# iBT20-UART COMBO

## Command and Response

| | | |
|---|---|---|
| Doc. Name | : | iBT20-UARTCombo-Rev0.2.01.docx |
| Date | : | 17 February, 2017 |
| Revision | : | 0.2.01 |

## Table of Content

## Table of Figures

## 1. General Descriptions

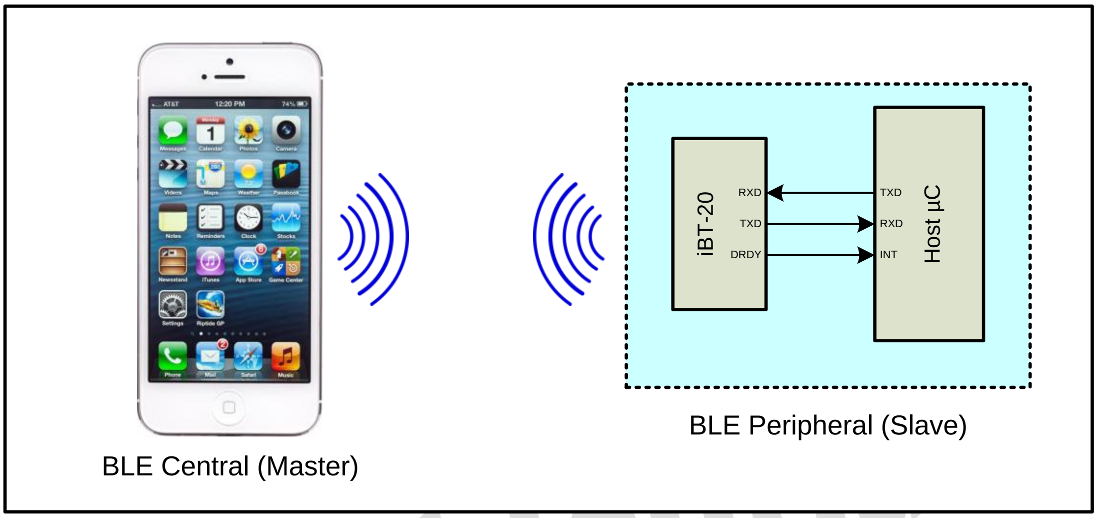


**Figure 1 System Block Diagram**

iBT20-UART COMBO (iBT20) is a Bluetooth LE module designed for data transfer between a host microcontroller (Host-µC) and a BLE central device, normally a mobile phone. Table 1 below shows the supporting models of the iBT20 module.

| Model Number | Supporting Profile and Service | Firmware Code |
|---|---|---|
| iBT20-UC-GT | Device Information Profile<br>GATT Profile<br>Battery Service | XAA1101xx.hex |

**Table 1 iBT20-UART Model Table**

## 1.1. Interfacing Signals

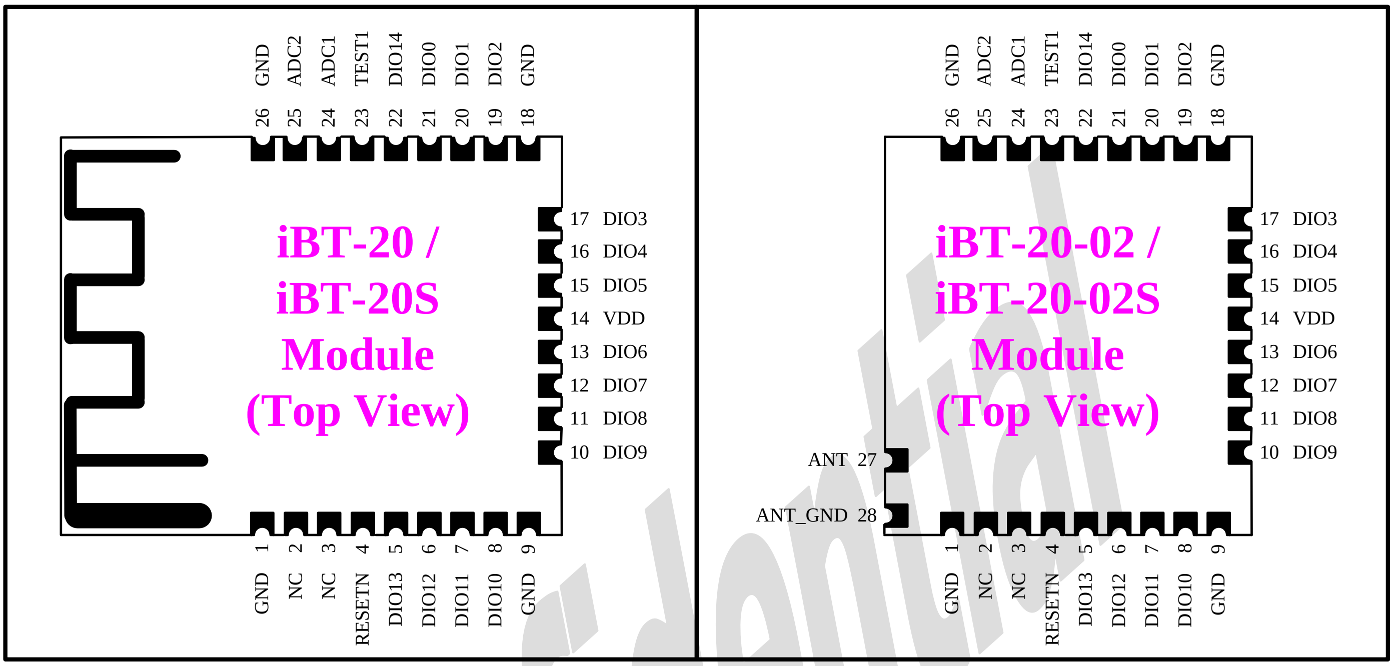


**Figure 2 iBT20 Module**

The iBT20 module communicates with the Host-µC through the UART interface together with a data ready (DRDY) status signal.

