one of the end devices. It will rejoin the network. Exhibit B shows the typical exchange of messages seen when a device rejoins.
14. Press the Bind button on one end device twice and on the second end device five times. This binds an endpoint on each end device. Exhibit C shows the typical exchange of messages seen when devices bind.
15. Press the EP1 button on the first end device. The EP4 light should toggle on the other end device. Exhibit D shows the typical exchange of messages seen when a switch message is sent.
16. Note, in Exhibit D, the bottom pane where the timeline is shown. It is reproduced below:



The yellow 4D frame is the input sending the message to the coordinator. The coordinator acknowledges (green). Then it forwards the

message to the other end device (yellow 69 frame), which then acknowledges (green).

The PixieMAC firmware for the Pixies and the MACdongle firmware for the UZBee have the same interface. If you have an UZBee, you can use it as a sniffer by loading it with the UMDf firmware which you will find in the MACdongle development kit.

## Tutorial 5: MailBox

Tutorial 5 demonstrates the MailBox Gateway and PixieDARC firmware. They use then Mailbox profile which is a generic protocol for sequenced transmission of data payloads of up to 64 bytes long. Its main advantages are that the same protocol can be used for any custom applications, a new ZigBee profile does not need to be developed each time, and it allows products to interface to existing MailBox devices.

MailBox Gateway is a command/response style interface to the MailBox profile. It is very flexible and provides access to all MailBox functions via a serial port.

The PixieDARC firmware is a simple Data Acquisition / Remote Control device using the MailBox protocol. You can send it commands to read analog and digital inputs, and also set digital outputs. Readings can also be streamed for retransmission at regular intervals.

PixieDARC is a very simple ZigBee application developed using the MailBox API (application programming interface). The API takes care of the ZigBee technicalities and allows the developer to concentrate on the application-specific tasks. To assist in the development of applications, the source code for PixieDARC is available in the MailBox API development kit.

## Reprogramming the Pixies

You will need to reprogram a Pixie as a Gateway coordinator and a Pixie Lite as a PixieDARC device. If you wish to add further boards, they could be Gateways or PixieDARCs, although only the PixieDARC end devices are small enough to run on the Pixie Lites.

If you have boards to spare, you might like to set use one as a sniffer so you can watch the messages. If you do so, ensure the Sniffer menu item File > Parse MailBox Frames is checked in order to translate MailBox messages.

To reprogram a Pixie as a Gateway coordinator:

1. On your Windows PC, start up MPLAB and connect the ICD2 In-Circuit Debugger.
2. Remove the jumper pins A1-B1, A2-B2, A3-B3 from the board. Connect the RJ11 programming connector from ICD2 to the RJ11 socket marked IDC2 on the board. Power up the board.
3. Choose *File > Import* in the MPLAB menu. Open the file PGCH.hex from in the Pixie Gateway development kit. This is the firmware for the Gateway coordinator.
4. Choose *Programmer > Program* to reprogram the Pixie. After a few moments you should get the message "Programming Succeeded".
5. Turn off the power to the board, replace jumper pins A1-B1, A2-B2, A3-B3.
6. Change the HyperTerminal interface so the baud rate is 115200 baud 8N1, no flow control.
7. Ensure the *Config SW* is in the *Low* position and ensure it is connected to the pin by fitting jumper A8-A9. This is the flow control. If it is high, you will not receive any messages from the Gateway.
8. Reset the board. The EP5 LED should flash briefly and then the following message will appear in HyperTerminal:

**+DRYI**

The message is from the Gateway indicating that it is ready.

9. The first time you use the Gateway after programming, you will need to set the MAC address. Do this by typing the following:

**+DSMR=0000000000C81500**

You should receive the following message in confirmation:

**+DSMC=00**

This only needs to be done once. If you try to repeat it, you will receive an error message.

If at any time you wish to abort typing the line you are entering on the Gateway, press Z and then type Enter. You will receive a DERI error message and may then start the command from the beginning.

