# PIC-BEE2 Installation and Operation Instructions

**Version 1.1**

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 PICBEE2 is a second generation device that incorporates many new features that were not available on the previous.

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

The PICBEE2 is designed specifically to mount in a plastic wireless enclosure that includes a 2 cell (AA) battery pack. This allows for simple and easy installation for almost any application. In addition, it can also support a 2 cell (AA) battery pack that can be mounted on the back side of the board, and the assembly can then be mounted to a panel or enclosure using the four mounting holes as shown below.

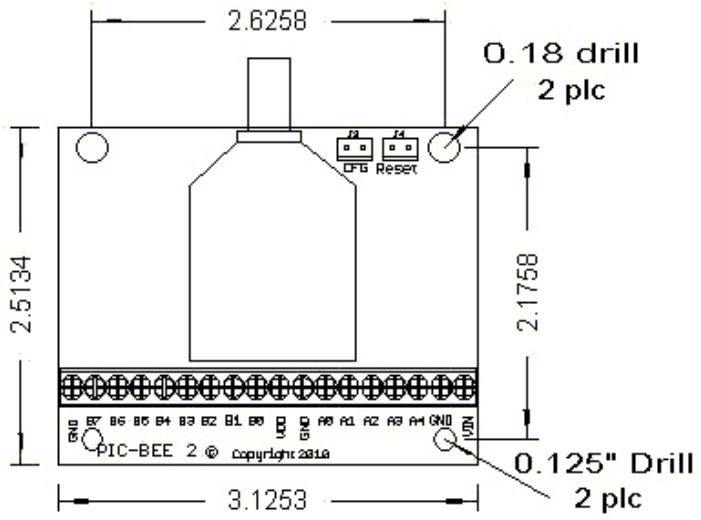


Figure 1 PICBEE2 Mechanical Drawing

An onboard boost regulator accepts the battery voltage on the terminal block labeled VIN and converts it to the required 3.3VDC for the XBee, processor and supporting circuitry. The very nature of the boost circuit allows for longer operation from a set of batteries as it will keep the voltage at the operating level until the cells can no longer provide the current necessary. The length of operation of the battery is dependent on battery type and operating mode.

In addition, power can be supplied directly to the module on the terminal blocks labeled VDD and GND. Be sure that the voltages being supplied are in the range of 3.0VDC and 3.3VDC. Any voltage outside this range cannot guarantee correct operation and may lead to damage of components.

The module is designed of course to allow additional IO capabilities beyond that of the standard XBEE module. Specifically, it provides 8 digital inputs or outputs on terminals labeled B0 thru B7. The levels expected are 3.3VDC, or referenced to the VDD terminal and the GND terminal. For those signals that are inputs, there is an internal weak pullup provided internally to the processor. What this means, is for a unconnected signal it will read as a logic 1 or high. If the input is being used to monitor a switch closure (mechanical or solid state) the contact must be closed referenced to the module GND terminal. This provides a logic 0 or low when the input is active. If the terminal is selected to be an output, it is suggested that the controlled device "sink" the current to the processor, meaning that it generates a logic 0 to activate the device. Sold state devices are much more suited to take the current into the part than try and generate it.

There are five other inputs, that can be configured for analog or digital that are labeled A0 thru A4. If these are configured for digital and external pullup or pulldown must be used in order to provide the necessary logic levels, as these are termed floating inputs, that can be undetermined at any time. The reference to the pullup/pulldown must be made relative to the board VDD and GND terminals. For analog inputs, the range is VDD to GND, and also requires the input be referenced to the board ground well.

**Hardware Configuration**

There are two jumper blocks on the assembly and are only monitored when the board is powered up.

RESET - 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. The LED will turn on during this process. Once it goes off, remove the jumper and power cycle the unit.

CFG - 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. The LED will turn on during this process. Once it goes off, remove the jumper and power cycle the unit.

**Optional Connectors**

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

- RS232/TTL serial (3.3VDC power,ground,serial in, serial out) thru a 4 pin 2.54mm header *note : serial interface is dependent on board model type.
- ICSP – in circuit serial programming using the Microchip PICKIT2 or ICD2/3 2.54mm 6 pin header
- I2C/SPI – serial communications, power and ground on a 2.54mm header

**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.

```
PICBEE2
[MY=0] [ID=3332] [DH=0] [DL=0]
System Configuration
Firmware Version : 1.2
Port A:1F
Port B:FF
LED Off
Polling
Periodic Time (H:M:S) 1:59:59
Current  Time (H:M:S) 1:59:59
>
```

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**: Sets up for the 8 bit digital port to be input or output
**C**: Xbee command mode
**E:** low battery reporting enabled
**G**: Gets the values on PortA and PortB, reports in ASCII format
**g:** Gets the values on PortA and PortB, reports in hex format
**H**: Timer module hours setting
**L** : LED settings
**M**: Timer module minutes setting
**P** : Puts a value on portB, for those set to output
**Q** : Query mode – polling (0) or Periodic reports(1)
**S** : Timer module seconds setting
**T** : Temperature in Celsius, decimal
**t** : Temperature in Celsius, hex output
**V** : Reports back module configuration

