Rigado LLC
3950 Fairview Industrial Dr. SE Suite 100
Salem, Oregon 97302
866-6-RIGADO ◆ modules@rigado.com
www.rigado.com/modules

BMDware

Firmware, Done.

1. Introduction

The BMDware firmware package pre-programmed to all Rigado BMD-200 and BMD-300 Series modules provides built-in configurable beaconing and a pass-through UART connection. Beacon products only need some configuration via the Bluetooth Low Energy interface. No firmware development or programming is necessary. Wireless connectivity is easily added to existing systems with external connectors using the pass-through UART connection. With the built-in AT style command interface, a BMD 200 or 300 series module is easily integrated into an existing design allowing for control of BMDware features through a connected UART.

1.1 Feature List

- Fully Configurable Beacon Packet structure
- Support for Apple iBeacon built-in; custom beacon packets also available
- Configurable Transmit power for connectable and beacon advertisements
- Pass-through UART up to 80000 baud without flow control
- Configurable Parity and Flow Control
- AT Command interface via connected microcontroller
- Passcode protected interface
- Beacon broadcast while still connectable
- Configurable advertising name and interval
- Configurable GPIO
- Configurable BLE connected status pin
- Password protected

TABLE OF CONTENTS

2. Definitions

- **UUID**: Universally Unique Identifier
- **Beacon:** In the context of this document, a beacon refers to a Bluetooth Low Energy based advertisement packet which contains device identifiable information using a non-connectable advertisement.
- **iBeacon:** An iBeacon is a specific implementation of a Beacon using a specification provided by Apple, Inc. To obtain a copy of the iBeacon specification and/or use the iBeacon logo for a product, the product design must join the Apple MFi program. See https://developer.apple.com/programs/mfi/ for more details.
- **UriBeacon:** URI Beacon is a specification provided by Google. More details can be found here: https://github.com/google/uribeacon
- **AltBeacon:** AltBeacon is another beacon specification. More details here: http://altbeacon.org
- **UART:** Universal Asynchronous Receiver Transmitter

3. BMDware Pin Descriptions

3.1 BMD-200

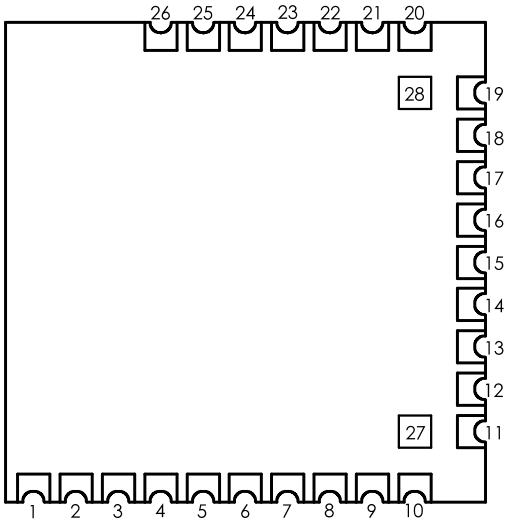

Figure 1- BMD-200 Pin out (Top View)

Pin	Name	Direction	BMD-200 BMDware Pin Functions
11	P0.00	In In/Out	**DTM UART RX** **GPIO pin map 03**, N/C if not used. On BMDware Start-up: **High** = Enter DTM mode; **Low** = Enter Normal Operation *Internal 12kΩ pull-down during BMDware start-up, then Hi-Z*
12	P0.01	Out In/Out	**DTM UART TX** – Active if DTM UART is enabled by P0.00 **GPIO pin map 00**
15	P0.04	In/Out	**GPIO pin map 04**
16	P0.05	In	**Beacon-Only mode selection**, N/C if not used. On BMDware Start-up: **High** = Bridge UART enabled; **Low** = Bridge UART disabled *Internal 12kΩ pull-up during BMDware start-up, then Hi-Z*
17	P0.06	In In/Out	**UART AT Command mode selection** **GPIO pin map 05**, N/C if not used. On BMDware Start-up: **High** = Full pass-through mode; **Low** = AT command mode *Internal 12kΩ pull-up during BMDware start-up, then Hi-Z*
20	P0.08	In In/Out	**Bridge UART CTS** **GPIO pin map 06** Disabled in Beacon-Only & DTM modes, N/C if not used.
21	P0.09	In	**Bridge UART RX** Disabled in Beacon-Only & DTM modes, N/C if not used.
22	P0.10	Out	**Bridge UART TX** Disabled in Beacon-Only & DTM modes, N/C if not used.
23	P0.11	Out	**Bridge UART RTS** Disabled in Beacon-Only & DTM modes, N/C if not used.

Pin	Name	Direction	BMD-200 BMDware Pin Functions
24	SWDIO	In/Out	**Debug I/O / RESET**
25	SWDCLK	In	**Debug Clock**
5	P0.24	In/Out	**GPIO pin map 01**, N/C if not used.
6	P0.25	In/Out	**GPIO pin map 02**, N/C if not used.
8, 9, 13, 14	RSV	N/A	**Reserved N/C**
18	V_{CC}	Power	**+1.8V to +3.6V DC** 1µF - 4.7µF ceramic capacitor is recommended between V_{CC} and GND
1, 2, 3, 4, 7, 10, 19, 26, (27, 28 opt.)	GND	Power	**Electrical Ground**

Table 1 – BMD-200 BMDware Pin Functions

Nordic Pin Ref	BMD-200 Pin	BMD-200 BMDware Function	BMD-200 BMDware GPIO Map	BMD-200 Eval Kit Pin Ref	BMD-200 Eval Kit Pin Name
P0.00	11	DTM UART RX/GPIO	03	P0.00	SW2
P0.01	12	DTM UART TX/GPIO	00	P0.01	LED_Red
P0.02	13	-	-	P0.02	Bridge UART CTS
P0.03	14	-	-	P0.03	Bridge UART RTS
P0.04	15	GPIO	04	P0.04	I2C INT
P0.05	16	Beacon Only Mode	-	P0.05	I2C SCL
P0.06	17	AT Only Mode/GPIO	05	P0.06	Ambient/AREF
P0.08	20	Bridge UART CTS/GPIO	06	P0.08	I2C SDA
P0.09	21	Bridge UART RX	-	P0.09	Bridge UART RX
P0.10	22	Bridge UART TX	-	P0.10	Bridge UART TX
P0.11	23	Bridge UART RTS	-	P0.11	AT Only / SW1
P0.24	5	GPIO	01	P0.24	LED_Green
P0.25	6	GPIO	02	P0.25	LED_Blue
P0.26	8	-	-	P0.26	Analog In
P0.27	9	-	-	P0.27	Analog In

