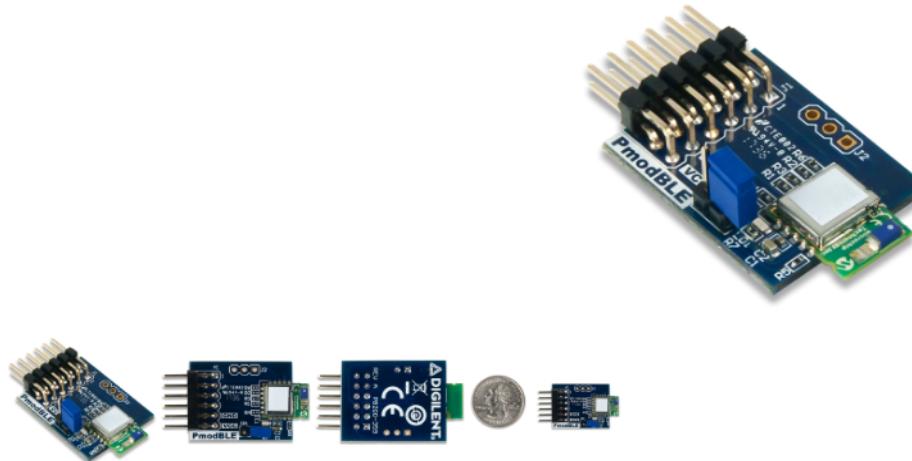


The Digilent Pmod BLE (Revision A) is a powerful peripheral module meant for use with any UART enabled development board. This Pmod employs the Roving Networks RN4870 to create a fully integrated Bluetooth Low Energy interface. This chip features an onboard Bluetooth stack that makes connecting and communicating with any Bluetooth 4.0+ device possible through simple UART commands. The RN4870 also features four built in GATT services: Device Information, Airpatch, BeaconThings, and UART Transparent data streaming. It also allows up to five custom public services and up to 4 custom private services, each allowing up to eight custom characteristics. The PmodBLE is a cheap, versatile, low energy addition to the wireless Pmod library.



## Features

- Fully Qualified Bluetooth® Low Energy Module
- On-Board Bluetooth 4.2 Low Energy Stack
- Certified to FCC, IC, CE, KCC, NCC and SRRC
- [ASCII Command Interface API](#)
- Secure communication AES128 Encryption
- Remote Configuration Over The Air
- Small PCB size for flexible designs 1.0" x 0.8" (2.57 cm x 2.03 cm)
- 12-pin Pmod connector with UART interface
- Follows Digilent Pmod Interface Specification Type 3A

## Specifications

Parameter	Min	Typical	Max	Units
Power Supply Voltage	1.9	3.3	3.6	V
Baud Rate	2400	115200	921000	bps
Data Rate (Transparent UART)			10	kbps
Parameter	Value			
RX Sensitivity	-90			
TX Power	0			
Frequency Band	2.402 - 2.480			

## Pinout Table Diagram

Pin	Signal	Description
1	RTS	Ready to Send

2	RXD	Receive Data
3	TXD	Transmit Data
4	CTS	Clear to Send
5	GND	Power Supply Ground
6	VCC	Power Supply (3.3V/5V)
7	P1_2	Configurable GPIO
8	RST_N	Reset (Active Low)
9	MODE	App/Test Mode pin
10	P1_6/STATUS	Configurable Connection Status
11	GND	Power Supply Ground
12	VCC	Power Supply (3.3V/5V)

## Jumpers

The Pmod BLE has one jumper connected to the MODE pin. When left unshorted, the PmodBLE will start up in standard application mode. When shorted and power cycled, the board will go into test mode, which is used to update the firmware on the RN4870. There is also an unloaded jumper J2 which provides access to 3 configurable pins P1\_3, P1\_7, and P2\_7.