To reprogram a Pixie Lite as a PixieDARC sleepy end device:

10. On your Windows PC, start up MPLAB and connect the ICD2 In-Circuit Debugger.
11. Remove the jumper pins A1-B1, A2-B2, A3-B3 from the board. Connect the RJ11 programming connector from ICD2 to the RJ11 socket marked IDC2 on the board. Power up the board.
12. Choose *Configure > Select Device* in the MPLAB menu and set the device to an 18F2520. The select *File > Import* and open the file LHDS.hex which is in the PixieDARC development kit. This is the firmware for the DARC sleepy end device.
13. Choose *Programmer > Program* to reprogram the Pixie. After a few moments you should get the message "Programming Succeeded".
14. Turn off the power to the board, replace jumper pins A1-B1, A2-B2, A3-B3.
15. Change the HyperTerminal interface so the baud rate is 19200 baud 8N1, no flow control. (Yes that's different to the Gateway! You will only use the serial communications for setting the MAC address on the Pixie DARC.)
16. Reset the board. The Status LED should light and then the following message will appear in HyperTerminal:

**MAC->**

17. It is asking for you to set the MAC address. Type the following:

**0100000000C81500**

You should receive the following message in confirmation:

**OK**

This only needs to be done once. Now turn off the PixieDARC until you are ready to use it.

## Joining

You are now ready to send instructions to the Gateway.

18. The Gateway is the coordinator. To start the network, type the following:

**+MJNR=01**

It will scan for a new channel on which to start the network. The 01 parameter indicates that all previous network information should be erased.

If you want to power-up an existing network, the parameter should be 00.

Starting the network will take 10 seconds or so. You will receive the following message in confirmation:

**+MJNC=000000**

The first pair of digits 00 indicates that the operation was a success. The last four 0000 are the short address of this device. If you are sniffing for messages, you should scan channels now to find out what frequency the coordinator is using.

19. Before the PixieDARC can join the network, the gateway must be told it may permit other devices to join. To do this, type the following into the Gateway:

**+MPJR=FF**

That tells the gateway to permit joining indefinitely. You will receive the following message in confirmation:

**+MPJC=00**

The pair of digits 00 indicates that the operation was a success.

20. Power up the PixieDARC with the Bind button pressed down. Holding the button down during power-up ensures any previous network information is erased. (To power up a previous network, do not hold the bind button down.) After a short while you should see the following appear at the gateway:

**+MPJI= 80004601293503026F79000000  
C815000015C80000000001**

**+MPRI= 6F79080000FFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFF**

The first message indicates that a device joined this node. The second message is broadcast to all MailBox devices on the network, indicating that device has joined the network. The first four digits 6F79 indicate that the device address is 796F. (Multi-byte values are little-endian, so read pairs of digits in reverse order.)

21. You can now send the DARC device instructions. For example, type the following into the Gateway:

**+MDAR=006F7903010101**

This is an instruction to set RA1 as output high. This is connected to the LED marked A5, and you should see the LED light and also get the following responses:

**+MDAC=00FF7A01**

**+MDAI=006F79FF0100**

The MDAC message confirms your message was sent. (Don't worry about the last 6 digits, just so long as the first two are 00 indicating success.) The MDAI message is from the DARC device. The final two digits, 00, indicate that the operation was successful.

22. Ensure the EP2 switch is in the low position and type the following into the Gateway:

**+MDAR=006F79020214**

This is an instruction to set RB4 as a digital input and read its value. You should get the following responses:

**+MDAC=00007801**

**+MDAI=006F7900020000**

The MDAC message confirms your message was sent. (Don't worry about the last 6 digits, just so long as the first two are 00 indicating success.) The MDAI message is from the DARC device. Right now you only need to be concerned with the last two digits, 00, which indicate that the value is logic 0. Try changing the switch position to high and resending the message. You should get a response similar to the following:

**+MDAI=006F7901020001**

The final two digits, 01, indicate that the value is logic 1.

23. To read an analog voltage, we will use the EP6 preset, which is connected to AN2. To use the preset, connect jumper pins A8 to B8.
24. To set inputs AN0 – AN2 as analog inputs, send the following command:

**+MDAR=006F79020403**

An MDAC message starting 00 confirms your message was sent. An MDAI message starting 00 will indicate that the DARC completed the operation successfully.

