VBLE-01 USER MANUAL

Introduction:

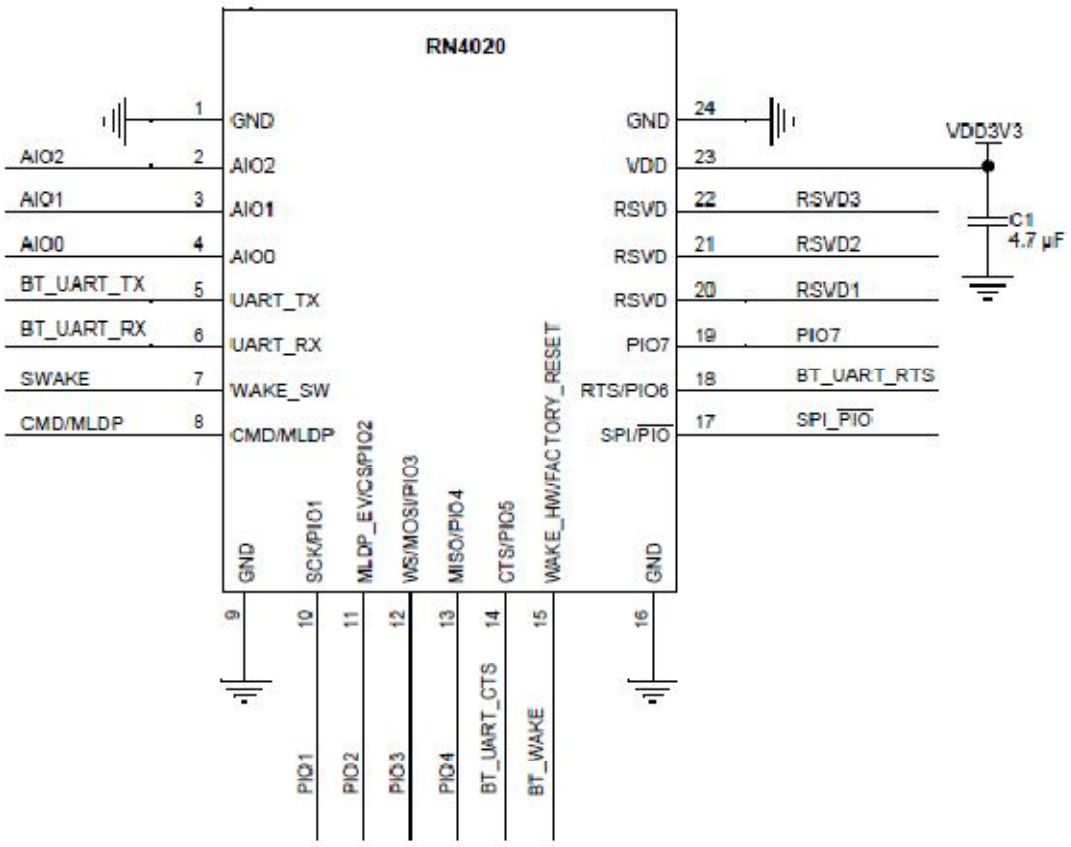
TheVBLE-01 Bluetooth module is a single mode Bluetooth Smart module that complies with Bluetooth Core Specification v4.1. Through its high-speed UART interface, this module can be configured to act as either a central or peripheral role when establishing a connection. This module supports 13 public profiles and 17 public services, which are adopted by the Bluetooth Special Interest Group (SIG). For all supported profiles and services, the RN4020 module can be configured to act as server and client roles at the same time. Furthermore, the RN4020 module supports the private Microchip Low-energy Data Profile (MLDP), which provides an asynchronous serial data connection between two RN4020 devices. Finally, the Microchip RN4020 module also supports a user-defined private profile/service, which can precisely fit a user's particular application. All configurations will be saved in on-board non-volatile memory (NVM), so users need to set up the module only once. The Microchip RN4020 module is easy to use and provides users with a fast-to-market, flexible, and powerful solution for BTLE technology.