**%A** The micro controller has a 5 bit port that is always configured to be an input, it cannot be set for outputs. By default, the inputs are all analog with a range of 0VDC to 3.3VDC (the power on the micro itself). Using the "%A" command without any value associated, it will return the current pattern of the inputs, where a "1" indicates its analog, and a '0" for digital. See the section on using PORTA for further information. The pattern returned is in the order of MSB to LSB of the port assignment. On the PICBEE2 terminals it shows A4 to A0. Sending a %A for all inputs set for analog would return the following:

A: 1F this translates to a pattern of '0001 1111' for the port

To set the LSB (bit 0) for a digital input, sending the following command would accomplish this:

%A1E<CR> this translates to a pattern of '0001 1110' for the port

These settings are retained in non-volatile memory so they will restore when power cycling the PICBEE2 board. When sending commands for setting the pattern, the string can be sent in either hex or ASCII but must always be in the following sequence (shown in ASCII)

<%><A><upper nibble><lower nibble><CR>

**%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, with the analog values represented by two bytes, upper and lower. The A/D conversion is 10 bits, with the first byte containing the upper two bits and the second byte is the lower 8 bits.

```
B=FF[11111111]
A=[011F][003F][0004][0003][0003]
```

In this reply, no inputs for port B are active, since there is an internal pullup, and an active input would read zero. The analog results can be converted using the following formula:

(VDD/1024) * decimal value

Typically VDD is 3.3V from the boost circuit, but it can begin to drop when the batteries begin to weaken. Of course if running from a power supply this will always be the power supply value.

Using the first value 011F, this is converted to a decimal of 287. Using the above formula, this is converted to (3.3/1024) * 287 or 0.925 V

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

```
B=FF[11111111]
A=[0][1][0][0][0]
```

This would be interpreted that bit 3 is high or 1, and all others or low or 0

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

```
B=FF[11111111]
A=[011F][1][0][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, the default that keeps the XBEE active all the time, and waits for commands and requests. This is referred to as polling where the host will poll for information, and is placed into this by sending a "%Q0" command. The other mode is wake the XBEE from sleep, read the ports, transmit and then return back to sleep mode, and is put into this mode by sending a "%Q1" command. While the XBEE is asleep, it cannot of course receive commands over the RF Zigbee link. Because of this, the XBEE is kept awake after transmitting for 5 seconds in the event the host wishes to send a new command.

The period of reading the ports is set by the H,M and S settings. Prior to putting the PICBEE2 in this mode, you should set these values to the time period you wish to wake and read the ports.

**%V** Reports the current board configuration

```
System Configuration
Firmware Version : 1.3
A:1F
Port B:FF
LED Off
Polling Mode
Periodic Time (H:M:S) 01:59:59
Current  Time (H:M:S) 01:53:01
```

**%H** Hour timer value. The processor includes a timer that is represented in hours:minutes:seconds and is used as a timer for sending data on a periodic basis. 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 01

**%M** Minutes timer value. The processor includes a timer that is represented in hours:minutes:seconds and is used as a timer for sending data on a periodic basis. 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 01

**%S** Seconds timer value The processor includes a timer that is represented in hours:minutes:seconds and is used as a timer for sending data on a periodic basis. 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 00

**%T** Reports current temperature in Celsius, in decimal `Current Temperature 27C`

**%t** Reports current temperature in Celsius, in hex

**How to properly use the digital and analog ports**

PORTA is a 5 bit input port, and can be configured for either analog or digital using the "%A" command. The analog range is set for 0VDC at the low end, and the board "VDD" at the upper end. The VDD level is established by the onboard boost regulator and is normally 3.3V, where the incoming battery voltage can actually drop below this level. Refer to the board description for additional details on how this may affect the analog readings.

The analog inputs are referenced to the circuit ground, so any analog readings must also be referenced to this. Examples of analog inputs may the thermistors, voltage references and isolated voltage outputs from sensors.

When a particular bit is set to be a digital input, this signal **must** be pulled high by a resistor, 4.7K to 10K to the board VDD reference. These are what are referred to as open collector inputs and without the pull up they will float. Please note that for example when monitoring a normally open switch, the reading will show a "1", and when its closed, with the contact going to the board ground reference, it will then show a "0" for closure.

Please note that as the battery level decreases to a lower level, it reaches a point where the boost regulator can no longer supply the necessary power.

The incoming battery voltage can be easily monitored by connecting the VIN to one of the analog inputs and polling for its level as well as logging the data on transmission.

PORTB is a digital only eight bit port. Any of the port bits can be set to be an input or output using the %B command. For inputs, an internal pull up is enabled, such that no external resistors are required when its used for inputs. The outputs can either sink or source up to 20mA of current, and is suggested as a good design practice to "sink" the current, or drive the output low when enabling and external device that requires more than a few mA of current.