Table 2 - BMDware assignments for BMD-200 module and BMD-200 Evaluation Kit

3.2 BMD-300

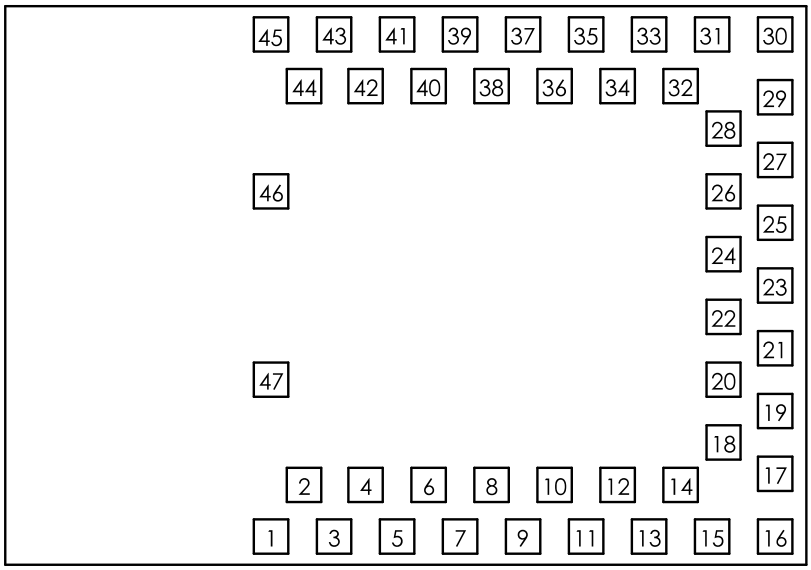

Figure 2 – BMD-300 Pin out (Top View)

Pin	Name	Direction	BMD-300 BMDware Functions
13	P0.00	In/Out	**GPIO pin map 0x00**, N/C if not used.
14	P0.01	In/Out	**GPIO pin map 0x01**, N/C if not used.
15	P0.02	In/Out	**GPIO pin map 0x02**, N/C if not used.
19	P0.03	In/Out	**GPIO pin map 0x03**, N/C if not used.
20	P0.04	In/Out	**GPIO pin map 0x04**, N/C if not used.
21	P0.05	-	Unused, N/C
22	P0.06	Out	Bridge UART TX Disabled in Beacon-Only & DTM modes, N/C if not used.
23	P0.07	In	Bridge UART CTS Disabled in Beacon-Only & DTM modes, N/C if not used.
24	P0.08	In	Bridge UART RX Disabled in Beacon-Only & DTM modes, N/C if not used.
25	P0.09	In/Out	**GPIO pin map 0x05**, N/C if not used.
26	P0.10	In/Out	**GPIO pin map 0x06**, N/C if not used.
27	P0.11	Out	DTM UART TX Only enabled in DTM mode; N/C if not used.
28	P0.12	In	DTM UART RX / DTM Mode Only enabled in DTM mode; N/C if not used. On BMDware Start-up: High = Enter DTM mode; Low = Enter Normal Operation Internal 12kΩ pull-down during BMDware start-up, then Hi-Z
31	P0.13	In	Beacon Only Mode On BMDware Start-up: High = Bridge UART enabled; Low = Bridge UART disabled Internal 12kΩ pull-up during BMDware start-up, then Hi-Z

Pin	Name	Direction	BMD-300 BMDware Functions
32	P0.14	In	UART AT Command Mode On BMDware Start-up: High = Full pass-through mode; Low = AT command mode Internal 12kΩ pull-up during BMDware start-up, then Hi-Z
33	P0.15	In/Out	**GPIO pin map 0x07**, N/C if not used.
34	P0.16	In/Out	**GPIO pin map 0x08**, N/C if not used.
35	P0.17	In/Out	**GPIO pin map 0x09**, N/C if not used.
36	P0.18	In/Out	**GPIO pin map 0x0A**, N/C if not used.
37	P0.19	In/Out	**GPIO pin map 0x0B**, N/C if not used.
38	P0.20	In/Out	**GPIO pin map 0x0C**, N/C if not used.
39	P0.21	In/Out	BMDware Reset Pin
40	P0.22	In/Out	**GPIO pin map 0x0D**, N/C if not used.
41	P0.23	In/Out	**GPIO pin map 0x0E**, N/C if not used.
42	P0.24	In/Out	**GPIO pin map 0x0F**, N/C if not used.
6	P0.25	In/Out	**GPIO pin map 0x10**, N/C if not used.
7	P0.26	In/Out	**GPIO pin map 0x11**, N/C if not used.
8	P0.27	In/Out	**GPIO pin map 0x12**, N/C if not used.
9	P0.28	In/Out	**GPIO pin map 0x13**, N/C if not used.
10	P0.29	In/Out	**GPIO pin map 0x14**, N/C if not used.
11	P0.30	In/Out	**GPIO pin map 0x15**, N/C if not used.
12	P0.31	In/Out	**GPIO pin map 0x16**, N/C if not used.

Table 3 – BMD-300 BMDware Pin Functions

BMD-300 Pin	Name	Pin Map Index
13	P0.00	0x00
14	P0.01	0x01
15	P0.02	0x02
19	P0.03	0x03
20	P0.04	0x04
25	P0.09	0x05
26	P0.10	0x06
33	P0.15	0x07
34	P0.16	0x08
35	P0.17	0x09
36	P0.18	0x0A
37	P0.19	0x0B
38	P0.20	0x0C
20	P0.22	0x0D
41	P0.23	0x0E
42	P0.24	0x0F
6	P0.25	0x10
7	P0.26	0x11
8	P0.27	0x12
9	P0.28	0x13
10	P0.29	0x14
11	P0.30	0x15
12	P0.31	0x16

Table 4 – BMD-300 BMDware GPIO Pin Mappings

4. Bluetooth Low Energy Architecture

The Bluetooth low energy interface of BMDware has two main services: Beacon Configuration and UART Configuration. Each interface is configured independently over the other. However, some control commands are sent only through the Beacon Configuration service (such as setting a password for the device).

Note: *All Data for fixed length characteristics (e.g. not the control point) are little endian hexadecimal. For example, a baud rate of 57600, which has a hexadecimal value of 0xE100 is represented as 0x00E1. A baud rate of 115200, which has a hexadecimal value of 0x1C200 is represented as 0x00C20100.*

4.1 Beacon Configuration Service

The Beacon Configuration Service provides all characteristics necessary for configuration of a beacon broadcast from the device. The configurable components are: UUID, Major, Minor, Advertisement Interval, Beacon TX Power, Connectable TX Power, and Enable. In addition, it is possible, through the Control Point of this service, to deploy a fully customized beacon packet for beacons such as Eddystone and AltBeacon.