VBLE-01 Features

- Smart module complies with Bluetooth Core Specificationv4.1
- Easy to Access
- Simple and efficient
- High-Speed UART interface
- Low powered module (Battery powered)

- Cost-effective
- Indication to denote the connection status
- On-board peripherals
- RS232/USB interface with PC

RN4020 Pin Diagram



RN4020 Pin Description:

Pin	Symbol	Description	Function
1	GND	Ground	Ground
2	AIO2	Bidirectional with programmable analog I/O.	1.65V input, 1.35V out, and 30 mA max out
3	AIO1	Bidirectional with programmable analog I/O.	1.65V input, 1.35V out, and 30 mA max out
4	AIO0	Bidirectional with programmable analog I/O.	1.65V input, 1.35V out, and 30 mA max out

5	UART_TX	UART Transmit	UART Output, 3.3V TTL
6	UART_RX	UART Receive	UART Input, 3.3V TTL
7	WAKE_SW	Deep Sleep Wake; active-high to wake module from Deep Sleep.	Input; weak pull-down
8	CMD/MLDP	CMD – Command Mode – Module enters Command mode where UART commands and responses sent over UART are exchanged between the RN4020 command interpreter and the MCU host. MLDP Mode – Data Mode – Data through UART is sent over the Bluetooth Low Energy connection to the remote device using MLDP data service.	Input; active-high to enter Command
9	GND	Ground	Ground
10	CONNECTION_LED/ PIO[1]/ SCK	Default state is output: Active-high indicates the module is connected to a remote device. Active-low indicates a disconnected state. Configurable as PIO[1] via software command. SCK for diagnostics and factory calibration if pin 17 is asserted.	• Green LED • PIO[1] • SCK
11	MLDP_EV PIO[2] CS	Default function is output used for MLDP data event indicator. Active-high indicates MLDP data received or UART console data pending. Low level indicates no events. Even only triggered in CMD mode, when CMD/MLDP (pin 8) is high. Configurable as PIO[2] via “|O” and “|I” commands. CS for diagnostics and factory calibration if pin 17 is asserted.	• MLDP Data Event (Red LED) • PIO[2] • CS
12	WS MOSI PIO[3]	Default function is an output used for an Activity indicator. High level indicates the module is awake and active. Low level indicates the module is in a Sleep state. Accessible as PIO[3] via “|O” and “|I” commands. MOSI for diagnostics and factory calibration if pin 17 is asserted.	• WS (Blue LED) • PIO[3] • MOSI
13	MISO PIO[4]	Trigger pin to generate event @PIOH and @PIOL. MISO for diagnostics and factory calibration if pin 17 asserted.	• PIO[4] • MISO

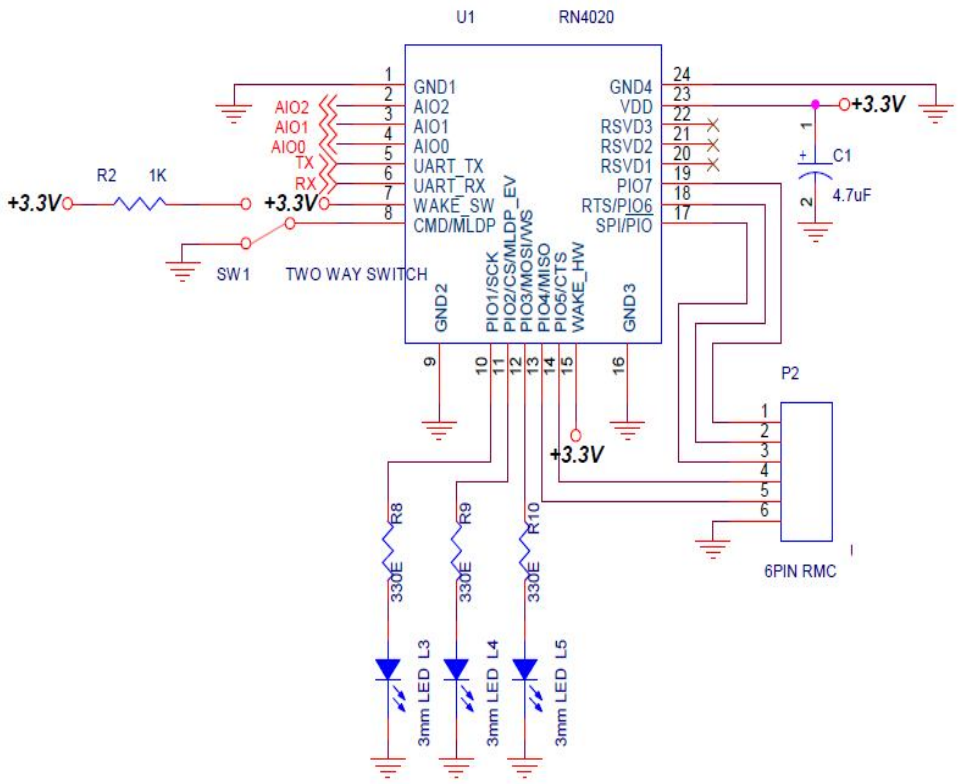
14	CTS PIO[5]	Reserved for CTS if hardware flow control is on the UART	• CTS (input) • PIO[5]
15	WAKE_HW FACTORY_ RESET	Hardware wake from Hibernate or Dormant state. Setting WAKE_HW (pin 15) high wakes the RN4020 module from Dormant mode. After powering up, if WAKE_HW flips up and down three cycles (putting the WAKE_HW pin into high, and then low, and then high again, is considered as one flip cycle) in the first five seconds, the RN4020 module performs a factory Reset. If WAKE_SW (SWAKE) is high when a factory Reset is performed, the factory Reset is complete; otherwise, it is a partial factory Reset that retains the device name, private service, and scripts.	Active-high; internal pull down
16	GND	Ground	Ground
17	SPI/PIO	SPI/PIO for pins 10-13, active-high.	Input with internal pull down; selects SPI on 10-13
18	PIO[6]	Reserved for RTS if hardware flow control on UART.	• RTS (output) • PIO[6]
19	PIO[7]	Spare PIO.	Spare PIO configurable as input or output
20	RSVD	Do not connect. Factory diagnostics.	No Connect
21	RSVD	Do not connect. Factory diagnostics.	No Connect
22	RSVD	Do not connect. Factory diagnostics.	No Connect
23	VDD	Supply Voltage	1.8V to 3.6V
24	GND	Ground	Ground