| iBT20 Module | | I/O | Description |
|---|---|---|---|
| Pin No | Name | | |
| 11 (DIO8) | TX | Output | This is the iBT20 UART transmit data pin. It is to be connected to the UART receive data pin of the Host-µC |
| 7 (DIO11) | RX | Input | This is the iBT20 UART receive data pin. It is to be connected to the UART transmit data pin of the Host-µC |
| 17 (DIO3) | DRDY | Output | This pin is used to indicate to the Host-µC that data is ready for it to receive or a change in connection status. It should be connected to an external interrupt pin of the Host-µC.<br>This pin will be in input mode after iBT20 power up reset. An external pull-up / pull-down resistor is required to be added to this pin for setting up its inactive state.<br>When a pull-down resistor is added to this pin, a low to high change (rising edge) on this pin indicates for data ready.<br>When a pull-up resistor is added to this pin, then a high-to-low change (falling edge) on this pin indicates for data ready. |
| 24 (ADC1) | AIN0 | Input | Analog Input |
| 25 (ADC2) | AIN1 | Input | Analog Input |

**Table 2 Interface Signals**

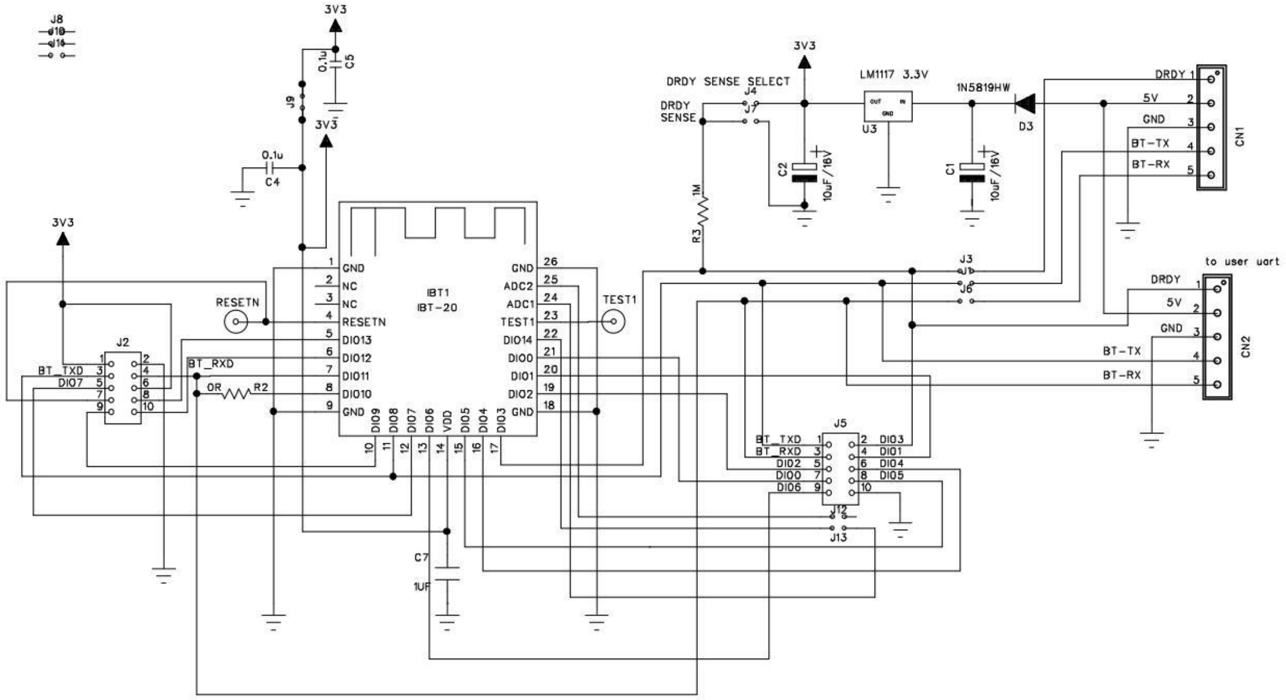
## UART Setting

For correct communication between the iBT20 and the Host-µC, please configure the Host-µC's UART port as below