25. To read analog input AN2, send the following command:

**+MDAR=006F79020302**

This is an instruction to read AN2. An MDAC message starting 00 confirms your message was sent. You should also receive a response similar to the following:

**+MDAI=006F79030300E801**

The last four digits of the MDAI message indicate the voltage. Swap the pairs of digits around to see the hex voltage, in the range 0000 (0V) to 03FF (3.3V). In this example, the hex voltage is 01E8, or about 1.575V.

Try adjusting the preset and sending the message again.

26. To get the DARC device to take regular readings for you, type in the following command:

**+MDAR=006F7903050282**

This is an instruction to read and transmit the value of AN2 every 2 sleep cycles, or approximately 4 seconds. You will receive an MDAI message every 4 seconds with the preset value. Try adjusting the preset to see the output change.

Note that the sensor spends almost all of its time asleep and so this is a very low power device.

## Tutorial 6

Tutorial 6 demonstrates the MailBox Gateway and PixieDARC firmware. They use the StarLite firmware to provide IEEE 802.15.4 single-hop communications between nodes in a network.

The IEEE 802.15.4 protocol differs from ZigBee in that it does not implement routers; there is just a coordinator and end devices. End devices can sleep, and if necessary the coordinator can cache messages for them while they sleep.

StarLite is the simplest way to implement a transparent serial data communications. There are three transparent transmission modes, star mode and broadcast mode, and addressed mode.

In Broadcast mode, data is transmitted to all devices. Broadcast mode messages are not acknowledged, so no indication is given that a message was not received.

In Star mode, end devices transmit to the coordinator, and the coordinator communicates to the last end device to transmit to it. Star mode messages are acknowledged, so an indication can be given that a message was not received.

In Addressed mode, data is transmitted to a specific device whose address is specified. Star and addressed modes are more advanced and will not be covered in this tutorial.

StarLite is implemented on both the Pixies and the UZBee. This tutorial will require one Pixie Evaluation Kit and one UZBee. The UZBee will act as coordinator, since this is the most common configuration. One Pixie will be a fast end device and the other will be a sleepy end device. Ideally you would use two serial cables for this tutorial. If you only have one, you will need to switch it between boards as required.

## Reprogramming the UZBee

You will need to reprogram the UZBee as a StarLite USB coordinator. To do this:

1. Insert the UZBee into a USB socket with the recessed button pressed down. The LED should flash slowly. If a driver information file is requested, specify the *USBoot.inf* file from the StarLite USB development kit.
2. Determine the COM port which has been assigned to USBoot by selecting Start > My Computer > (right click) > Properties > Hardware > Device Manager > Ports. The Com port used by USBoot should be listed.
3. Run the *SLUC service pack.exe* application, which you will find in the StarLite USB development kit. The StarLite USB Coordinator will be installed. If a driver information file is requested, specify the *StarLite USB.inf* file, also from the StarLite USB development kit.
4. Determine the COM port which has been assigned to StarLite USB following the same process as in step 2.
5. The UZBee LED should be flashing because it automatically goes into configuration mode after programming. If the LED is not flashing because you haven't just performed the previous steps, press and hold the recessed button for five seconds until the LED flashes.
6. You are in configuration mode. Run the Packetizer application in StarLite USB development kit. Enter the StarLite USB COM port in the box provided and press Open.

7. Type the following into the Transmit box::

**#G^9UB33R0015C83841000000**

Then press *Tx Now*. A “<” character will appear in the Receive window. You have just specified the MAC address and reverted all other settings to their defaults. Most importantly, it puts the StarLite USB into broadcast mode. Note that the last 16 digits are the MAC address. If you intend to ship UZBees commercially, contact us for an allocation of unique MAC addresses.

8. Ensure you close the Packetizer application now.

## Reprogramming the Pixies

You will need to reprogram the Pixie as a StarLite Fast End Device and the Pixie Lite as a StarLite Sleepy End Device.

