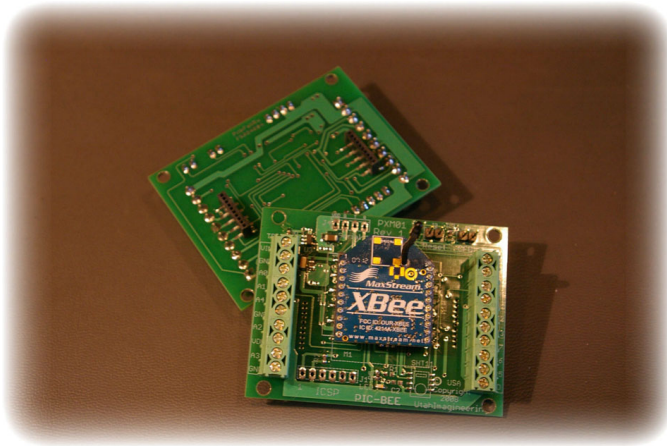


# Pic-Bee

## The simple yet powerful interface for the XBee radio

### Specifications



- Small size only 2.5" x 2"
- 18F4321 processor
- onboard voltage regulation (4-12VDC)
- Mounting holes 0.130" (3.3mm or #4 screws)
- TTL level serial interconnect (optional)
- XBee factory reset jumper
- XBee forced addressing jumper
- User programmable (PICKIT or compatible)
- No code required ! Fully functional command interface embedded in the processor.

The PIC-BEE is the first XBee adapter and interface board that embeds a command interface using simple ASCII commands to monitor and control up to five analog inputs and a combination of up to 13 digital I/O points. Terminal blocks on each edge provide the interface for input power (4-12VDC), 3.3VDC and grounds for analog and digital interfacing (analog range is limited to 0 - 3.3VDC).

Using a host system that interfaces to the XBee radio, the following commands are available (command preceded by the ASCII "%" and terminated with a return):

A : control analog or digital settings of PortA	M : timer module minutes setting
B : control the direction of PortB	P : puts a value on portB, for those set to output
C : XBee command mode	Q : query mode – polling (1) or Periodic reports(0)
G : Gets the values on PortA and PortB	S : timer module seconds setting
H : timer module hours setting	V : reports back software version
L : LED control	Z : controls sleep mode 1 of the XBee

For more information on the commands refer to <http://www.pic-bee.com>

# PIC-BEE Installation and Operation Instructions

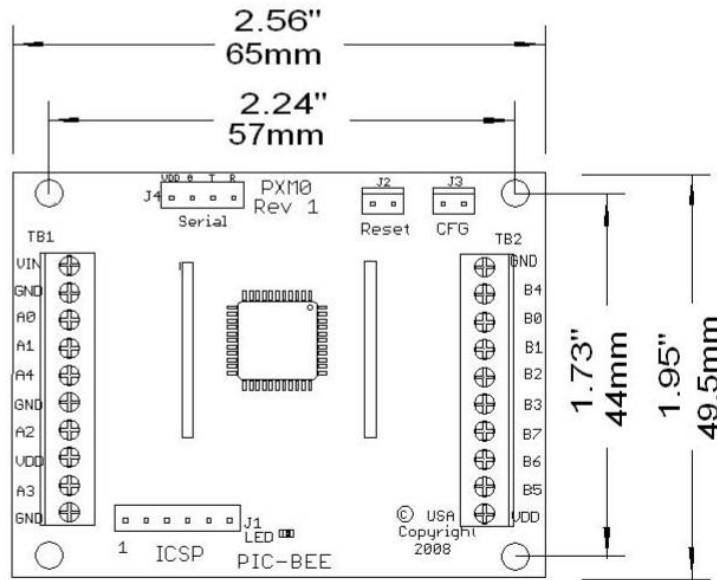
The PIC-BEE module creates a simple interface between the Maxstream/Digi XBee module and real world data acquisition and control. At the center of the design is a Microchip PIC processor that communicates with the XBee and transfers data between its digital and analog interfaces over the Zigbee RF link.

The simple nature of using any terminal program to communicate with the Xbee makes this product very easy to use. The ASCII commands can be embedded into any embedded or stand alone computing system. The baud rate is set for 9600/1/N as is the default mode for the Xbee when received from the factory. Although you can change the baud rate in the Xbee, it does not change for the PICBEE, so that is essentially a fixed rate. However for most data transmissions, that speed is sufficient.