4.2 Custom Beacon Data

In BMDware, it is possible to broadcast as either an iBeacon style beacon or as a Custom style beacon. The custom beacon provides complete control over the full advertisement packet manufacturer specific data section. It does not support handling scan requests. The two beacon styles are mutually exclusive. BMDware does not support broadcasting both types at the same time.

4.2.1 Service

UUID: 2413B33F-707F-90BD-2045-2AB8807571B7

The Beacon configuration service provides necessary beacon configuration information and a control point for performing additional actions such as password protection and setting calibration data. This section lists the various characteristics along with their UUIDs and data types.

4.2.2 Control Point (0 to 20 bytes)

UUID: 2413B43F-707F-90BD-2045-2AB8807571B7

The control point provides an interface for additional commands. See the Control Point Commands section for a detailed description of all available commands.

4.2.3 Beacon UUID (128 bits)

UUID: 2413B53F-707F-90BD-2045-2AB8807571B7

The Beacon UUID characteristic sets the UUID broadcast within the beacon advertisement packet for an iBeacon style beacon.

4.2.4 Major Number (16 bits)

UUID: 2413B63F-707F-90BD-2045-2AB8807571B7

The major number characteristic sets the major number within the beacon advertisement packet for an iBeacon style beacon. This characteristic is 16 bits. Valid values are 0 to 0xFFFF.

4.2.5 Minor Number (16 bits)

UUID: 2413B73F-707F-90BD-2045-2AB8807571B7

The minor number characteristic sets the minor number within the beacon advertisement packet for an iBeacon style beacon. This characteristic is 16 bits. Valid values are 0 to 0xFFFF.

4.2.6 Beacon Advertising Interval (16 bits)

UUID: 2413B83F-707F-90BD-2045-2AB8807571B7

This characteristic sets the advertising interval for the Beacon advertisement packet. This characteristic is 16 bits. Values are represented in milliseconds and range from 50 to 4000 ms. Higher advertising intervals will reduce battery life.

4.2.7 Beacon TX Power (8 bits)

UUID: 2413B93F-707F-90BD-2045-2AB8807571B7

The TX power characteristic provides a way to set the output radio power for the beacon advertisement packet. The default value is -4. Available values are:

- -30, -20, -16, -12, -8, -4, 0, 4 in dBm

4.2.8 Beacon Enable (8 bits)

UUID: 2413BA3F-707F-90BD-2045-2AB8807571B7

The beacon enable characteristic enables and disables the beacon advertisement. If set to 0, no beacon advertisement will be generated. Any other value will cause the device to generate beacon advertisements. The default value is 0.

4.2.9 Connectable TX Power (8 bits)

UUID: 2413BB3F-707F-90BD-2045-2AB8807571B7

The connectable TX power sets the radio output power for connectable advertisement packets. These packets are sent out every so often providing a way for configuration apps to connect and update settings. The default value is -4. Valid values are:

- -30, -20, -16, -12, -8, -4, 0, 4 in dBm

4.3 UART over Bluetooth Smart Service

The UART over Bluetooth Smart service provides all necessary characteristics for configuration of the Bluetooth Smart pass-through UART as well as characteristics for transmission and reception of UART data. The configurable components are Baud Rate, Flow Control, Parity, and Enable.

Note: *The Pass-through UART settings do not affect AT Mode settings. AT Mode UART settings are fixed. See the AT Mode section for further details.*

4.3.1 Service

UUID: 6E400001-B5A3-F393-E0A9-E50E24DCCA9E

The UART over Bluetooth Smart service provides a configurable pass-through UART. Data written to the TX characteristic will be sent from the hardware UART to a connected device. Data received on the hardware UART from a wired connection will be sent back to the host connection via a characteristic notification. This service makes no attempt to ensure data was received by either side of the connection.

4.3.2 TX Characteristic (1 to 20 bytes)

UUID: 6E400002-B5A3-F393-E0A9-E50E24DCCA9E

This characteristic is used to send data to the physical UART.

4.3.3 RX Characteristic (1 to 20 bytes)

UUID: 6E400003-B5A3-F393-E0A9-E50E24DCCA9E

This characteristic receives data from the physical UART via notifications.

4.3.4 BAUD Rate Characteristic (32 bits)

UUID: 6E400004-B5A3-F393-E0A9-E50E24DCCA9E

The baud rate characteristic configures the BAUD rate of the physical UART connection. It does not have any bearing on over the air transfer rate. Default value is 57600. This value is 32 bits. Available values are:

- 1200
- 2400
- 4800
- 9600
- 14400
- 19200
- 28800
- 38400
- 57600
- 76800
- 115200 (flow control required)

- 230400 (flow control required)

Note: *The BMD modules support higher baud rates, but due to MTU size limitation, the effective throughput is reduced.*

Note: *The UART Baud Rate does not directly correlate to actual throughput. With an average 30 mS connection interval, the average throughput is expected to be about 4800 bps. So, while the hardware UART can communicate much faster, high throughput is not directly achievable on the module.*

4.3.5 Parity Characteristic (8 bits)

UUID: 6E400005-B5A3-F393-E0A9-E50E24DCCA9E

The parity characteristic enables and disables odd parity for physical UART transmissions. It does not have any bearing on over the air data transfers. The default is off or 0. Any other value will enable parity.

4.3.6 Flow Control Characteristic (8 bits)

UUID: 6E400006-B5A3-F393-E0A9-E50E24DCCA9E

The flow control characteristic enables and disables hardware flow control for the physical UART transmissions. It does not have any bearing on over the air data transfers. The default state is off or 0. Any other value will enable hardware flow control.

4.3.7 UART Enable Characteristic (8 bits)

UUID: 6E400008-B5A3-F393-E0A9-E50E24DCCA9E

Much like the enable characteristic for the beacon, this characteristic enables and disables the physical UART connection to the over the air connection. The default is off or 0. Any other value will enable the over the air UART.

4.4 Device Information Service (0x180A)

The device information service is also available. This service provides information as specified by the Bluetooth SIG. The available fields are Manufacturer Name (0x2A29), Serial Number (0x2A25), Firmware Version (0x2A26), and Model Number (0x2A24).

For more information on the Device Information Service, refer to https://developer.bluetooth.org/gatt/services/Pages/ServiceViewer.aspx?u=org.bluetooth.service.device_information.xml.

5. Control Point Commands

BMDware has additional commands that can be sent to the control point. This section details those commands.

5.1 Command Architecture

All commands, except for the start bootloader command, are one byte long with a variable amount of parameter data. When notifications are enabled for the control point, a response is sent for every command. The response is either one byte long or as long as noted by the response data from the command.