To reprogram a Pixie as a StarLite Fast End Device:

9. On your Windows PC, start up MPLAB and connect the ICD2 In-Circuit Debugger.
10. Remove the jumper pins A1-B1, A2-B2, A3-B3 from the board. Connect the RJ11 programming connector from ICD2 to the RJ11 socket marked IDC2 on the board. Power up the board.
11. Choose *File > Import* in the MPLAB menu. Open the file *SLPF.hex* from in the StarLite development kit. This is the firmware for the StarLite Fast End Device (Pixie).
12. Choose *Programmer > Program* to reprogram the Pixie. After a few moments you should get the message “Programming Succeeded”.
13. Turn off the power to the board, replace jumper pins A1-B1, A2-B2, A3-B3.
14. For RS232 connection, fit jumper A4-B4. For Pixie Config Tool connection (ZEVr4 boards and higher only), remove the jumper. Start a HyperTerminal session on the serial port with settings 9600 baud 8N1, hardware flow control.
15. Ensure the *Config SW* is in the *Low* position and ensure it is connected to the pin by fitting jumper A8-A9. This is the flow control. If it is high, you may not receive any messages from the board.
16. Ensure the *EP2* switch is in the *Low* position. This is the sleep input. If it is high, you will not receive any messages from the board.
17. Power up the board. If you have just programmed it, the status LED should be

flashing because it automatically goes into configuration mode after programming. If the LED is not flashing because you haven't just performed the previous steps, press and hold the Bind button for five seconds until the status LED flashes

18. A “>” character should appear in the HyperTerminal window. Type the following characters in the HyperTerminal window:

**#G^9UB933R0015C83841000001**

A “<” character will be received in response. You have just specified the MAC address and reverted all other settings to their defaults. Most importantly, it puts the StarLite into broadcast mode. Note that the last 16 digits are the MAC address. If you intend to ship Pixies with StarLite loaded commercially, contact us for an allocation of unique MAC addresses.

19. Turn off the board now until you are ready to use it in the next section, *Joining then Network*.
20. Repeat steps 10-19 for the Pixie Lite, except using the *SLLS.hex* file (Sleepy end device, Pixie Lite) and typing the following configuration string:

**#G^9UB933R0015C83841000002**

This specifies a different MAC address.

## Joining the Network

Since the end user might need to perform the joining step, we've tried to keep it simple. If you have a spare Pixie or UZBee, you may like to set it up as a sniffer.

21. The UZBee is the coordinator, so you will be joining devices to it. Press the recessed button on the UZBee. The LED will come on, indicating it is allowing a new end device to join.
22. Power up the Pixie evaluation boards with the button held down. The status light will come on while it searches for a new network to join. When it finds the UZBee coordinator, the LEDs on both devices will extinguish and they will be joined.
23. Power down and up the Pixie board. When it powers up, note how both LEDs flash to indicate they have successfully reconnected.
24. Repeat steps 21-23 for the Pixie Lite evaluation board.

## **Sending Data**

We have set up a broadcast network. No messaging is acknowledged, but packets are error checked, so a device will not output erroneous data.

25. Ensure you have a HyperTerminal associated with all devices.
26. Type data into any HyperTerminal window. The data should appear at all the others.