## Physical Dimensions

The pins on the pin header are spaced 100 mil apart. The PCB is 1.011 inches (2.570 cm) long on the sides parallel to the pins on the pin header and 0.8 inches (2.032 cm) long on the sides perpendicular to the pin header. The PCB on the RN4780 module extends the length of the PCB approximately .158 in (4.00 mm) and is .355 in (9.00 mm) wide.

## Functional Description

The Pmod BLE uses a 12-pin Pmod port and communicates via UART on the top row of pins. The bottom row of pins include 2 GPIO pins, a reset pin, and a mode pin that is used to update the firmware of the BLE module.

The typical application for the Pmod BLE is to replace a wired UART connection between two Bluetooth capable devices. When paired with an Android, Linux, Mac OS X, or Windows computer, the Pmod BLE behaves like a serial port similar to how a USB-UART bridge or RS-232 serial port behaves. The Pmod BLE can be easily configured from the Bluetooth connected computer by entering a "Command Mode" that allows settings such as UART baud rate to be programmed into non-volatile configuration registers.

When used with another Pmod BLE, a wireless UART bridge can be achieved with no need for software configuration of the Bluetooth link. This is accomplished by entering a command mode while connected to the PmodBLE through UART. Once in command mode, the device can connect to any BLE address using the "C,0,<address>" command. In this way you can communicate wirelessly between two embedded devices without complicated software like is required with WiFi or 802.15.4 solutions. Communicate wirelessly with simple serial prints and reads!

**WARNING:** Hot-plugging (un-plugging or plugging in the device while powered) the Pmod BLE can possibly wipe the internal memory on the device. Fixing this requires a UART terminal connection between a PC and the Pmod BLE.

## Serial Communication

The Pmod BLE communicates with the host board via the [UART protocol](#). By default, the UART interface uses a baud rate of 115.2 kbps, 8 data bits, no parity, and a single stop bit. The startup baud rate may be customized to predefined rates or set to a specific user customized baud rate ranging from 2400 bps to 921 kbps.

By default, the PmodBLE is configured as a transparent UART device. UART data sent to the PmodBLE will be sent directly to the device it is connected to and vice versa.

Command mode can be entered by sending the module the characters "\$\$\$" to which it will respond "CMD>". When in command mode, the module will respond to a large number of commands allowing the user to customizing the module for specific applications. In order to exit command mode, send "—<cr>" (three minus signs in a row followed by the carriage return character) to which the device will respond "END". Remote configuration, or configuration over a Bluetooth connection, is possible using the command "!,!". It is important to note that any of the "set" commands available for the Pmod BLE must be followed by a power cycle to take effect in any design.

## GPIO

The reset pin (RST) on J1 is active low. If the RST pin is pulled low, the device will undergo a hard reset until the RST pin is pulled high. This hard reset performs similarly to a power cycling of the device.

The P1\_6/STATUS pin directly reflects the connection status of the device. STATUS is driven low by the device when connected and is driven high otherwise. P1\_6 can be reconfigured to several different functions using the SW command (see RN4970 User Guide).

The P1\_2 GPIO pin can be written to by entering command mode and sending the command "|0,08,0X\r\n" (replace X with 8 to set the pin high, or 0 to set the pin low). It can also be read from using the command "|I,08". P1\_2 can also be reconfigured to several different functions using the SW command.

## Quick Start

Here is an example of using the PmodBLE to echo data via bluetooth.

1. Disconnect jumper JP1
2. Power on PmodBLE
3. Wait for data to be received, when data is received send it back

```
while(1){  
    if(!UART_Buffer_Empty){  
        ReadUARTData(readBuffer);//Read UART data into a buffer  
        SendUARTData(readBuffer);//Send that same buffer back  
    }  
}
```

4. Connect to the PmodBLE using a BLE terminal app on your phone
5. Select a characteristic to read and write data in the BLE terminal
6. Type something into the terminal and send it
7. The PmodBLE will echo this data back