5.1.1 Control Point UUID

2413B43F-707F-90BD-2045-2AB8807571B7

5.2 Custom Beacon Data Commands

5.2.1 Set Custom Beacon Data 1: 0x20

Protocol Version: 1
Description: This command is sent to set the first half of the custom beacon packet data. In total, the custom beacon packet is up to 31 bytes long. Up to the first 19 bytes are sent via this command.
Parameters: 1 to 19 bytes of beacon data

5.2.2 Set Custom Beacon Data 2: 0x21

Protocol Version: 1
Description: This command is sent to set the second half of the custom beacon packet data. In total, the custom beacon packet is 31 bytes long. Only the last 1 to 12 bytes are sent via this command. If at least 19 bytes have not first been sent via the Set Custom Beacon Data 1 command, invoking this command will result in Invalid Data error response.
Parameters: 1 to 11 bytes of beacon data

5.2.3 Save Custom Beacon Data: 0x22

Protocol Version: 1
Description: This command is sent to commit the custom beacon data to non-volatile storage. If no custom beacon data has been sent, invoking this command will result in an Invalid Data error response.
Parameters: None

5.2.4 Clear Custom Beacon Data: 0x23

Protocol Version: 1
Description: This command is sent to clear all custom beacon data. After sending this command, beaconing will revert back to iBeacon style beaconing.
Parameters: None

5.2.5 Set RSSI Calibration Data: 0x40 <TX Power> <RSSI>

Protocol Version: 1

Description: This command is sent to set the RSSI calibration data. The RSSI calibration is sent with iBeacon packets. This value is the RSSI value expected when measured at 1 meter from the device. Using this value, the iBeacon infrastructure on iOS is able to determine the relative distance from the device and then reconcile the information to produce Immediate, Near, and Far ratings for consumption by iOS applications interfacing with the iBeacon.

Parameters:

TX Power: The power at which the RSSI was measured

RSSI: The RSSI value measured at a 1-meter distance from the device, line of sight

5.2.6 Get RSSI Calibration Data: 0x41

Description: This command is sent to retrieve the RSSI calibration data.

Return format: <0x41><TX Power><RSSI>

5.3 GPIO Commands

5.3.1 Configure GPIO: 0x50 <Pin> <Direction> <Pull>

Protocol Version: 1 (3.0.0)

Description: This command provides configuration of free GPIO pins. Pins are configurable as input or output and with or without a pull up or pull down for input pins.

Note: *GPIO configurations are not saved across power cycles.*

Parameters:

Pin: Pin mapping:

(BMD-200, Table 1) or (BMD-300 Series, Table 4)

Direction: Pin direction:

00 = input | 01 = output

Pull: Internal pin pull-up or pull-down, 13KΩ (nominal):

00 = no pull | 01 = pull-down | 03 = pull-up

5.3.2 Write GPIO: 0x51 <Pin> <State>

Protocol Version: 1 (3.0.0)

Description: The write GPIO command changes the state of a GPIO configured as an output.

Parameters:

Pin: Pin mapping:

(BMD-200, Table 1), (BMD-300 Series, Table 4)

State: Pin state:

00 = Low / Clear (GND)| 01 = High / Set (V_{CC})

5.3.3 Read GPIO: 0x52 <Pin>

Protocol Version: 1 (3.0.0)

Description: The read GPIO command reads the current value of an input or the state of the output buffer and returns this value via a notification.

Parameters:

Pin: Pin mapping:

(BMD-200, Table 1), (BMD-300 Series, Table 4)

Return Format: <0x52> <Pin> <State>

Pin: Pin mapping:

(BMD-200, Table 1), (BMD-300 Series, Table 4)

State: Pin state:

00 = Low / Clear (GND)| 01 = High / Set (V_{CC})

5.3.4 Get GPIO Configuration: 0x53 <Pin>

Protocol Version: 1 (3.0.0)

Description: The get GPIO configuration command provides the current configuration for the supplied pin via a notification.

Parameters:

Pin: Pin mapping:

(BMD-200, Table 1), (BMD-300 Series, Table 4)

Return Format: <0x53> <Pin> <Direction> <Pull>

Pin: Pin mapping:

(BMD-200, Table 1), (BMD-300 Series, Table 4)

Direction: Pin direction:

00 = input | 01 = output

Pull: Internal pin pull-up or pull-down, 13KΩ (nominal):

00 = no pull | 01 = pull-down | 03 = pull-up

5.3.5 Configure Status Pin: <0x54> <Pin> <Polarity>

Protocol Version: 1 (3.0.0)

Description: The status pin provides a BLE connection indication. When a device is connected to BMDware via BLE, the status pin is active, otherwise it is inactive. The active state is defined by the polarity. This pin is configured to be a high drive output. If configured for 'Active High', the pin is driven to VCC during an active connection. When configured for 'Active Low', the pin is driven to Ground during an active connection. Otherwise, the pin will be driven to the inactive state.

Note: The pin configured as the status pin cannot be used for GPIO operations while configured as a status pin. Pin configuration is stored across power cycles. The default configuration is no status pin.

Parameters:

Pin: Pin mapping:

(BMD-200, Table 1), BMD-300 Series (BMDware Value 0x00 - 0x04), See Table 4

Polarity: 00 - Active Low; 01 - Active High

5.3.6 De-configure Status Pin: <0x55>

Protocol Version: 2
Description: This command disables the status pin.

5.3.7 Get Status Pin Configuration: <0x56>

Protocol Version: 2
Description: Retrieves the current status pin configuration.
Response Format: <0x56> <BMDware Value> <Polarity>

5.3.8 Read Status Pin: <0x57>

Protocol Version: 2
Description: Returns the current active or inactive state of the status pin.
Response Format: <0x57> <State>
Parameters:
State: 00 - Inactive; 01 - Active

5.4 Connectable Advertisement Commands

5.4.1 Set Connectable Advertising Interval: 0x42 <Interval>

Protocol Version: 2
Description: This command adjusts the advertising interval of the connectable advertisement. This setting is retained across power cycles and applies anytime the device is an advertising state. The default value is 1285 ms.
Parameters:
Interval: The advertising interval in little endian byte order. This is a 16-bit value and the valid range is 20 to 2500 ms.

5.4.2 Get Connectable Advertising Interval: 0x43

Protocol Version: 2
Description: This command retrieves the current advertising interval of the connectable advertisement.
Response Format: <0x43> <Interval>

5.5 System and Miscellaneous Commands

5.5.1 Get Bootloader Info: <0x60>

Protocol Version: 2
Description: Returns the current bootloader information. This data is ONLY available if supported by the bootloader. The data was added to RigDfu version 3.2.0. Previous bootloader versions do not have this information. If the bootloader information is unavailable, this command will return the 'Invalid Data' error.
Response Format: <0x60><Bootloader Info>; See Appendix A for format.