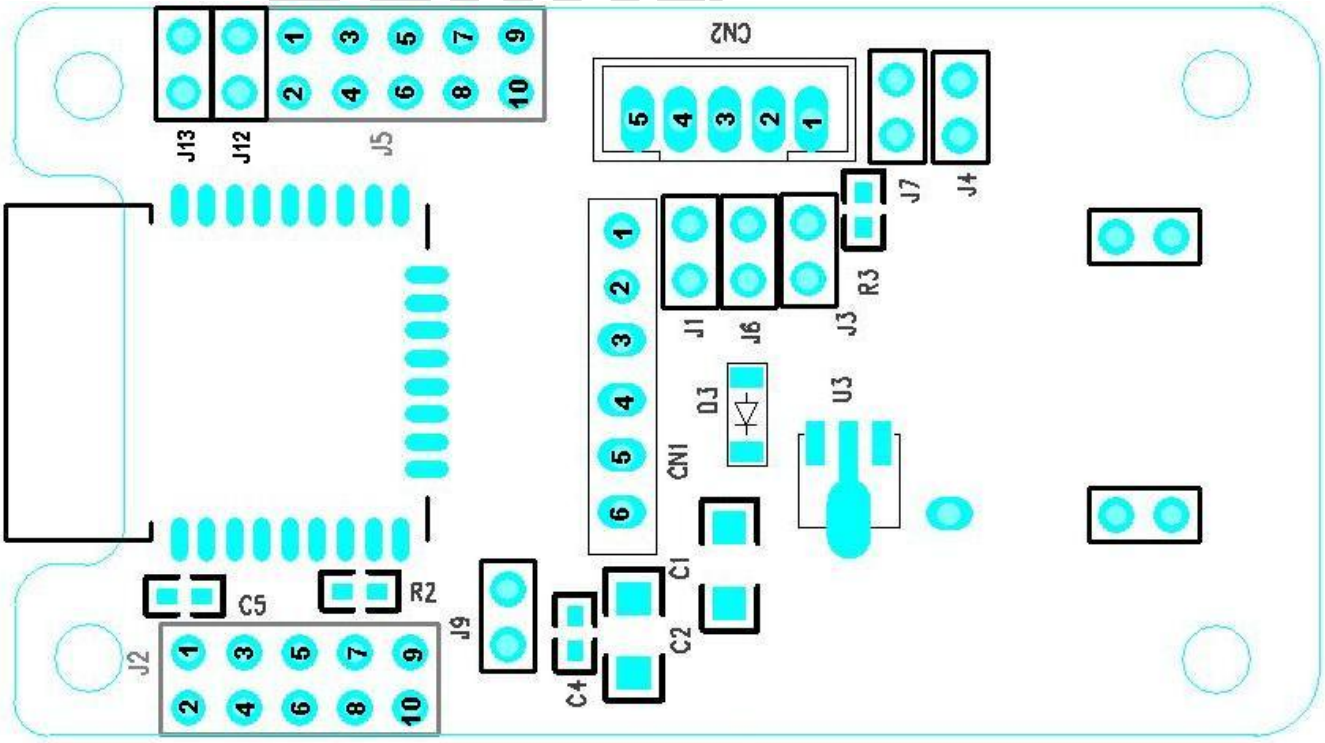
| | | |
|---|---|---|
| Baud Rate | : | 9600 (power up default) |
| No. of Data Bit | : | 8 |
| No. of Stop Bit | : | 1 |
| Parity | : | No parity |

# 2. iBT-20 Demo Board

## 2.1. iBT-20 Demo Board Schematic



## 2.2. iBT-20 Demo Board Component Side Silk Screen



## 2.3. iBT-20 Demo Board Connectors

### 2.3.1. CN1 - To CP2102 USB –to-UART Interface Board

This connector is used to connect iBT-20 to a PC through a USB-to-UART adaptor dongle. ***Please note that when CN1 is connected, CN2 must be removed***.

| Pin Number | Name | Type | Description |
|---|---|---|---|
| 1 | NC | | No Connect |
| 2 | NC | | No connect |
| 3 | 5V | Power | 5V Power Input from External |
| 4 | GND | Power | Ground |
| 5 | BT-TX | Output | Bluetooth UART TX signal |
| 6 | BT-RX | Input | Bluetooth UART RX signal |

### 2.3.2. CN2 - To External MCU

This connector is used to connect iBT-20 to an external MCU. ***Please note that when CN2 is connected, CN1 must be removed***.

| Pin Number | Name | Type | Description |
|---|---|---|---|
| 1 | DRDY | Output | Data Ready Signal |
| 2 | 5V | Power | 5V Power Input from External |
| 3 | GND | Power | Ground |
| 4 | BT-TX | Output | Bluetooth UART TX signal |
| 5 | BT-RX | Input | Bluetooth UART RX signal |

### 2.3.3. J1, J3, J6 Jumpers

These 3 jumpers are used to connect / disconnect the below signals from iBT-20 to CN1 connector

| Jumper | Position | Description |
|---|---|---|
| J1 | Open | Disconnect BT-TX signal to CN1 |
| | Close | Connect BT-TX signal to CN1 |
| J3 | Open | Disconnect DRDY signal to DRDY Test Point |
| | Close | Connect DRDY signal to DRDY Test Point |
| J6 | Open | Disconnect BT-RX signal to CN1 |
| | Close | Connect BT-RX signal to CN1 |

### 2.3.4. J2 – Header

| Pin Number | Name | Type | Description |
|---|---|---|---|
| 1 | 3.3V | Power | 3.3V Power Output |
| 2 | GND | Power | Ground |
| 3 | BT-TX | Input | Bluetooth UART TX signal |
| 4 | BT-RX | Output | Bluetooth UART RX signal |
| 5 | DIO7 | I/O | General Purpose I/O Pin |
| 6 | 3.3V | Power | 3.3V Power Output |
| 7 | RESETN | Input | Active low iBT-20 module reset |
| 8 | DIO13 | I/O | General Purpose I/O Pin |
| 9 | DIO9 | I/O | General Purpose I/O Pin |
| 10 | DIO12 | I/O | General Purpose I/O Pin |

### 2.3.5. J4, J7 – DRDY Sense

These 2 jumpers are used to select the in-active state of the DRDY signal.  Only one jumper should be connected

| Jumper | Position | Description |
|---|---|---|
| J4 | Open | Not connected |
| | Close | The DRDY signal is active low |
| J7 | Open | Not connected |
| | Close | The DRDY signal is active high |

