ZigBee Board

User Manual

1.0, Oct 2011

Table of Contents

Chapter 1. Introduction

ZigBee is a low power, wireless mesh network standard, largely used in the areas of home automation, medical data collection and industrial control. The ZigBee Board connects to Zilogic's development kits with a UART-I²C header. In addition, the ZigBee Board can also be used with other deveolpment kits that have an RS-232 UART interface, using the ZKit RS232 transceiver [http://www.zilogic.com/zkit-51-rs232.html] and a null modem cable.

At the core of the ZigBee Board is the XBee module from Digi, which implements the ZigBee stack. The program running on the development kit can interact with the ZigBee stack, by sending commands through the UART interface.

1. Features

- Powered from motherboard
- Built in +3.3V regulator for XBee module
- XBee ZB and XBee-PRO ZB compatible
- LEDs for TX/RX and ZigBee state
- Connects through UART
- Standard TTL compatible inputs used in most microcontrollers and control systems
- 0.1" 10 pin FRC header for connection to control logic/MCU
- Ready to go with Zilogic's motherboards

2. Wireless Parameters

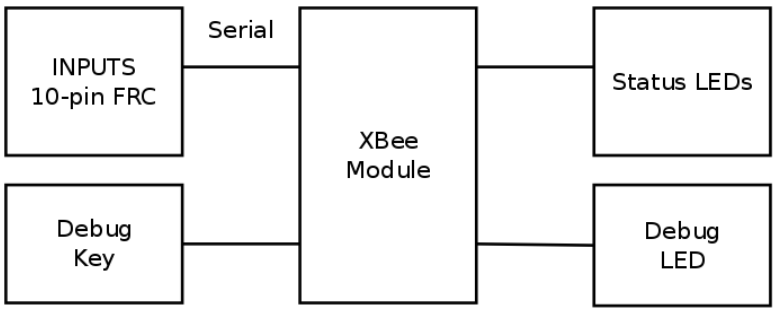
Parameter	XBee ZB	XBee-PRO ZB
Data rate	250Kbps	250Kbps
Indoor range	40m	90m
Line of sight	120m	1500m
Tx. Power	1.25mW / 2mW	63mW

Chapter 2. Board Design

1. Overview

The devices availabe on the board, is shown in the following block diagram. Each device is described in detail in the following sections.

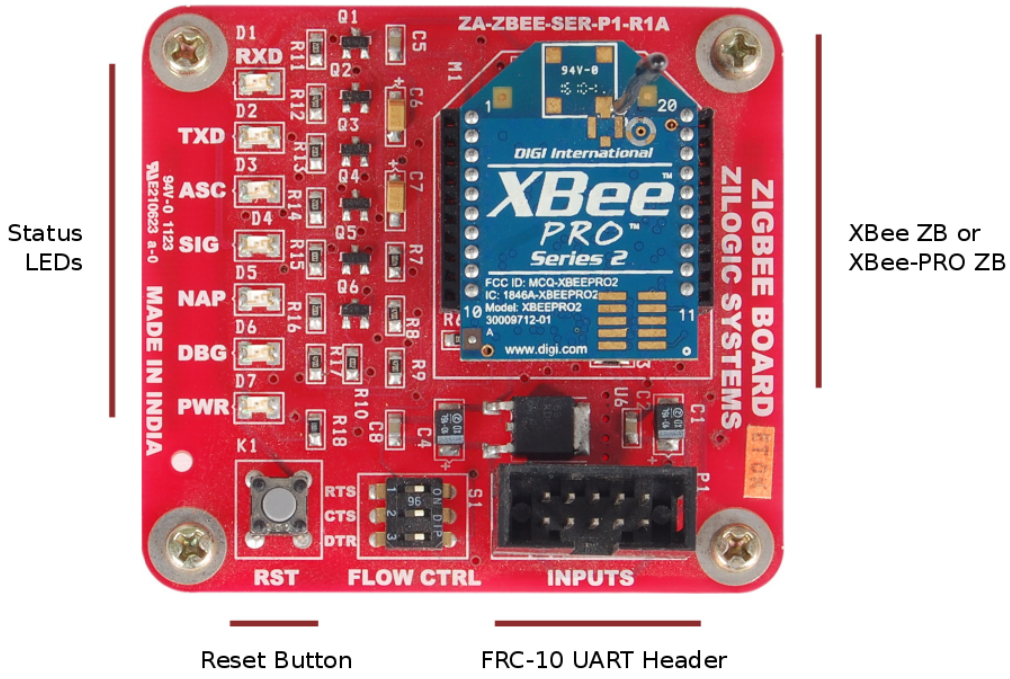
Figure 2.1. Block Diagram



2. Locating Components

The location of the components on the board is indicated in the following diagrams.