5.5.2 Set Advertised Name: <0x61> <Device Name>

Protocol Version: 2
Description: This command configures the advertisement name of the device. This data is placed in the Device Name of the advertisement data packet. One to eight characters may be specified. The characters must be ASCII characters. The default name is 'RigCom'.
Parameters:
Device Name: 1 to 8 ASCII characters

5.5.3 Get Advertised Name: <0x62>

Protocol Version: 2
Description: Retrieves the current device name.
Response format: <0x62><Device Name>

5.5.4 Get BMDware Protocol Version: <0x63>

Protocol Version: 2
Description: Returns the current protocol version of BMDware.
Response Format: <0x63> <Version>

5.5.5 Unlock: 0xF8 <Password Bytes 0 to 18>

Protocol Version: 1
Description: This command is sent to unlock the device.
Parameters:
Password: The unlock password of up to 19 bytes. See Set Password for further information.

5.5.6 Set Password: 0x31 <Password Bytes 0 to 18>

Protocol Version: 1
Description: This command is sent to set an unlock password for the device. Once the password is set, all configuration settings are locked unless the Unlock command with the correct password is first sent to the device.

The password may be comprised of any ASCII printable character, except a space, up to 19 characters in length. Passwords containing non-printable characters (such as control codes) will not be accepted.

Passwords that are not accepted will received a response of Update Pin Failed. Once unlocked, the device will remain unlocked until a disconnection or reset occurs.
Parameters:
Password: The unlock password of up to 19 bytes.

5.5.7 Start Bootloader: 03563057 (hex bytes)

Protocol Version: 1
Description: Sending this command to an unlocked device will cause the device to reset into the Bootloader. If a firmware update is necessary, this command is used to start the bootloader to perform the firmware update. If no firmware update is started, the device will reset after 2 seconds and resume running the application.

5.5.8 Run Bootloader: 96dff40b (hex bytes)

Protocol Version: 2
Description: This command behaves exactly the same as the 'Start Bootloader' command but causes the bootloader to run for 3 minutes instead of 2 seconds. This command is only available when at least bootloader version 3.2.0 is installed (API version 3). To check the bootloader version, issue the 'Get Bootloader Info' command and verify the verify number.

5.5.9 System Reset: 4b102f37 (hex bytes)

Protocol Version: 2
Description: This command resets BMDware and bypasses the bootloader reducing the startup time of BMDware. Use it when only BMDware needs to be reset and the bootloader is not needed.

5.6 Command Response Codes

Command Success – 0x00
Description: This response is sent anytime a successful command has been received and executed.

Device Locked – 0x01
Description: This response is sent anytime a command is sent to the device when the configuration interface is locked with a password. The unlock command is the only accepted command in this state.

Command Invalid Length – 0x02
Description: This response is sent anytime a command is sent with the wrong parameter data length.

Unlock Failed – 0x03
Description: This response is when the unlock command is executed with an incorrect password.

Update Pin Failed – 0x04 (only for set password command)
Description: This response is sent if the new password contains non-printable ASCII characters or spaces.

Invalid Data – 0x05
Description: This response is sent when any command parameter has invalid data.

Invalid State – 0x06
Description: This response is sent when any command is received when the device is in an invalid state. For example, changing the UART settings while it is enabled is an invalid state.

Invalid Parameter – 0x07
Description: This response is sent when a command parameter is out of range of the available values.

Invalid Command – 0x08
Description: This response is sent when a command to the control point is not valid.

6. AT Command Interface

The AT command interface allows a device or microcontroller connected via a UART interface to configure BMDware through a physical connection rather than over the Bluetooth Low Energy interface. While the Bluetooth Low Energy configuration interface is always available, the AT interface is enabled through a toggle pin on the BMD module. Each command has an associated protocol version. It is the minimum protocol version required to support the command and the version in which the command was added.

6.1 Enabling AT Mode

To enable AT command processing mode, Pin 17 (P0.06) on the BMD-200, Pin 32 (P0.14) on the BMD-300, and Pin 23 (P0.11) on the BMD-200 Evaluation board must be pulled or driven low at boot. A connected microcontroller can set this pin low and then reset the module via the reset pin. If AT Mode needs to be always enabled, the pin can be tied directly to ground.

Note: *When AT Mode is enabled, the Pass-through UART is not available for use.*

6.1.1 AT Mode UART Settings

The settings for the AT Mode UART are fixed. They are completely independent of the pass-through UART settings.

- Baud rate: 57600
- Data bits: 8
- Stop bits: 1
- Parity: None
- Flow Control: None

6.2 Locked Devices

Devices with a password will have to be unlocked before commands can be sent to the device. Due to the nature of UART connections, the device will be relocked after each successful command is sent to the device. In this way, if the UART connection is severed for any reason, the device will not be left in an unlocked state except for one command.

6.3 AT Command Format

Unless otherwise noted, all parameter data for AT mode commands are supplied as hexadecimal big-endian ASCII values. See individual commands for further details.

6.4 AT Command Structure

All AT commands have the following structure:

6.4.1 Write Commands

at$<cmd> <param0> <param1> … <paramN><\n or \r>

Note: *All parameters are separated with a space.*

Note: *Send only \n or \r. If both are sent, the next command will produce an error.*

6.4.2 Read Commands

at$<cmd>?<\n or \r>
The read response will contain the data as defined below along with a `\n` only.
Note: *Some of the GPIO commands are in exception to this. Since they require parameters for the read operation, the '?' postfix is not used.*

6.4.3 Interface Test Command: at

Protocol Version: 1
Description: The 'at' command is a simple command to test that the interface is available. The response is always OK when AT Command mode is enabled.