### 2.3.6. J5 - Header

| Pin Number | Name | Type | Description |
|---|---|---|---|
| 1 | BT-TX | Output | Bluetooth UART TX signal |
| 2 | DRDY | Output | Data Ready Signal |
| 3 | BT-RX | Input | Bluetooth UART RX signal |
| 4 | DIO1 | I/O | General Purpose I/O Pin |
| 5 | DIO2 | I/O | General Purpose I/O Pin |
| 6 | DIO4 | I/O | General Purpose I/O Pin |
| 7 | DIO0 | I/O | General Purpose I/O Pin |
| 8 | DIO5 | I/O | General Purpose I/O Pin |
| 9 | DIO6 | I/O | General Purpose I/O Pin |
| 10 | GND | Power | Ground |

### 2.3.7. J9 - Header

Always short together.  Not used.

| Jumper | Pin | Description |
|---|---|---|
| J9 | 1 | Shorted on PCB to J9 pin 2 |
| | 2 | Shorted on PCB to J9 pin 1 |

### 2.3.8. J12 - Jumper

This jumper can be used to connect an external analog voltage to ADC2 pin for analog signal measurement.  The voltage level must lie between GND and 3.3V.

| Jumper | Pin | Description |
|---|---|---|
| J12 | 1 | Not connected |
| | 2 | Connect to iBT-20 ADC2 pin for analog voltage measurement |

### 2.3.9. J13 - Jumper

This jumper, when closed will connect ADC1 with DIO14 pin.

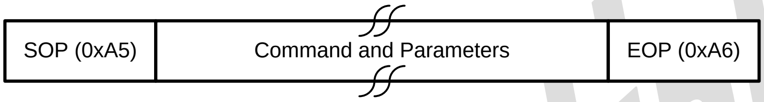
| Jumper | Pin | Description |
|---|---|---|
| J13 | 1 | Connect to iBT-20 ADC1 pin for analog voltage measurement |
| | 2 | Connect to iBT-20 DIO14 pin |

## 3. Communications

P Packet based communication is used between the Host-µC and the iBT20 module. All communications are controlled by the Host-µC. The Host-µC initiates a command packet and the iBT20 responses a with a response packet to the requesting command. In case that iBT20 has data to be sent to the Host-µC, it can activate the DRDY signal to request the Host-µC to initiate a command packet.

### 3.1. Packet Format

All commands, data and responses are transferred in packet format. Each packet starts with a Start of Packet (SOP) and ends with an End of Packet (EOP). Commands, responses and parameters and filled in between the SOP and EOP.



There are 3 characters that need to be special handled inside the command and parameter field. They are the SOP (0xA5), EOP (0xA6) and the Special Character Header (SCH, 0x5A). These 3 characters will be coded in the transmitting side and decoded in the receiving side.

| Symbol | Original Data | Coded Data |
|---|---|---|
| SOP | 0xA5 | 0x5A, 0x5A |
| EOP | 0xA6 | 0x5A, 0x5C |
| SCH | 0x5A | 0x5A, 0x5B |

### 3.2. Command and Response

In order to save power, the iBT20 will enter power save mode after completing a command. A UART activity will wake up the iBT20 from power save mode. During wake up, iBT20 may not be able to receive a complete packet sending from the host uC and so no response packet will be issued from it. The Host-µC, if cannot receive the iBT20 response packet after waiting for $T_{RES}$ (response time) should assume that the previous packet has been loss and retransmit the packet to iBT20 until a valid response packet has received.

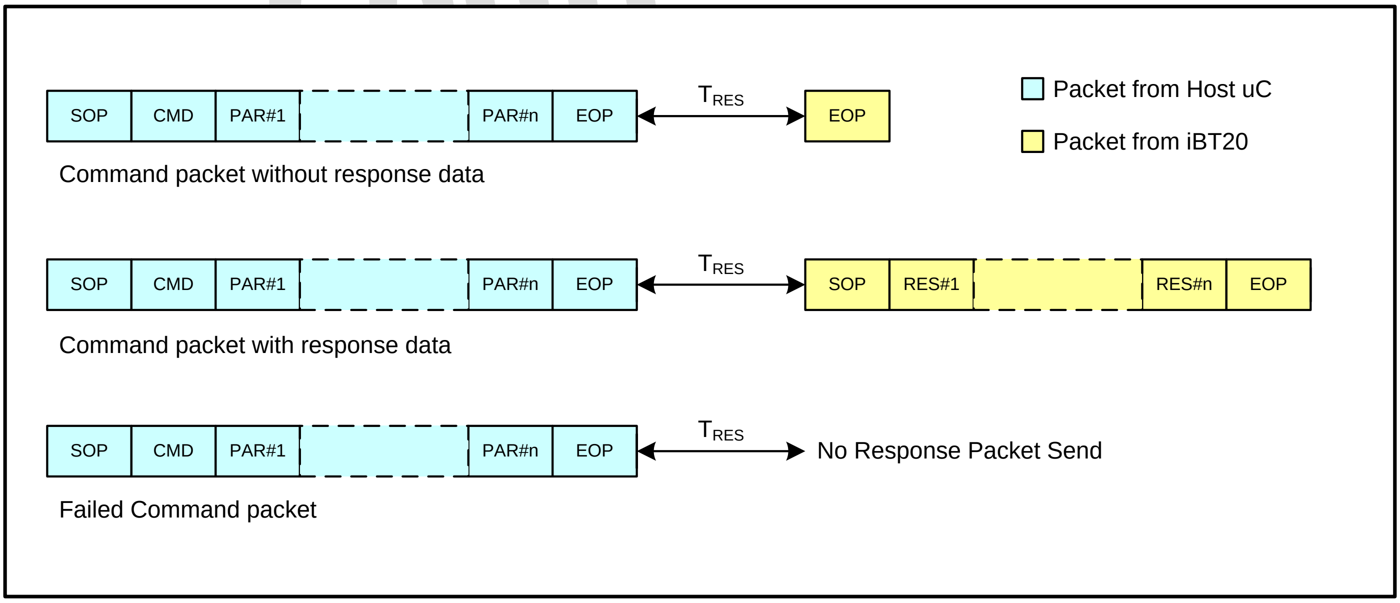


Figure 3 Command and Response Behavior

## 4. Commands

Commands can be divided into two categories.