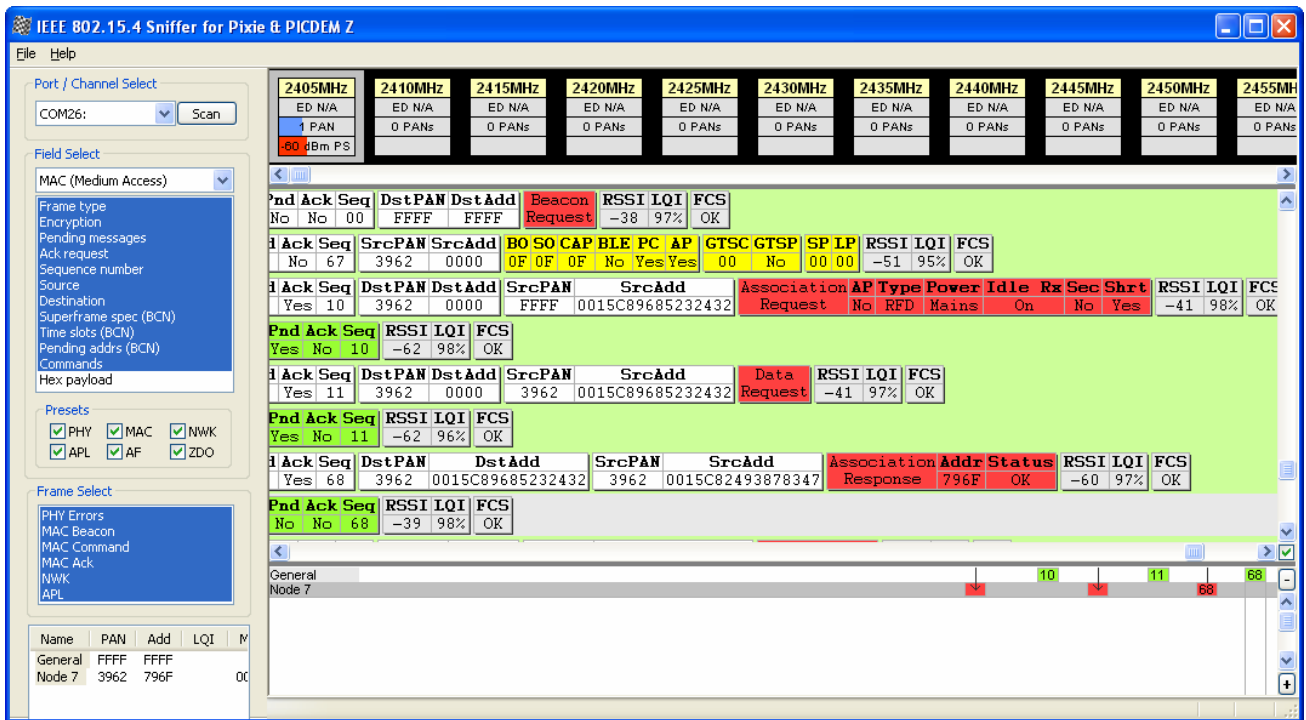


Exhibit A: Typical sniffer log when joining

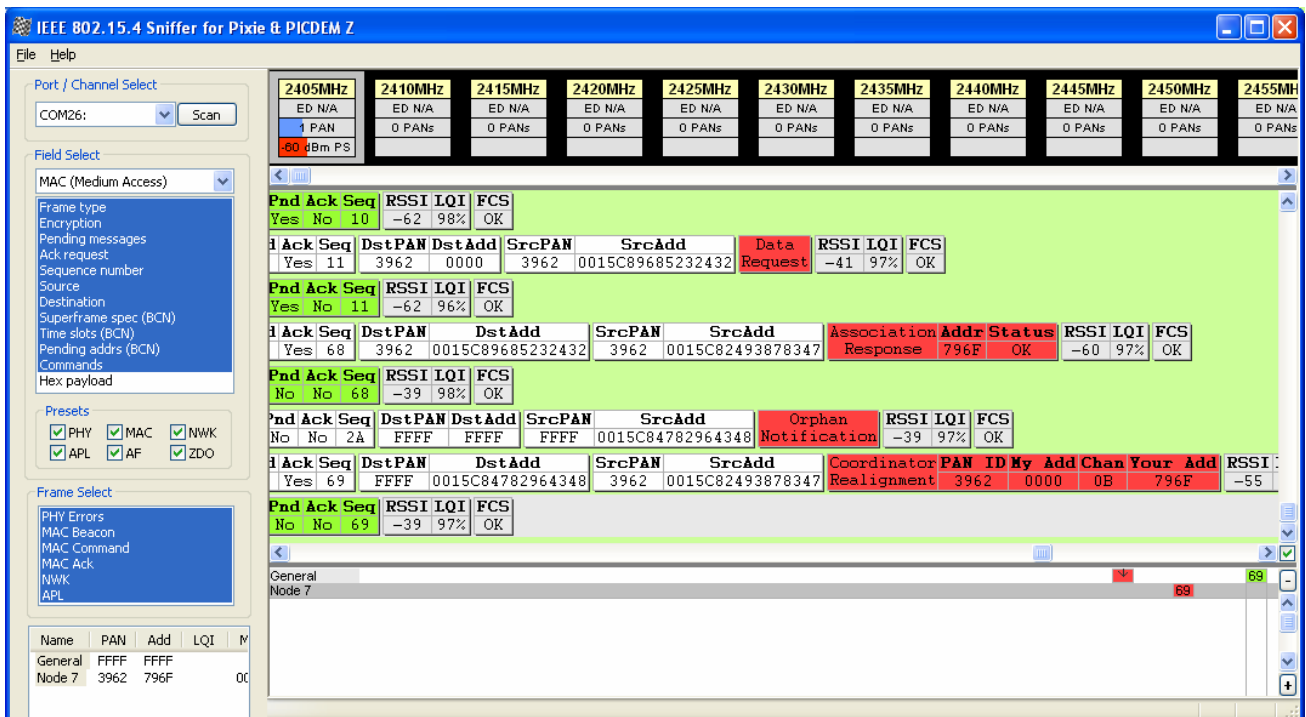


Exhibit B: Typical sniffer log when rejoining (from the "Orphan Notification" line onwards)