6.5 Beacon Commands

6.5.1 Beacon UUID: at$buuid, at$buuid?

Protocol Version: 1
Invocation: at$buuid <16-byte uuid>
Description: The UUID command sets the UUID for the iBeacon style beacon data.
Example: Configure UUID as 00112233-4455-6677-8899-aabbccddeeff
at$buuid 00112233445566778899aabbccddeeff
Note: *at$uuid and at$uuid? are deprecated but retained for backward compatibility*

6.5.2 Beacon Major Number: at$bmjid, at$bmjid?

Protocol Version: 2
Invocation: at$bmjid <16-bit major number>
Description: The major number command sets the major number for the iBeacon style beacon data.
Example: Configure Major ID as 0x960a (2710 decimal)
at$bmjid 0a96
Note: *at$mjid and at$mjid? are deprecated but retained for backward compatibility*

6.5.3 Beacon Minor Number: at$bmnid, at$bmnid?

Protocol Version: 2
Invocation: at$bmnid <16-bit minor number>
Description: The minor number command sets the minor number for the iBeacon style beacon data.
Example: Configure Minor ID as bf17 (6079 decimal)
at$bmnid 17bf
Note: *at$mnid and at$mnid? are deprecated but retained for backward compatibility*

6.5.4 Beacon Advertising Interval: at$badint, at$badint?

Protocol Version: 2
Invocation: at$badint <interval>
Description: The advertising interval command sets the broadcast interval for the beacon. See the Bluetooth low energy interface description for available advertising intervals.

Example: Set beacon advertising interval to 100 ms
at$badint 0064
Note: *at$adint and at$adint? are deprecated but retained for backward compatibility*

6.5.5 Beacon TX Power: at$btxpwr, at$btxpwr?

Protocol Version: 1
Invocation: at$btxpwr <tx power>
Description: The Beacon TX power command sets the transmit power for the beacon broadcast data. This TX power is different from the connectable advertisement. See the Bluetooth low energy interface description for available transmit power values. The parameter value is a signed 8-bit integer.
Example: Set beacon TX Power to -4 dBm
at$btxpwr fc

6.5.6 Beacon Enable: atben, atben?

Protocol Version: 1
Invocation: at$ben <0x01> or <0x00>
Description: The beacon enable command is used to enable or disable beacon functionality. Send a value of 0x01 will enable the beacon. Sending any other value will disable the beacon. If the beacon is enabled, the read command will respond with 0x01, otherwise 0x00.
Example: Enable beacon
at$ben 01

6.5.7 Custom Beacon Data Set: at$cusbcn, at$cusbcn?

Protocol Version: 1
Invocation: at$cusbcn <1 to 31 hex bytes>
Description: The custom beacon data command will switch the BMDware beaconing to a fully customized non-connectable advertising data. This allows for alternate beacon specifications such as AltBeacon or URIBeacon. The custom beacon data will have an advertising interval as set by Advertising Interval property.
Example: at$cusbcn 0303FED81A16fed80011223344556677889 9aabbccddeeff0123456789abcd

6.5.8 Custom Beacon Data Clear: at$cbclr

Protocol Version: 1
Invocation: at$cbclr
Description: The custom beacon data clear command will clear the custom beacon data and beacon advertisements will revert to iBeacon style as configured by the UUID, Major, and Minor numbers.

6.5.9 Set Beacon RSSI Calibration: at$bcal, at$bcal?

Protocol Version: 1
Invocation: at$bcal <tx_power><rssi>
Description: The beacon RSSI calibration value sets the calibrated RSSI value for the provided TX power. This field is broadcast as part of an iBeacon packet and is used by iOS to determine how close the device is to a particular beacon. There should be no space between the two parameters.

Example: Set Beacon calibration value to -4 dBm at -68 RSSI
at$bcal fcbc (-4 dBm, -68 RSSI)

6.6 Pass-through UART Commands

These commands do not affect the configuration of the AT Mode UART settings. AT Mode UART settings are fixed. See the AT Mode Introduction section for the AT Mode UART configuration.

6.6.1 Baud Rate: atubr, atubr?

Protocol Version: 1
Invocation: at$ubr <baud rate>
Description: The baud rate command will set the baud rate for the UART pass-through mode only. The baud rate for commands processed during AT mode. The new baud rate takes effect after a reset has been issued. See the Bluetooth Low Energy interface for available Baud Rates and their limitations with and without flow control.
Example: Set Baud Rate to 57600
at$ubr 57600
Example: Get Baud Rate
at$ubr?
Returns 0000E100
Note: *Baud rate values are sent as ASCII decimal (e.g. 57600, 38400, etc.) and read as little-endian hexadecimal (e.g. 0xE100 for the rate of 57600 baud is represented as 0x00E1)*

6.6.2 Flow Control Enable: atufc, atufc?

Protocol Version: 1
Invocation: at$ufc <0x01> or <0x00>
Description: The flow control enable command is used to enable or disable flow control. Sending a value of 0x01 will enable flow control. Send any other value will disable flow control. If flow control is enabled, the read command will respond with 0x01, otherwise 0x00.
Example: Enable flow control
at$ufc 01

6.6.3 Parity Enable: at$upar, at$upar?

Protocol Version: 1
Invocation: at$upar <0x01> or <0x00>
Description: The parity enable command is used to enable or disable parity. The BMD module only has Odd parity. Sending a value of 0x01 will enable parity checking. Sending any other value will disable parity checks. If parity is enabled, the read command will respond with 0x01, otherwise 0x00.
Example: Enable parity
at$upar 01

6.6.4 UART Pass-through Enable: atuen, atuen?

Protocol Version: 1
Invocation: at$uen <0x01> or <0x00>
Description: The UART pass-through enable command is provided as a means to enable and disable the pass-through UART only. If the pass-through UART is disabled, then no UART will be available on the

device after boot unless the AT Mode pin is driven low. Sending this command will not disable the AT Mode UART until a reset has been issued. Sending a value of 0x00 will disable the pass-through UART. Sending any other value will enable the pass-through UART. If UART pass-through is enabled, the read command will respond with ON, otherwise OFF.

Example: Enable Pass-through UART

at$uen 01

6.7 GPIO Commands

6.7.1 Configure GPIO: at$gcfg

Protocol Version: 1
Invocation: at$gcfg <pin> <direction> <pull>
Description: The input GPIO pin is configured with a direction and pullup type. The default power-up state is input, Hi-Z.
Note: *GPIO configurations are not saved across power cycles*.
Parameters:
<pin> refers to the pin mapping:
(BMD-200, Table 1), (BMD-300 Series, Table 4)
<direction> = pin direction:
00 = input | 01 = output
<pull> = internal pin pull-up or pull-down, 13KΩ (nominal):
00 = no pull | 01 = pull-down | 03 = pull-up
Return Codes: OK on success, ERR on invalid parameter
Example: Configure BMD-300 pin 13 (P0.00) as Output No-pull
at$gcfg 00 01 00

6.7.2 Write GPIO: at$gset

Protocol Version: 1
Invocation: at$gset <pin> <state>
Description: The selected GPIO pin, if set for output, is set to the selected state.
Parameters:
<pin> refers to the pin mapping:
(BMD-200, Table 1), (BMD-300 Series, Table 4)
<state> = pin state:
00 = low | 01 = high
Return Codes: OK on success, ERR on invalid parameter
Example: Drive BMD-300 pin 13 (P0.00) to GND
at$gset 00 00