Cat1 commands are communicated between the Host-µC and iBT20 only. Cat1 commands are used for setting up the functionality and getting the status of the iBT20 module.

Cat2 commands are communicated between the Host-µC, through iBT20 to the remote BLE central device. Cat2 commands are used for transferring data between the BLE central and the BLE peripheral.

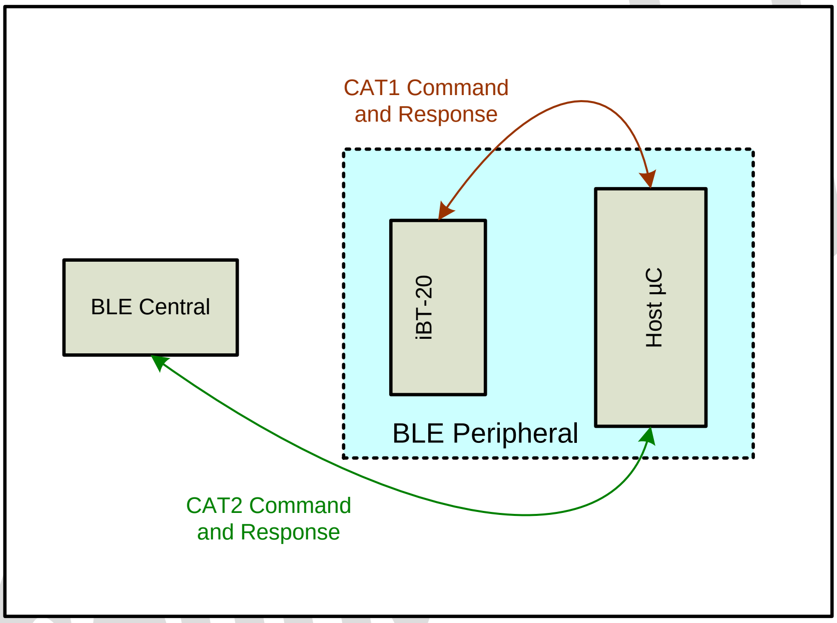


**Figure 4 Command Category**

| Code | Name | Category | Description | Command Validity |
|---|---|---|---|---|
| 0x01 | CMD_ADV_ENABLE | 1 | Enable BT Advertisement | S1,S3,S4 |
| 0x02 | CMD_ADV_DISABLE | | Disable BT Advertisement | S2,S3,S4 |
| 0x03 | CMD_READ_STATUS | | Read BT Status | S1,S2,S3,S4 |
| 0x06 | CMD_DISCON | | Disable Connection | S4 |
| 0x07 | CMD_READ_ADC | | Read ADC Value from VDD, ADC1 and ADC2 | S1,S2,S3,S4 |
| 0x08 | CMD_READ_RSSI | | Read RF RSSI | S4 |
| 0x09 | CMD_READ_FEATURE | | Read iBT20 Feature | S1,S2,S3,S4 |
| 0x0A | CMD_SET_CON_INTERVAL | | Set BT Connection Interval | S1 & S3 |
| 0x0B | CMD_SET_DEVICE_NAME | | Set BT Device Name | S1 & S3 |
| 0x0C | CMD_SET_UART_BR | | Set UART Baud Rate | S5 |
| 0x04 | CMD_WRITE_DATA | 2 | Send Data to BLE Central | S4 |
| 0x05 | CMD_READ_DATA | | Get Data from BLE Central | S4 |

**Table 3 Command Summary**

S1 – advertisement off
S2 – advertisement on
S3 – non-connected
S4 – connected
S5 – advertisement off and non-connected

## 4.1. CMD_ADV_ENABLE (0x01)

This command is used to enable iBT20 to send the advertisement packet at a fixed interval of time defined by the ADV_INV parameter. This command can also be used to change the advertising interval. The time between each advertisement packet, the advertisement interval, can be calculated by the below equation. For example, ADV_INV=0x00A0 means an advertising interval of 100mS. The power-up default of the advertising interval is 1.28S, ADV_INV = 0x0800.

Advertising Interval = 0.625ms x ADV_INV[15:0]

The range of the advertising interval, as defined in the Bluetooth specification, should be between 20mS to 10.24S. That is, the value of ADV_INV should be in between 0x0020 to 0x4000. iBT20 will not perform the validity check of the ADC_INV, it is the responsibility of the Host-μC to send an in-range ADV_INV data.

An invalid CMD_ADV_ENABLE command caused by either missing or invalid ADV_INV parameters, will cause iBT20 to stop executing the command, same behavior as not receiving the command.

***Command from Host-μC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x01 | CMD_ADV_ENABLE |
| Parameter #1 | 1 | ADV_INV[7:0] | Low Byte of ADV_INV |
| Parameter #2 | 1 | ADV_INV[15:8] | High Byte of ADV_INV |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For successful command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| EOP | 1 | 0xA6 | Command executed successfully |

## 4.2. CMD_ADV_DISABLE (0x02)

This command is used to disable the iBT20 from sending the advertisement packet.

