Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

XBee SIP Adapter (#32402)

The XBee SIP Adapter comes fully assembled and provides a small-footprint solution for interfacing your microcontroller to any XBee or XBee-Pro module. A 3.3 volt regulator and 74LVC244A buffer on board provide safe interfacing to a 5 volt supply and easy compatibility with any Parallax microcontroller.

The 2 x 5 dual SIP header makes a sturdy connection to your breadboard or through-hole board, and brings the basic connections to your prototyping area. The more advanced XBee features are still accessible, through an additional header and plated through-holes on the board.

Two very simple example applications that are compatible with all BASIC Stamp 2 models are included in this document.

Features

- Onboard 3.3 V regulator

- 5 V to 3.3 V logic translator buffers common I/O pins

- Six status indicator LEDs for Power, Tx, Rx, RSSI, Associate and mode (Sleep/ON)

- Small footprint dual SIP header provides support and allows easy interfacing to DOUT (TX), DIN (RX), RTS, 5 V supply and ground

- 5-pin female header connections provides interfacing to other XBee pins such as sleep, reset and associate

- A row of 10 plated through-holes with 01″ spacing allows the option of soldering jumper wires or a header (not included) for access to the remaining XBee pins in advanced applications

- An additional plated through-hole gives access to 3.3 V output for ADC reference (VREF) when required

- Adapter board is pre-assembled—no soldering is required for using most common XBee features

- Compatible with all Parallax microcontrollers, including the 5 V BASIC Stamp modules and 3.3 V Propeller P8X32A

Key Specifications

- Power Requirements5 VDC

- Communication: Serial pass-through to XBee module

- Operating temperature: -40 to + 158 °F (-40 to +70 °C)

- Dimensions: 1 x 2.4 x 0.36 in (25.4 x 61 x 9.14 mm)
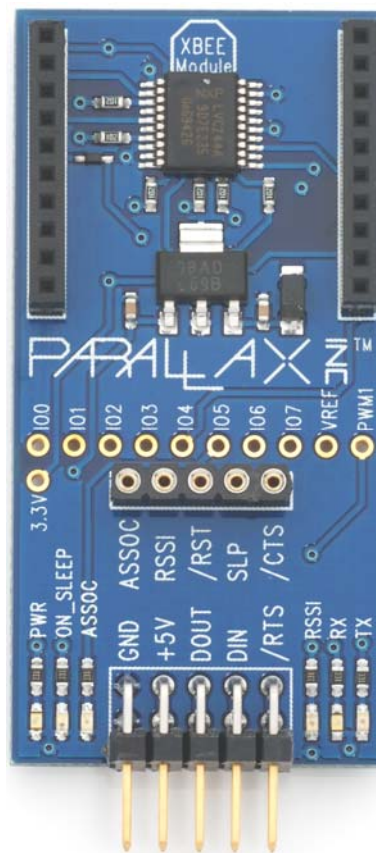
Pin Definitions
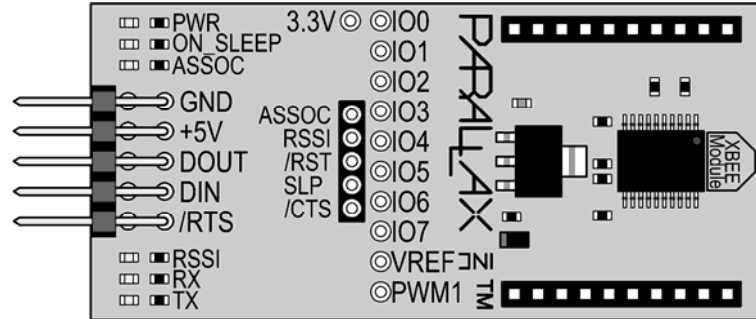
5-Pin Dual Row Male Header

Pin	Name	Type	Function
1	GND	G	Ground
2	+5V	P	5 V supply
3	DOUT	O	Serial Data output from XBee
4	DIN	I	Serial Data input to XBee
5	/RTS	I	Ready-to-Send input to XBee for flow control when configured

Pin Type: P = Power, G = Ground, I = Input, O = Output

5-Pin Female Header

Pin	Name	Type	Function
1	ASSOC	O	Associate indication output
2	RSSI	O	Received Signal Strength Indicator PWM output
3	/RST	I	Low-level reset to XBee
4	SLP	I	Pin-Sleep input to XBee when configured
5	/CTS	O	Clear-to-Send from XBee when configured

Pin Type: P = Power, G = Ground, I = Input, O = Output

11 Plated Through-holes

CAUTION: These connections are NOT buffered. Connections to 5 V may damage the XBee module. Use of these through-holes to access XBee pins is recommended for advanced users only.