6.7.3 GPIO Get Pin State: at$gread

Protocol Version: 1
Invocation: at$gread <pin>
Description: Reads the current state of a GPIO pin.
Parameters:
<pin> refers to the pin mapping:
(BMD-200, Table 1), (BMD-300 Series, Table 4)
<state> = pin state:
00 = low | 01 = high
Return Codes: 00 or 01 for a valid pin number, ERR on invalid parameter

Example: Read state of BMD-300 pin 13 (P0.00)
at$gread 00

6.7.4 GPIO Get Pin Configuration: at$gcget

Protocol Version: 1
Invocation: at$gcget <pin>
Description: The selected GPIO pin configuration is returned in the format of <pin> <dir> <pull>:
Parameters:
<pin> refers to the pin mapping:
(BMD-200, Table 1), (BMD-300 Series, Table 4)
<direction> = pin direction:
00 = input | 01 = output
<pull> = internal pin pull-up or pull-down, 13KΩ (nominal):
00 = no pull | 01 = pull-down | 03 = pull-up
Return Codes: <pin> <dir> <pull> on success, ERR on invalid pin number
Example: Get configuration for BMD-300 pin 13 (P0.00)
at$gcget 00

6.7.5 GPIO Configure Status Pin: at$gstc, at$gstc?

Protocol Version: 2
Invocation: at$gstc <pin> <polarity>
Description: The selected GPIO pin <pin> is configured as the connection status pin with <polarity> as the active state.
Parameters:
<pin> refers to the pin mapping:
00 – 06 (BMD-200, Table 1), BMD-300 Series: BMDware Values 0x00 - 0x04, See Table 4,
FF - Un-configure
<polarity> = active state polarity:
00 = active low | 01 = active high
Return Codes: OK on success for write, <pin> <polarity> for read, ERR on invalid pin number
Example: Configure BMD-300 pin 15 (P0.02) as the status pin with `Active Low` polarity
at$gstc 02 00

6.7.6 GPIO Read Status Pin: at$gstr?

Protocol Version: 2
Invocation: at$gstr?
Description: Read the current state of the configured status pin. Returns a value based on the active or inactive state of the status pin. This command does not return the logic level of the pin.
Return Codes: 00 for inactive, 01 for active, ERR on invalid pin number

6.8 Advertisement Control

6.8.1 Connectable Advertisement Enable: at$conadv, at$conadv?

Protocol Version: 2
Invocation: at$conadv <0x01> or <0x00>
Description: The Connectable Advertisement Enable command is used to enable or disable the connectable advertisement functionality. Sending a value of 0x01 will enable connectable advertisements. Sending a value of 0x00 will disable connectable advertisements. If connectable advertisements are enabled, the read command will respond with 0x01, otherwise 0x00.
Notes: This setting is saved across power cycles.

6.8.2 Connectable Advertisement Interval: at$cading, at$cadint?

Protocol Version: 1
Invocation: at$cadint <interval>
Description: The Connectable Advertisement Interval command is used to adjust the advertisement interval of the connectable broadcast of BMDware. Valid values are 20 ms to 2500 ms supplied as a two octect big-endian value.
Notes: This setting is saved across power cycles.
Example: Set Connectable advertising interval to 1200 ms
at$cadint 04b0

6.8.3 Connectable TX Power: at$ctxpwr, at$ctxpwr?

Protocol Version: 1
Invocation: at$ctxpwr <tx power>
Description: The Connectable TX power command sets the transmit power for the connectable advertisement data. This TX power is different from the beacon broadcast. See the Bluetooth low energy interface description for available transmit power values.

6.9 System and Miscellaneous commands

6.9.1 Unlock: at$unlock <1 to 19 byte password>

Protocol Version: 1
Invocation: at$unlock <1 to 19 byte alpha numeric password>
Description: This command attempts to unlock the device using the provided password. If the password is incorrect, the device will respond with LOCKED.
Example: Unlock with password set as `test`
at$unlock test

6.9.2 Reset: at$devrst

Protocol Version: 1
Invocation: at$devrst
Description: The reset command is used to reset the BMD module. This command causes the programmed bootloader to run for 2 seconds.

6.9.3 BMDware Version (read only): at$ver?

Protocol Version: 1
Invocation: at$ver?
Description: The version command reports the version of BMDware currently programmed to the module.

6.10 Bootloader Version (read only): at$blver?

Protocol Version: 2
Description: Reports the version of the Rigado bootloader programmed to the module. Only available if bootloader version is >= 3.2.0. If the bootloader version is not available, “unavailable” is returned.

6.11 API Protocol Version (read only): at$pver?

Protocol Version: 2
Description: Reports the protocol version for BMDware. See Appendix B for the protocol version command availability.

6.12 Start Bootloader: at$stbl

Protocol Version: 2
Description: Starts the bootloader and commands it to run for 3 minutes. This command only invokes this operation if at least RigDFU version 3.2.0 is installed. Previous versions of the bootloader do not support this operation and the normal 2 second startup time will apply.

6.13 Restart BMDware: at$restart

Protocol Version: 2
Description: Restarts BMDware immediately. The bootloader is not started when this command is used.

6.14 Get Hardware Info (read only): at$hwinfo?

Protocol Version: 2

Description: Retrieves information about the hardware on which BMDware is running. Returns the following formatted string:

NRF5<x>/<Flash>/<RAM>

<x> = 1 or 2

<Flash> = Amount of flash space in kB, typically 512 or 256

<RAM> = Amount of RAM available in kB, typically 64, 32, or 16

If for some reason the above information is not available or not recognized, “Unknown Part!” is returned.

6.15 Device Name: at$name, at$name?

Protocol Version: 2

Description: Sets or gets the Device Name advertised as part of the connectible advertisement. The name may be set to any combination of 1 to 8 ASCII characters. Invalid input will cause an ‘ERR’ response from BMDware.

Example: Set name to BMD300

at$name BMD300

6.15.1 Set Password: at$password

Protocol Version: 1

Invocation: at$password <1 to 19 byte alphanumeric password>

Description: The set password command is used to set or change the password currently used to protect the Bluetooth Low Energy and AT command interfaces. The password is provided as a non-null terminated string. All characters of the password must be ASCII printable characters. The ‘ ‘ (space) character is not allowed.