***Command from Host-μC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x02 | CMD_ADV_DISABLE |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| EOP | 1 | 0xA6 | Command executed successfully |

## 4.3. CMD_READ_STATUS (0x03)

This command is used to get the status of the iBT20 module. It will also report to the Host-μC if there have data that can be read from the write property UUIDs, data wrote to iBT20 by the BLE central. In this case, the DRDY signal will also be set to the active state. Byte 3 and Byte 4 of the response packet will indicate the UUID that has data. Host-μC, should issue a read data command, CMD_READ_DATA, with the indicated UUID to get back the data.

If there's only one UUID has data to be read, iBT20, after receiving the read data command to the UUID and sending out the data to the Host-μC, will de-activate the DRDY signal and further CMD_READ_STATUS command will response with the byte1 bit 3= '0' and no UUID parameter bytes.

If there are more than one UUID has data to be read, iBT20, after receiving the read data command to the UUID, will keep the DRDY signal in active state and further CMD_READ_STATUS command will response with the byte1 bit 3= '1' and the UUID number of the next data ready port.

Please note that the DRDY signal will also be activated for a change in connection state, ie. change from disconnect state to connected state and vice versa. In this case, there should be no UUID data follows.

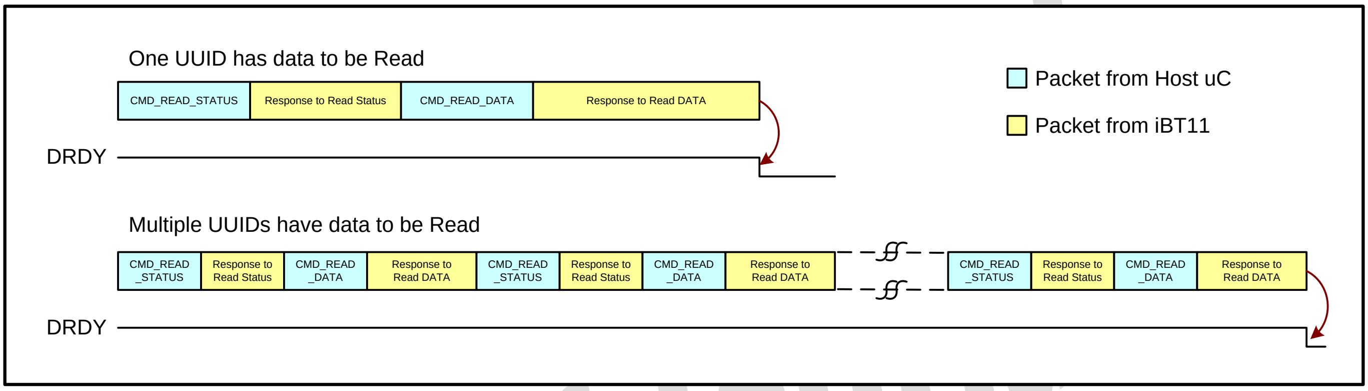


**Figure 5 Read Status Flow**

***Command from Host-μC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x03 | CMD_READ_STATUS |
| EOP | 1 | 0xA6 | |

***Response from iBT20***

For succeeded command, without UUID data

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Response #1 | 1 | STATUS | Bit 0 = 1, Connected<br>Bit 1 = 1, Bonded<br>Bit 2 = 1, Advertising<br>Bit 3 = 0, No Data for Host-μC to read<br>1, connection state change notification<br>Bit4 – bit 7, RFU |
| EOP | 1 | 0xA6 | |

For succeeded command, without UUID data

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Response #1 | 1 | STATUS | Bit 0 = 1, Connected<br>Bit 1 = 1, Bonded<br>Bit 2 = 1, Advertising<br>Bit 3 = 1, Data Ready for Host-μC to read<br>Bit4 – bit 7, RFU |
| Response#2 | 1 | UUID[7:0] | The UUID that has data for Host-μC to read. These 2 bytes only exists if bit3 of response#1 is set to '1' |
| Response#3 | 1 | UUID[15:8] | |
| EOP | 1 | 0xA6 | |

## 4.4. CMD_WRITE_DATA (0x04)

This is a Category 2 command used for the Host-μC to send data to the BLE Central device. This command is only applicable in the Bluetooth connected mode.

***Command from Host-μC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x04 | CMD_WRITE_DATA |
| UUID #0 | 1 | UUID[7:0] | Low byte of the UUID to be sent data |
| UUID #1 | 1 | UUID[15:8] | High byte of the UUID to be sent data |
| data #1 to data #n | n | | Data to be sent to the UUID |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| EOP | 1 | 0xA6 | Command executed successfully |

## 4.5. CMD_READ_DATA (0x05)

This is a Category 2 command used for the Host-μC to get data from the BLE Central device. This command is only applicable in the Bluetooth connected mode.

***Command from Host-μC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x05 | CMD_READ_DATA |
| UUID #0 | 1 | UUID[7:0] | Low byte of the UUID to be sent data |
| UUID #1 | 1 | UUID[15:8] | High byte of the UUID to be sent data |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| data #1 to data #n | n | | Data to be read from UUID |
| EOP | 1 | 0xA6 | |

## 4.6. CMD_DISCON (0x06)

This command is used to terminate a BLE connection with the BLE Central Device. If iBT20 is not in the connection state when receiving this command, iBT20 will still response with an EOP to signify to Host-μC that it is in disconnected state. iBT20 will only response with an ERP only if it cannot disconnect from the BLE central due to whatever reason.