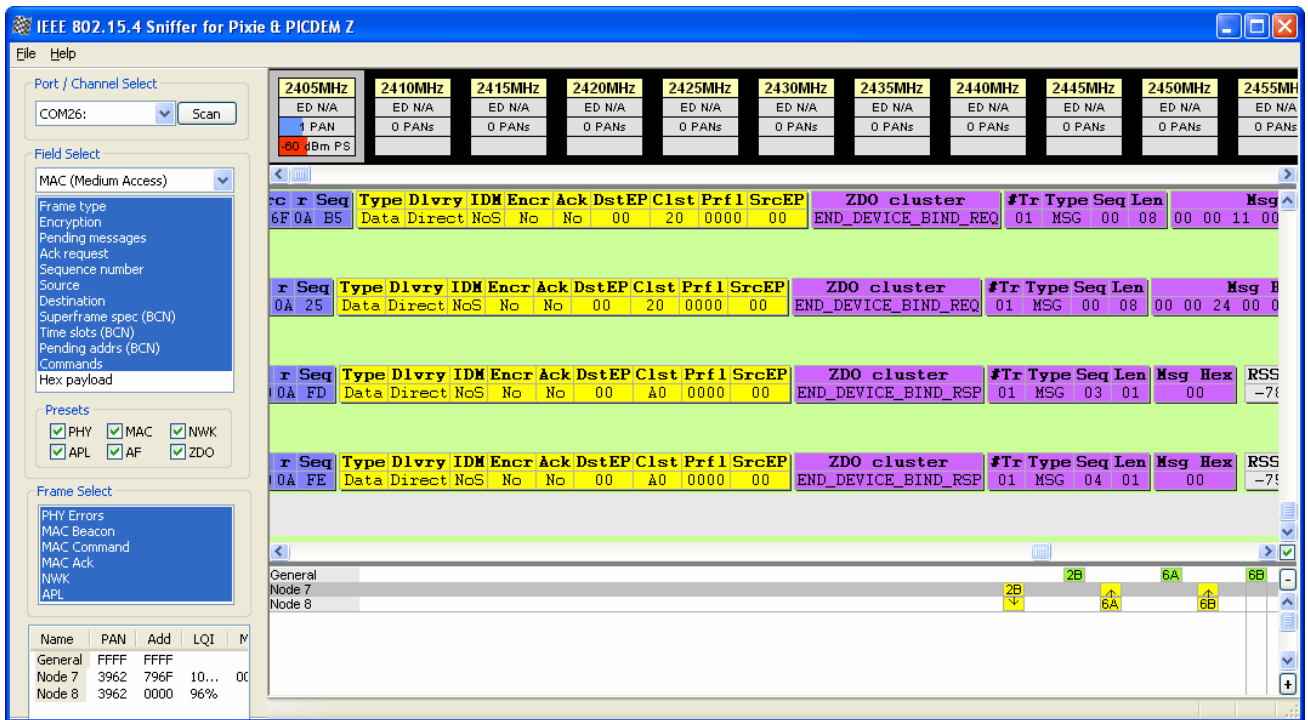


Exhibit C: Typical sniffer log when binding

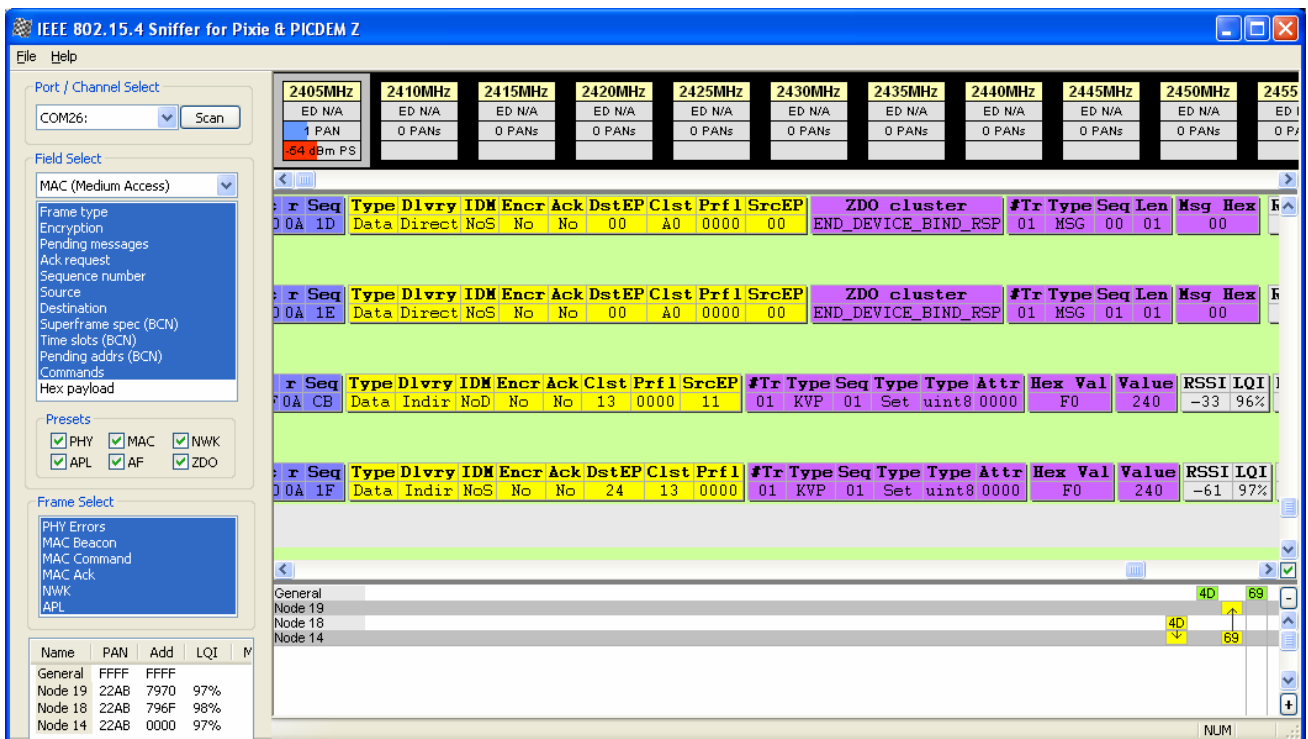


Exhibit D: Typical sniffer log when sending a switch message (from Seq CB onwards).

## Contact Details

### Sales

FlexiPanel ZigBee products are manufactured and distributed by:



R F Solutions Ltd  
Unit 21, Cliffe Industrial Estate,  
Lewes, E. Sussex BN8 6JL, United Kingdom  
*email* : [sales@rfsolutions.co.uk](mailto:sales@rfsolutions.co.uk)  
<http://www.rfsolutions.co.uk>  
*Tel*: +44 (0)1273 898 000 *Fax*: +44 (0)1273 480 661

### Technical Information

For technical support, contact FlexiPanel Ltd:



FlexiPanel Ltd  
2 Marshall St, 3rd Floor,  
London W1F 9BB, United Kingdom  
*email*: [support@flexipanel.com](mailto:support@flexipanel.com)  
[www.flexipanel.com](http://www.flexipanel.com)



# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[FlexiPanel:](#)

[EVAL-PIXIE](#)