Example: Set password to atMod3pa$$w0rd

at$password atMod3pa$$w0rd

7. Appendix A: Bootloader version information

The bootloader version information is described by the following typedefs:

```
typedef enum version_type_e
{
    VERSION_TYPE_RELEASE = 1,
    VERSION_TYPE_DEBUG = 2
} version_type_t;

typedef enum softdevice_support_e
{
    SOFTDEVICE_SUPPORT_S110 = 1,
    SOFTDEVICE_SUPPORT_S120 = 2, //Not used
    SOFTDEVICE_SUPPORT_S130 = 3,
    SOFTDEVICE_SUPPORT_S132 = 4,
    SOFTDEVICE_SUPPORT_RESERVED2,
    SOFTDEVICE_SUPPORT_RESERVED3,
    SOFTDEVICE_SUPPORT_RESERVED4,
    SOFTDEVICE_SUPPORT_RESERVED5
} softdevice_support_t;

typedef enum hardware_support_e
{
    HARDWARE_SUPPORT_NRF51 = 1,
    HARDWARE_SUPPORT_NRF52 = 2,
    HARDWARE_SUPPORT_RESERVED1,
    HARDWARE_SUPPORT_RESERVED2,
    HARDWARE_SUPPORT_RESERVED3,
    HARDWARE_SUPPORT_RESERVED4,
    HARDWARE_SUPPORT_RESERVED5
} hardware_support_t;

typedef struct rig_firmware_info_s
{
    uint32_t magic_number_a;        //Always 0x465325D4
    uint32_t info_size;             //Size of this structure
    uint8_t version_major;
    uint8_t version_minor;
    uint8_t version_rev;
    uint32_t build_number;
    version_type_t version_type;
    softdevice_support_t sd_support;
    hardware_support_t hw_support;
    uint16_t protocol_version;
```

```
    uint32_t magic_number_b;        //Always 0x49B0784C
} rig_firmware_info_t;
```

When the Get Bootloader Version command is executed over BLE, the above structure is returned as an array of bytes without the magic number fields. Over AT mode, the bootloader version is requested with the 'at$blver' command which only provides the version number. It does not provide the protocol version.

8. Appendix B: Protocol Version Support

The following tables show the commands supported by each protocol version. A few commands were added to version 3.0.0 before a protocol version was tracked. These commands are denoted as 1 (3.0.0). If the command returns an error, verify the version is at least 3.0.0 for those commands. Version 2 was added to BMDware version 3.1.0.

8.1 BLE Commands

Command (Octect)	Added	Deprecated
20	1	
21	1	
22	1	
23	1	
31	1	
40	1	
41	1	
42	2	
43	2	
50	1 (3.0.0)	
51	1 (3.0.0)	
52	1 (3.0.0)	
53	1 (3.0.0)	
54	1 (3.0.0)	
55	2	
56	2	
57	2	
60	2	
61	2	
62	2	
63	2	
F8	1	
4B102F37	2	
03563057	1	
96DFF40B	2	

Table 5 – BMDware BLE Command Availability

8.2 AT Commands

Command	Added	Deprecated
at$buuid	1 (3.0.0)	
at$uuid	1	2
at$bmjid	1 (3.0.0)	
at$mjid	1	2
at$bmnid	1 (3.0.0)	
at$mnid	1	2
at$badint	1 (3.0.0)	
at$adint	1	2
at$btxpwr	1	
at$ben	1	
at$cusbcn	1	
at$cbclr	1	
at$bcal	1	
at$ubr	1	
at$ufc	1	
at$upar	1	
at$uen	1	
at$gcfg	1 (3.0.0)	
at$gset	1 (3.0.0)	
at$gread	1 (3.0.0)	
at$gcget	1 (3.0.0)	
at$gstc	2	
at$gstr	2	
at$conadv	2	
at$cadint	2	
at$ctxpwr	1	
at$unlock	1	
at$devrst	1	
at$ver	1	
at$blver	2	
at$pver	2	
at$stbl	2	
at$restart	2	
at$hwinfo	2	
at$name	2	
at$password	1	

Table 6 – BMDware BLE Command Availability

9. Known Errata

9.1 Connected Status Pin (BMD-300 Only, BMDware 3.1.0 (50))

9.1.1 Description

Configuring certain BMD-300 pins as the status pin erroneously causes another GPIO pin to be configured as an output.

9.1.2 Details

When setting the status pin with the BMDware Value, the mapped pin is configured as the status pin as expected. However, the P0.x pin matching the BMDware Value is erroneously configured as an output. The erroneously configured P0.x pin is not affected by changes in the state of the status pin and its status as an output is not reflected when using the at$gcget or BLE GPIO Config Get commands. Commands to configure the P0.x pin will overwrite the erroneous configuration and the P0.x pin will then work as expected.

The updated set of available BMD-300 pins for the status indication is as follows:

BMD-300 Pin	Name	BMDware Value
13	P0.00	0x00
14	P0.01	0x01
15	P0.02	0x02
19	P0.03	0x03
20	P0.04	0x04

These pins are documented as the only available pins for the status pin although any free GPIO may be configured as the status pin.

9.1.3 Example of the errata

Configures BMD-300 Pin 10, Name P0.29, BMDware value 0x14 (**20**) as the status pin with:
`at$gstc 14 00`
Errata: Configures BMD-300 Pin 10 as expected, but sets BMD-300 Pin 38 (P0.**20**), as an output driven to its default value.

9.1.4 Additional Notes

If the affected pin not used (NC), then the errata can be ignored.

9.1.5 Workaround (Output pins)

If the affected pin is used as an output, make sure the affected pin is configured after configuring the status pin.

9.1.6 No Workaround (Input pins)

If the affected pin is used as an input, then the potential for multiple drivers on the same circuit exists. We strongly recommend avoiding this combination.

10. Revision History

Date	Revision	Author	Notes
05/27/2015	1.0	EPS	Initial Release
06/04/2015	1.1	EPS	Fix AT mode read UART parity command definition
08/14/2015	1.2	EPS	Fix available parity type, other minor edits
11/16/2015	1.3	BR	Added GPIO commands and related items Added at$conadv command Added at$buuid, at$bmjid, at$bmnid and at$badint Deprecated at$uuid, at$mjid, at$mnid and at$adint Updated Pin Description table Moved revision history to the end of the document
11/23/2015	1.3	EPS	Updates
03/01/2016	1.4	BR	Corrected response to 0x53 Get GPIO
03/29/2016	2.0	BR	Added BMD-300 information throughout
04/20/2016	2.0	EPS	Reorganized command sections, added new sections, add new at commands
04/26/2016	2.1	EPS	Added AT Mode UART configuration, fix pin number in AT Mode section, new logo
04/29/2016	2.2	EPS	Update with Errata regarding Connection Status pin, update documentation for status pin, add AT Mode examples and information regarding input and UART parameters for AT Mode
05/06/2016	2.3	EPS	Fix restart command bytes, fix section numbering