Pin	Name	Type	Function
1	IO0	I/O	AD0/DIO0 – Analog input and digital I/O
2	IO1	I/O	AD1/DIO0 – Analog input and digital I/O
3	IO2	I/O	AD2/DIO0 – Analog input and digital I/O
4	IO3	I/O	AD3/DIO0 – Analog input and digital I/O
5	IO4	I/O	Associate Indicator, AD0/DIO0 – Analog input and digital I/O
6	IO5	I/O	AD5/DIO0 – Analog input and digital I/O
7	IO6	I/O	/RTS, AD6/DIO0 – Ready to send, Analog input and digital I/O
8	IO7	I/O	AD7/DIO0 – Analog input and digital I/O
9	VREF	I	Analog reference input
10	PWM1	O	PWM output
11	3.3V	P	3.3 V Output for ADC reference (VREF) when required

Pin Type: P = Power, G = Ground, I = Input, O = Output

LED Indicators

| LED Name	Indication
PWR	Indicates power available, supplied from 3.3 V on-board regulator
ON_SLEEP	Indicates sleep status of XBee, On = awake
ASSOC	Indicates status when using association feature of XBee to join networks. Solid = Associated, Blinking = Not Associated. Most users manually setup the network instead of associating.
RSSI	Receive Signal Strength Indicator. PWM controlled. Lights for 5 seconds following reception of RF data. A dimming may be noticeable with poor signal strength.
RX	Indicates data received by BASIC Stamp from the XBee (DOUT pin) due to received RF data or command communications with XBee.
TX	Indicates data sent to the XBee (DIN pin) from the BASIC Stamp for RF transmission or command communications.

I/O Pin Buffering

XBee RF Modules are 3.3 V devices. The XBee SIP Adapter provides a means to interface the most frequently used XBee functions with a BASIC Stamp or other 5 V microcontrollers. All I/O on the male and female 5-pin headers are buffered; inputs to the XBee are buffered to translate 5 V to 3.3 V, while outputs from the XBee are buffered to protect the XBee I/O.

CAUTION: the 11 plated through-holes are NOT buffered. Connections to 5 V using these through-holes may damage the XBee. Use of these through-holes to access XBee pins is recommended for advanced users only. For more information on XBee configuration and I/O uses, please consult Digi International's XBee manual.

Communication Protocol

Communications interfacing between the BASIC Stamp and XBee uses non-inverted asynchronous serial data at 9600 bps, 8-N-1. The baud rate of the XBee is configurable in Command Mode. Please see the example programs and the Digi's XBee documentation for more information.

Module Dimensions
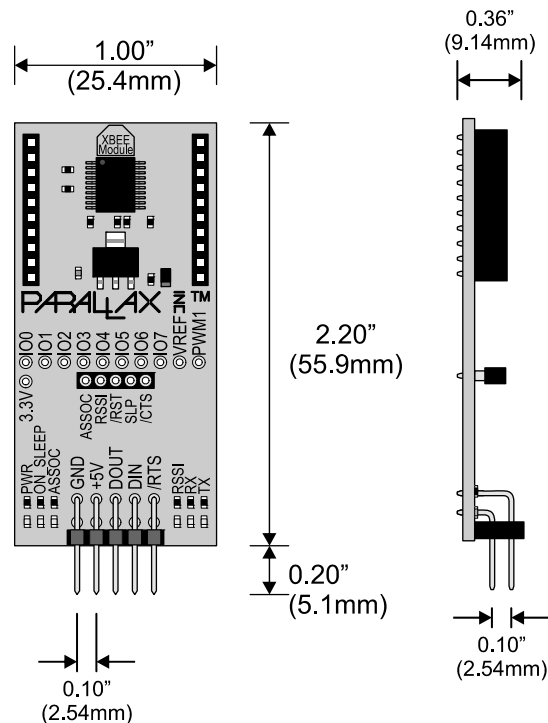
BASIC Stamp® Example Applications

The following two simple example applications use two BASIC Stamp 2 microcontroller modules, two XBee SIP Adapters, and two XBee modules. You may use any BASIC Stamp 2 series modules; since the example code uses conditional compilation it is not necessary to use two of the same model.

It IS necessary to use two XBee modules from the same series; Series 1 and Series 2 XBee modules are not cross-compatible.