Figure 2.2. Front View



3. Power Supply

The Zigbee board is powered from the motherboard through th FRC header.

4. XBee Module

The XBee ZB Module or XBee-PRO ZB Module can be mounted in the ZigBee board. The module provides a UART interface through which the motherboard can send and receive data. The module can also be configured through the serial interface.

5. Status LEDs

The status LEDs indicate the following information.

RXD LED	Blink indicates data has been received by the module from the host.
TXD LED	Blink indicates data has been transmitted by the module to the host.
ASC LED	Continuos blinking indicates associated with a network, in case of a router or end device. And PAN ID and radio channel has been selected in the case of coordinator.
SIG LED	Brighter LED indicates a more stable link. In other words, shows the received signal strength of the last received packet.
NAP LED	On indicates device has woken up from sleep state.

6. Debug LED

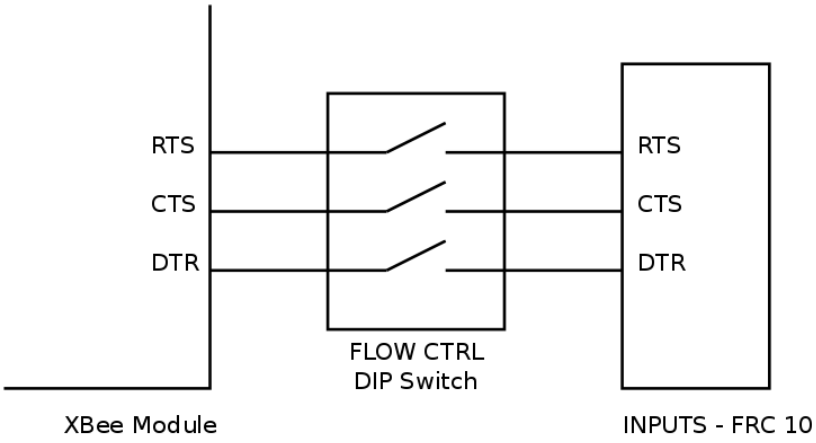
A Debug LED is provided for testing IOs. The Debug LED is connected to the pin DIO4 of the XBee module.

7. Flow Control DIP Switch

The RTS, CTS and DTR signals can be connected or disconnected from the FRC header using the FLOW CTRL DIP switch.

It is a good idea to use flow control, to prevent data loss. If flow control is required, then flow control should be enabled in the module using the D6 and D7 AT commands. And the RTS and CTS DIP switches should be in ON position.

Figure 2.3. Flow Control DIP Switch



8. Connectivity

The Zigbee board can interface with the UART connector on the motherboard using its 10 pin INPUTS FRC connector. The signal details are given below.

Table 2.1. FRC-10 Connector

Pin #	Signal	Signal Type
1	5V	Supply from motherboard

Pin #	Signal	Signal Type
2	TXD	TTL Out
3	RXD	TTL In
4	Not Connected	-
5	Not Connected	-
6	RTS	TTL In
7	CTS	TTL Out
8	DTR	TTL In
9	Not Connected	-
10	GND	Ground

Chapter 3. Board Usage

1. Programming XBee Firmware

The XBee module should be programmed with the appropriate firmware using the `XCTU` tool provided by Digi. Firmware programming can be done using the `USB ZigBee Adapter` board from Zilogic Systems. The instructions for programming the firmware is provided in the `USB ZigBee Adapter` manual.

2. AT Mode and API Mode

The XBee modules can communicate using two protocols with the host.

1. AT Mode
2. API Mode

In AT mode, the module provides an AT command interface, similar to AT modem commands, for configuration. The AT mode also supports transparent mode of operation in which serial data sent to the module, is transmitted over the air to a remote ZigBee node.

In API mode, the module uses a binary packet interface for communicating with the host. The available packets and their formats is described in the module user manual.

3. Communicating in AT mode

The module is first updated with an AT mode firmware. AT commands can then be sent to the device through the UART interface.

The serial parameters are specified in the following table.

No.	Parameter	Value
1	Baudrate	9600
2	Data Bits	8
3	Stop Bits	1
4	Parity	None
5	Flow Control	None

The rest of the discussion will assume the availablility of the following functions:

`write(s)` Writes the string `s` to the serial port.

`readln()` Reads and returns a line from the serial port. A line is terminated by a `\r` character.

`mdelay(t)` Generates a delay for `t` msecs.