It is assumed the user is familiar with the XBee products and operation. This document may refer to commands and functions of the XBee products, and can be used in conjunction with the Digi/Maxstream documentation.

## Installation and Connections

Referring to figure below, there are four mounting holes that can accept a #4 screw. The two 12 position terminal blocks can accept wire sizes from 16AWG to 26AWG.



DC power from 12VDC to 4VDC is connected to the terminals VIN and GND on TB1.

TB1 also supports the five analog inputs A0 thru A4. The range of the analog inputs are the Vdd power (typically 3.3VDC) to ground. These inputs can also be configured as digital inputs (see commands). In addition to the analog inputs are two grounds and a VDD to use with the analog devices.

TB2 supports the eight bits of digital input and outputs. These can be configured as either inputs or outputs. The inputs can be referenced to the board ground and 3.3VDC supply that are located on the terminal block. Inputs can be simple switch contacts, or other solid state sensors such as optical and infrared sensors. Outputs can either sink (current drives into the board) or source (takes current from the board) outputs. It is usually preferable to sink current into a device, rather than sourcing it. For turning on simple devices such as LED's, they can be connected directly (with a resistor of course) to the terminal block digital output and the power connection. For activating larger devices such as relays, you will want to use a transistor or FET driver, as they can switch the higher current that is required for those devices.

## Configuration

There are two jumper blocks on the assembly and are only monitored when the board is powered up. In the event the XBee module will not respond due to mis-configuration, or an unknown configuration, placing a shorting block across the 'RESET' block will send the factory reset command to the XBee and reset all the parameters. Shorting the CFG block, will temporarily force the DL value to be a value of 0xFFFF and a MY=0x00. This provides a communications link that can be reserved to be a configuration address only within a system, and allows the host to send configuration commands to the XBee. See the command instruction below for further information.

## Optional Connectors

There are a number of optional connection points that can support various communications interfaces :

- TTL serial (power,ground,serial in,serial out) thru a 4 pin header
- ICSP – in circuit serial programming using the Microchip PICKIT2 or ICD2 6 pin header
- I2C/SPI – serial communications on a backside connector pads. The connector that mounts on these pads is a Samtec RSM-110-02-S-S or equivalent. The mating connector on the target board would be a Samtec TMS-110-53-SM-S or equivalent.

## Operation

When the module is first powered up, the LED will turn on as it reads several registers from the XBee module, and transmit them over the Zigbee RF link. If a host is setup with the proper settings to communicate with the module, the following data will be displayed which provides the current addressing information. If the module cannot communicate with the XBee, the LED will begin to flash rapidly indicating an error condition.

```
.PIC-BEE V1.0  
[MY=0] [ID=3332] [DH=0] [DL=0]
```

Note: The values shown are only representative of typical values, and actual may be different depending on how the XBee module is configured.

All commands to the module are prefaced with a "%" character, and terminated with a carriage return (hex 0x0D) The following is a list of valid commands:

- A:** Control analog or digital settings of PortA
- B:** Control the direction of PortB
- C:** Xbee command mode
- G:** Gets the values on PortA and PortB
- H:** Timer module hours setting
- L:** LED control
- M:** Timer module minutes setting
- P:** Puts a value on portB, for those set to output
- Q:** Query mode – polling (1) or Periodic reports
- S:** Timer module seconds setting
- V:** Reports back software version

**%A** PortA on the micro controller can either be a digital or analog input (it cannot be an output port). Sending the command by itself will return a hex value that indicates the current settings for Port A of the micro controller, with a '1' indicates a digital while a '0' represents an analog input. The setting is stored in non-volatile memory such that on power cycles, it will retain its settings. Sending a ASCII value with the command will force a setting change for it. The valid range is 0xA thru 0xF per the following table (anything below 9 defaults to A)

Hex Value	Port Bit Assign
0xA	A A A A A
0xB	D A A A A
0xC	D D A A A
0xD	D D D A A
0xE	D D D D A
0xF	D D D D D

For example, sending a %AC is equivalent of "set PORTA inputs to a '111000', where a '0' makes the bit an analog input, and a '1' makes it an digital input. Note that the range for then inputs are limited from the reference ground to the board voltage of 3.3V (typical) on both analog and digital. In technical terms, this is setting the ADCON1 register in the micro controller.