Connection Diagrams

Transmitter Configuration

Receiver Configuration
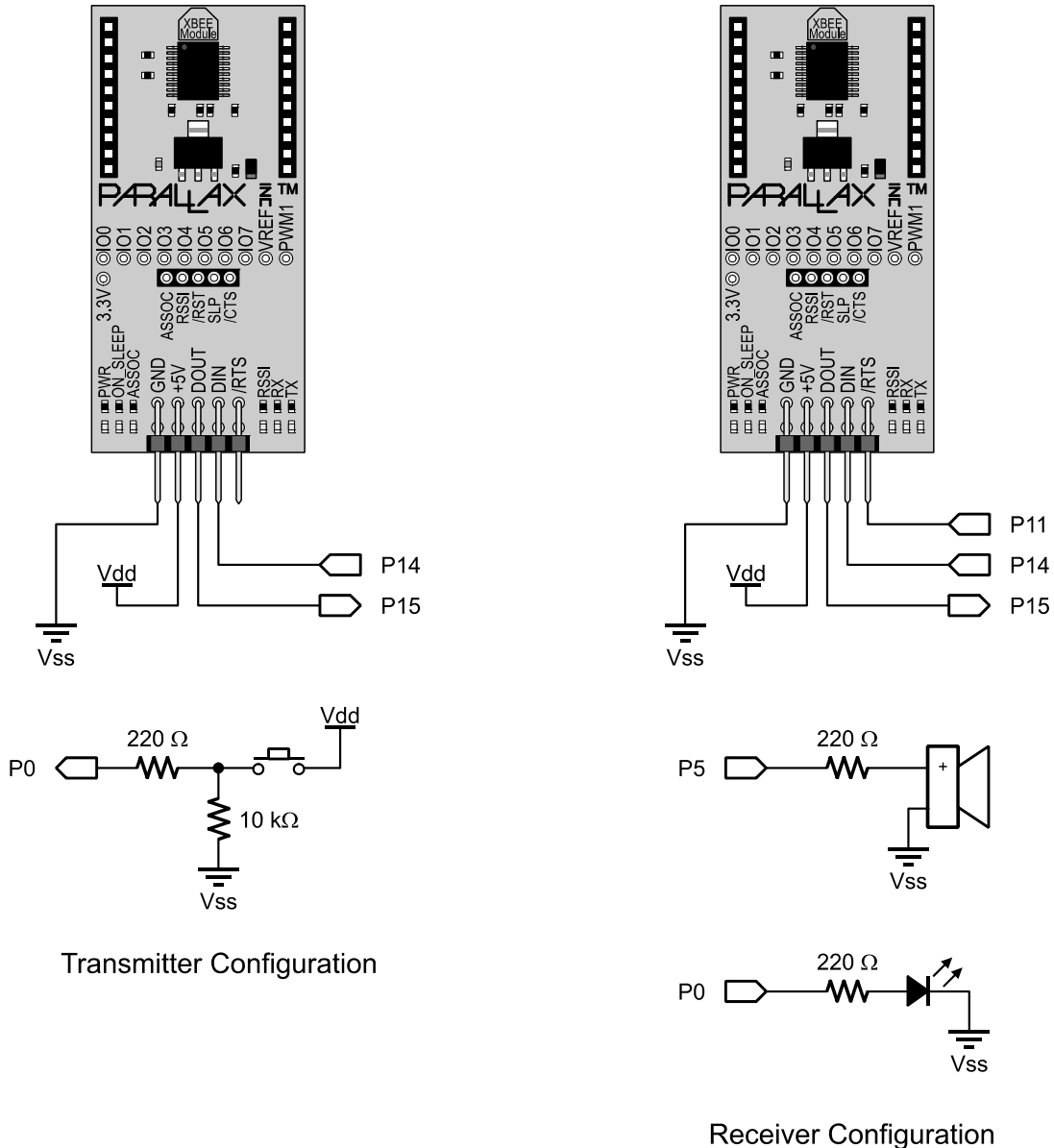
Example 1: Single Byte Transmission

This example illustrates sending a single byte between BASIC Stamp modules. The transmit code (Simple_Byte_Tx.bs2) will send the value of the pushbutton as a byte. The receiver (Simple_Byte_Rx.bs2) will accept the byte. If byte value is 1, the LED will light and sound buzzer. If 0, the LED will turn off and not sound the buzzer.