By default the module will be in transparent mode of operation, to enter AT command mode use the following sequence.

```
mdelay(1000);
write("+++");
readln() == "OK\r";
```

The module on entry into AT command mode, responds with `OK\r`. After which AT commands can be sent to the device. For example to get the firmware version, the `VR` command can be used.

```
write("ATVR\r");
```

```
readln() == "2070\r";
```

The character + used for entering the AT command mode, is called the Command Character. The guard time and command character is configurable using AT commands. The default is 1 second and +, respectively.

The module can be put back to transparent mode of operation manually using the CN AT command. The module also switches back to transparent mode of operation after a timeout, 10 seconds by default. The command mode timeout is also configurable using the CT AT command.

4. Sending and Receiving Data in Transparent Mode

Transparent mode of operation is demonstrated using a two node network, consisting of a coordinator and router. The following sequence shows how to send a message from the coordinator to the router.

1. Update the coordinator AT firmware in module A, and router AT firmware in module B.
2. Connect the ZigBee Boards to the motherboard. Ensure the ASC LED is blinking in both the modules.

Figure 3.1. Two Node Network

3. Note down the 64-bit MAC address specified in the module B. The 64-bit MAC of the module can be obtained from the label on the back of the module. The 64-bit MAC can also be obtained using the SH and SL AT commands. Lets assume the routers MAC address is 0011223344556677
4. In module A, set the destination addressing using the DH and DL AT commands. DH is used to specify the higher order 32 bits and DL is used to specify the lower order 32 bits.

Figure 3.2. Setting the Destination

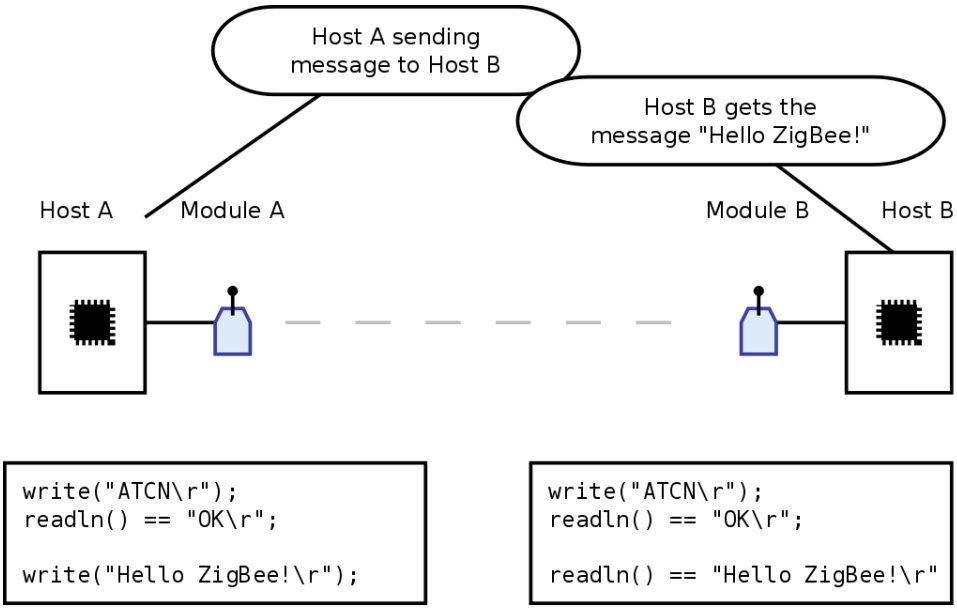

```
write("+++");
readln() == "OK\r";

write("ATDH 00112233\r");
readln() == "OK\r";

write("ATDL 44556677\r");
readln() == "OK\r";
```

5. Switch both the module into transparent mode of operation, using the CN AT command. Type a message to be sent to module B, the message should appear on module B's serial output.

Figure 3.3. Sending and Receiving Message



Appendix A. Legal Information

1. Copying

2. Disclaimers

NO WARRANTY. ZILOGIC SYSTEMS' DEVELOPMENT KITS (AND TECHNICAL SUPPORT, IF ANY) ARE PROVIDED "AS IS" AND WITHOUT ANY WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAWS, ZILOGIC SYSTEMS EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. ZILOGIC SYSTEMS DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN ZILOGIC SYSTEMS' DEVELOPMENT KITS WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN ZILOGIC SYSTEMS' DEVELOPMENT KITS WILL BE CORRECTED. FURTHERMORE, ZILOGIC SYSTEMS DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE ZILOGIC SYSTEMS' DEVELOPMENT KITS IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY OR MAY BE LIMITED.

Limitation of Liability. Zilogic Systems' development kits are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, not in applications where failure or malfunction of a Zilogic Systems product can resonably be expected to result in personal injury, death or severe property or environmental damage. Zilogic Systems accepts no liability for inclusion and/or use of Zilogic Systems' development kits in such equipment or applications and therefore such inclusion and/or use is at the customer’s own risk.