***Command from Host-μC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x06 | CMD_DISCON |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| EOP | 1 | 0xA6 | iBT20 disconnected from BLE Central |

## 4.7. CMD_READ_ADC (0x07)

This command is used to control the iBT11 to perform ADC conversion for specific input. There are 4 different analog sources This command is used to control the iBT20 to perform ADC conversion for specific input. There are 3 different analog sources can be select by this command. Table 4 below lists the different ADC sources. An internal reference source will be used for the reference voltage of the analog-to-digital converter with a resolution of 10-bits. Each ADC conversion takes around 1mS to complete. Host-µC, after issuing this command, should wait until result available which will be pushed from iBT20 to Host-µC automatically.

| Source | Analog Source | Location | Description | Measured Result |
|---|---|---|---|---|
| 0 | VDD | iBT20 internal | This can be used to check the battery voltage when iBT20 is supplied from battery directly | Result = VDD voltage in mV scale |
| 1 | ADC1 | External pin | This can be used to check any external voltage source | Result =ADC1 voltage in mV scale |
| 2 | ADC2 | External pin | This can be used to check any external voltage source | Result =ADC2 voltage in mV scale |

**Table 4 ADC Source**

***Command from Host-µC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x07 | CMD_READ_ADC |
| Source | 1 | ADC_INPUT | Select for the ADC conversion source<br>0x00 – VDD 0x01 – ADC1<br>0x02 – ADC2<br>Others - invalid |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| ADC_LVL[15:0] | 1 | ADC_LVL[7:0] | Converted ADC voltage in mV scale |
| | 1 | ADC_LVL[15:8] | |
| EOP | 1 | 0xA6 | |

## 4.8. CMD_READ_RSSI (0x08)

This command is used to read the receive signal strength (RSSI) of the iBT20 module. The RSSI is only available after iBT20 is connected.

***Command from Host-µC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x08 | CMD_READ_RSSI |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| RSSI | 1 | RSSI[7:0] | RSSI in dBm (2’s complement) |
| EOP | 1 | 0xA6 | |

## 4.9. CMD_READ_FEATURE (0x09)

This command is used to read the firmware version and the supported feature of iBT20.

***Command from Host-μC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x09 | CMD_READ_FEATURE |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Sub-minor Version | 1 | | iBT20 firmware version in MajorVersion.MinorVersion.Sub-minorVersion format |
| Minor Version | 1 | | |
| Major Version | 1 | | |
| Feature | 2 | Supporting Profiles | Supported Profiles<br>Each bit represents a supported profile. A value of '1' means profile is supported<br>Bit 0 : Device Information profile<br>Bit 1 : GATT Profile<br>Bit 2 – 14 : reserved<br>Bit 15 : Battery Service |
| EOP | 1 | 0xA6 | |

## 4.10. CMD_SET_CON_INTERVAL (0x0A)

This command is used to set the connection interval of iBT20. The connection interval is the time between 2 connections initialized by a BLE Central. It can be used when advertisement is off and is not connected to any central. From the Bluetooth specification, connection interval can be ranged from 7.5mS to 4000mS with a resolution of 1.25mS. Furthermore, a Connection Latency parameter can also be set by this command. In order to reduce power consumption, a BLE peripheral can be designed to response once per a certain number of consecutive connection events. The number of consecutive events that the BLE peripheral will not response is the Connection Latency.

There are 9 pre-defined settings that can be selected by user or user can set the CON_INV and CON_LAT parameters by themselves. When pre-defined setting is used, excluding the command byte, one additional parameter is required, Table 5. When user want to define their own connection parameters, then excluding the command byte, 9 more parameters are required, Table 6.

| CON_OPT | Connection Interval | | Slave Latency | Timeout |
|---|---|---|---|---|
| | Min | Max | | |
| 0x00 | 40 (50mS) | 160 (200mS) | 0 | 1S |
| 0x01 | 40 (50mS) | 160 (200mS) | 2 | 2S |
| 0x02 | 40 (50mS) | 160 (200mS) | 4 | 4S |
| 0x03 | 80 (100mS) | 400 (500mS) | 0 | 2S |
| 0x04 | 80 (100mS) | 400 (500mS) | 2 | 5S |
| 0x05 | 80 (100mS) | 400 (500mS) | 3 | 6S |
| 0x06 | 400 (500mS) | 800 (1000mS) | 0 | 4S |
| 0x07 | 400 (500mS) | 800 (1000mS) | 1 | 6S |
| 0x08 | 240 (300mS) | 480 (600mS) | 2 | 6S |
| 0x80 | User define | User define | User define | User define |

*Command from Host-μC (use pre-defined setting)*

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x0A | CMD_SET_CON_INTERVAL |
| Connection Interval Option | 1 | CON_OPT | Connection Interval<br>others: Invalid |
| EOP | 1 | 0xA6 | |

**Table 5 COM_SET_CON_INTERVAL Pre-defined Setting Command Format**

*Command from Host-μC (user defined connection parameter)*

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x0A | CMD_SET_CON_INTERVAL |
| Connection Interval Option | 1 | CON_OPT | Connection Interval<br>0x80: set by using the CON_INV and CON_LAT parameters<br>others: Invalid |
| Connection Interval Min low | 1 | CON_INV_MN[7:0] | The Connection Interval must lie between 0x0006 (7.5mS) to 0x0C80 (4000mS). |
| Connection Interval Min high | 1 | CON_INV_MN[15:8] | |
| Connection Interval Max low | 1 | CON_INV_MX[7:0] | |
| Connection Interval Max high | 1 | CON_INV_MX[15:8] | |
| Connection Latency low | 1 | CON_LAT[7:0] | The Connection Latency must lie between 0x0000 to 0x01F3. A value of 0x0 means iBT20 will response to every connection event. A value of 1 means that iBT20 can response once per 2 connection events. A value of n means that iBT20 can response once per (n+1) connection events. |
| Connection Latency high | 1 | CON_LAT[15:8] | |
| Connection Timeout low | 1 | CON_TO[7:0] | This is the time before a connection to terminate. The value must lie between 10 to 3200 which represents a timeout of 100mS to 32S. |
| Connection Timeout high | 1 | CON_TO[15:8] | |
| EOP | 1 | 0xA6 | |

**Table 6 COM_SET_CON_INTERVAL User Defined Setting Command Format**

***Response from iBT20***

For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| EOP | 1 | 0xA6 | Command executed successfully |

## 4.11. CMD_SET_DEVICE_NAME (0x0B)

This command is used to set the Bluetooth device name of iBT20. It can only be used when the advertisement is off and is not connected to any BLE central. The Bluetooth Device Name of iBT20 supports up to 14 characters. In order to differentiate products with same Bluetooth device name, parts of the Bluetooth Address can be concatenated to the device name. The concatenation of the Bluetooth address is an optional feature that can also be controlled by this command.