Transmitter Code

```
' **********************************************
' Simple Byte Tx.bs2
' Sends the state of pushbutton every 250mSec
' **********************************************

' {$STAMP BS2}
' {$PBASIC 2.5}

#SELECT $STAMP
  #CASE BS2, BS2E, BS2PE
    T9600        CON     84
  #CASE BS2SX, BS2P
    T9600        CON     240
  #CASE BS2PX
    T9600        CON     396
#ENDSELECT


' ***************** Variables, Constants and Pins

Baud            CON     T9600  ' Set baud rate

Rx              CON     15     ' XBee DOUT
Tx              CON     14     ' XBee DIN

PB              PIN     0      ' Pushbutton

State           VAR     Bit


' **************** Main Loop

DO
 State = PB                    ' Read pushbutton
 SEROUT Tx, Baud,[State]       ' Send pushbutton value as byte
 PAUSE 250                     ' short delay
LOOP
```

Receiver Code

```
' ***************************************************
' Simple Byte Rx.bs2
' Receives byte value (0/1) to control LED and buzzer
' ***************************************************

' {$STAMP BS2}
' {$PBASIC 2.5}

#SELECT $STAMP
  #CASE BS2, BS2E, BS2PE
    T9600        CON     84
  #CASE BS2SX, BS2P
    T9600        CON     240
  #CASE BS2PX
    T9600        CON     396
#ENDSELECT
```

```
' ***************** Variables, Constants and Pins

Baud            CON     T9600 ' Set Baud rate

Rx              CON     15    ' XBee DOUT
Tx              CON     14    ' XBee DIN

State           VAR     Bit

Led             PIN     0
Buzzer          PIN     5


' **************** Main Loop

DO
 SERIN Rx, Baud, [State]        ' Wait for byte and accept
 IF State = 1 THEN              ' Based on value,
    HIGH LED                    ' if 1, Turn on LED
    FREQOUT Buzzer,200,3000     ' Sound buzzer
 ELSE
    LOW LED                     ' if 0, turn off LED
 ENDIF
LOOP
```

Example 2: Multiple Decimal Values with Addressing & Flow Control

This example illustrates sending multiple decimal values between BASIC Stamp modules along with using RTS flow control on the receiver and addressing of nodes through the XBee Command Mode. The transmitting hardware uses the BASIC Stamp Editor's Debug Terminal. The receiving hardware may be connected to a Debug Terminal as well, but it is not required. Multiple receivers may be configured to test addressing.

Transmitter Code:

This code requests data from the user via the Debug Terminal to control an LED and buzzer connected to a remote BASIC Stamp. Major actions of the code:

- Configures the XBee by entering AT Command Mode and sending the AT Command to set the guard time to a low value. This allows for quickly entering AT Command mode to change the destination address of a packet "on-the-fly."
- Requests from the user the destination address of data (remote node address). *In testing, use the address set in receiver code (1 by default), a non-assigned address, or use the broadcast address of FFFF to send to all receivers.*
- Requests from the user the remote LED state (0/1) and remote frequency to sound.
- Enters AT Command mode to quickly set the destination address of the data packet.
- Sends a start delimiting character (!) and the two decimal values for LED state and buzzer frequency.

```
' *******************************************************
' Multiple_data_with_Config_Tx.bs2
' This program:
'   - Configures XBee for fast AT Command Mode
'   - Requests destination address, LED state
'     & buzzer frequency in Debug Terminal
'   - Sets address and sends start delimiter (!)
'     and data to selected node address
'   - Requires 802.15.4 XBee (Series 1)
' *******************************************************

' {$STAMP BS2}
' {$PBASIC 2.5}
```

```
#SELECT $STAMP
  #CASE BS2, BS2E, BS2PE
    T9600      CON      84
  #CASE BS2SX, BS2P
    T9600      CON      240
  #CASE BS2PX
    T9600      CON      396
#ENDSELECT


' ***************** Variables, Constants and Pins

Baud            CON      T9600  ' Set Baud rate

Rx              CON      15     ' XBee DOUT
Tx              CON      14     ' XBee DIN

Freq            VAR      Word   ' Frequency to send
State           VAR      Bit    ' State of remote LED
DL_Addr         VAR      Word   ' Destination address for data


' ***************** Configure XBee in AT Command Mode

PAUSE 500
DEBUG CLS,"Configuring XBee..."

PAUSE 3000                      ' Guard time
SEROUT Tx,Baud,["+++"]          ' Command Mode Sequence
PAUSE 2000                      ' Guard time
SEROUT Tx,Baud,["ATGT 3",CR]    ' Set low guard time
SEROUT TX,Baud,["ATCN",CR]      ' Exit Command Mode


' ***************** Main Loop

DO
  ' Request address, LED state and frequency in DEBUG
  DEBUG CLS,"Enter Node Address in Hex (1-FFFF):"
  DEBUGIN HEX DL_Addr
  DEBUG CR,"Enter LED State (0/1):"
  DEBUGIN DEC State
  DEBUG CR,"Enter Frequency:"
  DEBUGIN DEC Freq

  ' Configure XBee for destination node address
  PAUSE 10                                   ' Short guard time
  SEROUT Tx,Baud,["+++"]                     ' Command Mode sequence
  PAUSE 10                                   ' Short guard time
  SEROUT TX,Baud,["ATDL ", HEX DL_Addr,CR]   ' Set Destination Node Address
  SEROUT Tx,Baud,["ATCN",CR]                 ' Exit Command Mode

  ' Send Data - Extra CR's help ensure data accepted properly
  SEROUT Tx,Baud,["!",CR,CR]                 ' Send start delimiter
  SEROUT Tx,Baud,[DEC State,CR,CR]           ' Send LED state
  SEROUT Tx,Baud,[DEC Freq,CR,CR]            ' Send buzzer freq
  DEBUG "Data Sent!",CR

  PAUSE 2000
LOOP
```