**%B** The micro controller has a 8 bit digital I/O port, called "PortB". Sending this command by itself, will return a value that indicates if the particular input on the port is an input (1) or an output (0), represented by two byte hex value. Sending a hex value with the command will force a setting change for it. The setting is stored in non-volatile memory such that on power cycles, it will retain its settings. Valid range for setting is 00-FF

For example, sending the command "%BC3" is equivalent of saying "set PORTB

I/O equal to a '10110011', where a '0' makes the bit an output, and a '1' makes it an input (this is shown from B7:B0). In technical terms, this is setting the value for TRISB

**%P** Puts a value to PortB. This will allow those pins that are set as output to be turned on and off.

For example "%P12" would be interpreted as a binary string of 00010010 where if a bit was previously set for an output, a value of 1 would place a logic high on the terminal block and a 0 would place a logic low on the terminal block. If a port bit was set for an input, this would not affect that signal.

Please note that the digital range is limited to the board voltage of reference ground to 3.3VDC (typical) with a maximum current of 25mA per pin. It is typically suggested that outputs are current sinking, rather than sourcing. Both reference ground and power are available on the terminal blocks for user interfacing.

**%G** Get the input status for both port A and port B. This will read the A and B ports, and return a string of values for each port on separate lines. For those locations on port B that are outputs, a zero value will return. The format of the string is as follows:

**B=00[00000000]**

If Port A is set for digital, the output would be formatted as

**A=[00000]**

If Port A is set for both digital and analog, the output would be formatted as

**A=[00] 2:A9C0|1:6080|0:FFC0|**

If Port A is set for all analog, the output would be formatted as

**A=4:FF80|3:B000|2:5BC0|1:3740|0:FFC0|**

Note that Port A analog values are always returned in a two bytes since the A/D is 10 bits in value. The actual value is represented by [9/8/7/6/5/4/3/2] [1/0/-/-/-/-/-/-]

**%C** XBee commands. This may be used to query the XBee module using the XBee AT command set, or set a new value in the device. Setting a new value is always followed by a write command to the module to store the new value in non-volatile memory.

For example, sending a %CATID (return) would act as a query only, and return the PAN ID value. However sending a %CATID5 (return) would set the PAN ID to a value of 5 and store it to memory.

**%L** This command controls the LED on the module to be turned on, off, beacon once per minute or indicate the module is being accessed.

%L0 turns off the LED

%L1 turns on the LED where it remains on till another command affects it

%L2 beacons the LED so once per minute it will turn on for approximately one second

%L3 turns on the LED if the module is be addressed and is activated via the serial link.

**%Q** There are two query modes of operation, either a periodic reading of the ports, that occurs based on the H:M:S timer settings or wake on request (polling). When Q is set to 1, polling is enabled. This will put the processor to sleep, and wakes when a serial command is received. The Xbee module is not put to sleep in this mode. The processor will go back to sleep based on the H:M:S timer settings. When Q is set to 0, the processor will wait for the time period set in the H:M:S timer settings, take a reading of the ports, transmit and reset the timer. The Xbee module will not be put asleep in this mode. Using the command mode to place it asleep can be enabled, and the wake is asserted by the processor when it is ready to transmit.

**%V** Reports the firmware version number.

**%H** Hour timer value. The processor includes a timer that is represented in hours:minutes:seconds and is used for either a timer for sending data on a periodic basis or how long the processor stays awake after receiving a serial command, depending on its mode (see command Q). The value is stored in non-volatile memory, and is has a range of 00 – 59, and always must be two digits. For example, 5 hours is represented by '05'. Default value is 0

**%M** Minutes timer value. The processor includes a timer that is represented in hours:minutes:seconds and is used for either a timer for sending data on a periodic basis or how long the processor stays awake after receiving a serial command, depending on its mode (see command Q). The value is stored in non-volatile memory, and is has a range of 00 – 59, and always must be two digits. For example, 5 hours is represented by '05'. Default value is 1

**%S** Seconds timer value. The processor includes a timer that is represented in hours:minutes:seconds and is used for either a timer for sending data on a periodic basis or how long the processor stays awake after receiving a serial command, depending on its mode (see command Q). The value is stored in non-volatile memory, and is has a range of 00 – 59, and always must be two digits. For example, 5 hours is represented by '05'. Default value is 0.