***Command from Host-µC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x0B | CMD_SET_DEVICE_NAME |
| Option | 1 | OPT | Device name and Connection option<br>Bit 0 = 0, Disable concatenation of the BD Address to Device name<br>Bit 0 = 1, Enable concatenation of the BD Address to Device name<br>Bit 1 = 0, Not Bond after connected<br>Bit 1 = 1, Enable Bond after connected<br>Bits 2-7 RFU |
| Device Name #0 to #14 | 15 | | 15 byte Device name, Device Name is a null terminated string, i.e. it is max 14 characters.<br>All 15 bytes of Device Name MUST be transmitted, dummy data should be packed to the string if it is shorter than 15 bytes.<br>If connection Option bit 0 is set, the 3 least significant bytes of the BD address will be converted to ASCII and concatenated to the device name. |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| EOP | 1 | 0xA6 | Command executed successfully |

## 4.12. CMD_SET_UART_BR(0x0C)

This command is used to set the UART communication baud rate between iBT20 and the Host-µC. The power reset default baud rate is 9600. Host-µC can change the baud rate to 19200, 38400, 57600 or 115200. iBT20, after receiving this command, will response a EOP packet to the Host-µC and then re-configure the UART port to the use the new baud rate. Beginning with the next command sends from the Host-µC, the new baud rate should be used.

Please note that this command is only valid in the non-connected mode. If this command is issued in the connected mode, a ERP will be returned.

***Command from Host-µC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0x0C | CMD_READ_VER |
| New Baud Rate | 1 | BR | BR = 0, 9600 baud<br>BR = 1, 19200 baud<br>BR = 2, 38400 baud<br>BR = 3, 57600 baud<br>BR = 4, 115200 baud<br>Others, invalid |
| EOP | 1 | 0xA6 | |

***Response from iBT20***
For succeeded command

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| EOP | 1 | 0xA6 | Command executed  successfully and the next received command should use the new baud rate |

## 4.13. CMD_OAD_MODE(0xFF)

It is used to set BT module to the Over The Air Download (OAD) mode such that its firmware can be downloaded over the air. iBT20 will be disconnected after the command is received successfully.  It will be advertised again after a few seconds with the OAD device name of “iBT20 OTA Service Manager” and service.  Then it will be ready for update over the air.

***Command from Host-μC***

| Function | Length (byte) | Value | Description |
|---|---|---|---|
| SOP | 1 | 0xA5 | |
| Command Code | 1 | 0xFF | CMD_OAD_MODE |
| SIG_CODE0 | 1 | 0x66 | Signature Code |
| SIG_CODE1 | 1 | 0x44 | Signature Code |
| SIG_CODE2 | 1 | 0x55 | Signature Code |
| SIG_CODE3 | 1 | 0xAA | Signature Code |
| EOP | 1 | 0xA6 | |

No response will be transmitted for this command.

## 5. Profiles

One propriety profile, one standard profile and one standard service are supported. They are the UART Profile, Device Information Profile and the Battery Service.

### GATT – Proprietary

| UUID | Type | Properties | Description | Size / Bytes |
|---|---|---|---|---|
| 0xFFF0 | Service | | | |
| 0xFFF3 | Characteristics | Read | Battery Level (ADC Value) | 1 |
| 0xFFF5 | Characteristics | Read | RSSI | 1 |
| 0xFFF6 | Characteristics | Write | Data | 8 |
| 0xFFF7 | Characteristics | Read | Data | 8 |
| 0xFFF8 | Characteristics | Notify | Data | 8 |
| 0xFFF9 | Characteristics | Write | Data | Max. 20 |
| 0xFFFA | Characteristics | Read | Data | Max. 20 |
| 0xFFFB | Characteristics | Notify | Data | Max. 20 |
| 0xFFFC | Characteristics | Write without Response | Data | Max. 20 |

### 5.2. Device Information - Standard

| UUID | Type | Properties | Description |
|---|---|---|---|
| 0x180A | Service | | |
| 0x2A29 | Characteristics | Read | Manufacturer Name |
| 0x2A24 | Characteristics | Read | Model Number String |
| 0x2A25 | Characteristics | Read | Serial Number String |
| 0x2A27 | Characteristics | Read | Hardware Revision String |
| 0x2A26 | Characteristics | Read | Firmware Revision String |
| 0x2A28 | Characteristics | Read | Software Revision String |
| 0x2A23 | Characteristics | Read | System ID |
| 0x2A2A | Characteristics | Read | IEEE 11073-20601 Regulatory Certification Data List |
| 0x2A50 | Characteristics | Read | PnP ID |

### 5.3. Battery Service - Standard

| UUID | Type | Properties | Description |
|---|---|---|---|
| 0x180F | Service | | |
| 0x2A19 | Characteristics | Read / Notify | Battery Level (Percentage) |