Receiver Code:

This code receives data consisting of a start delimiter (!), LED state and buzzer frequency. It uses RTS flow control to allow the XBee only to send received data to the BASIC Stamp when it is ready for it. While DEBUG is used, monitoring is not necessary in the Debug Terminal. Major actions of the code:

- Configures the XBee using AT Command Mode to enable RTS flow control and to set the node's MY address. *Modify this address as desired (My_Addr constant) from $1 to $FFFE to test. Multiple receivers may be on the network. Use this address for the transmitting BASIC Stamp when requested in Debug Terminal.*
- Waits for a byte with timeout. If the byte is the start delimiter (!), accepts decimal values for LED state and buzzer frequency. The use of the timeout illustrates that other actions may be occurring on your BASIC Stamp since data is buffered on the XBee. In this example a dot will be displayed for each timeout.
- Controls the LED and buzzer as specified by the data received. Displays the values for the user if the Debug Terminal is used.

```
' *******************************************************
' Multiple_data_with_Config_Rx.bs2
' This program:
'   - Configures XBee for address (Modify Address below)
'     and to use RTS flow control
'   - Accepts LED state & buzzer frequency
'   - Sets LED state and sounds tone on buzzer
'   - Requires 802.15.4 XBee (Series 1)
'   You may monitor in Debug Terminal, but not required
' *******************************************************

' {$STAMP BS2}
' {$PBASIC 2.5}

#SELECT $STAMP
  #CASE BS2, BS2E, BS2PE
    T9600       CON     84
  #CASE BS2SX, BS2P
    T9600       CON     240
  #CASE BS2PX
    T9600       CON     396
#ENDSELECT


'****************** Variable, Constants and Pins

Baud            CON     T9600

LED             PIN     0
Buzzer          PIN     5

Rx              CON     15   ' XBee DOUT
Tx              CON     14   ' XBee DIN
RTS             CON     11   ' XBee RTS

Freq            VAR     Word ' Received frequency for buzzer
State           VAR     Bit  ' Received state of LED
DataIn          VAR     Byte ' General byte data

My_Addr         CON     $1   ' Set address of node, modify as desired, $1-$FFFE


' ****************** Configure XBee to use RTS and set Address

PAUSE 500
DEBUG CLS,"Configuring XBee...",CR
PAUSE 3000                      ' Guard time
SEROUT Tx,Baud,["+++"]          ' Command Mode Sequence
PAUSE 2000                      ' Guard time
SEROUT Tx,Baud,["ATD6 1",CR]    ' Enable RTS
```

```
SEROUT Tx,Baud,["ATMY ", HEX My_Addr,CR]  ' Set node address
SEROUT TX,Baud,["ATCN",CR]      ' Exit Command Mode
DEBUG "Awaiting data..."


' ***************** Main Loop

DO
  SERIN Rx\RTS,Baud,10,Timeout,[DataIn]      ' Accept byte
  IF DataIn = "!" THEN                       ' Is start delimiter?
    SERIN Rx\RTS,Baud,1000,Timeout,[DEC State]' Accept LED state
    SERIN Rx\RTS,Baud,1000,Timeout,[DEC Freq] ' Accept buzzer frequency

    DEBUG CR
    DEBUG ? state            ' Display data
    DEBUG ? freq
    DEBUG CR

    IF State = 1 THEN        ' Set LED based on State
      HIGH 0
    ELSE
      LOW 0
    ENDIF

    FREQOUT 5,1000,Freq      ' Sound buzzer based on frequency
  ENDIF

Timeout:
DEBUG "."                    ' Show dot while waiting for data
LOOP
```