RN4020 UART Configurations:

The RN4020 module will be connected with the PC through UART. The TX and RX lines must be connected with the UART drivers. The default UART settings are

Parameter	Value
Baud Rate	115200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

RN4020 Connection Diagram



Command Descriptions:

VBLE-01 is a low-powered Bluetooth module that supports UART commands. It is very convenient to the user to develop their own application within limited time. The UART commands are divided into following groups as per the functionality. The groups of UART commands are

• **Set/Get Commands** - commands used to configure specific functions of the RN4020 module. The Set commands start with the letter S and are followed by one or two letters as the command identifier. A reboot is required for most Set commands to ensure the new settings will take effect. All Set commands have a corresponding Get command to output the configurations to the UART.

• **Action Commands** - used to initiate functionality, as well as display critical information.

• **Characteristic Access Commands** – to provide access to the values and configurations of characteristics defined by the profiles and services

• **Private Service Configuration Commands** – used to configure private services

• **Microchip MLDP Commands –** used for MLDP mode functions

• **RN4020 Scripting Commands** - Scripts are ASCII commands written into the NVM (Non Volatile Memory) of the RN4020 module

• **Remote Command** – Used to enable remote command feature of the module

• DFU Commands – Used to upgrade device firmware

UART Commands

Command Name	Command Description	Format	Example
Set Commands			
S-	**Serialized Name -** sets the serialized Bluetooth-friendly name of the device	S-,<string>	S-,MyDevice
SB	**Set Baud rate -** sets the baud rate of the UART communication 0 – 2400 1 – 9600 2 – 19200 3 – 38400 4 – 115200 5 – 230400 6 – 460800 7 – 921600	SB,<0-7>	SB,4
SDF	**Set firmware revision -** sets the value of the firmware revision characteristic in the Device Information Service	SDF,<text>	SDF,0.9
SDH	**Set hardware revision** - sets the value of the hardware revision characteristics in the Device Information Service	SDH,<text>	SDH,2.1
SDM	**Set model name -** sets the value of the model characteristics in the Device Information Service	SDM,<text>	SDM,RN4020
SDN	**Set manufacturer name -** sets the value of the manufacturer name characteristics in the Device Information Service	SDN,<text>	SDN,Microchip
SDR	**Set software revision -** sets the value of the software revision characteristics in the Device Information Service.	SDR,<text>	SDR,1.0
SDS	**Set serial number -** sets the value of the serial number characteristics in the Device Information Service	SDS,<text>	DS,12345678

SF	**Factory default** - resets the configurations to the factory default at the next reboot 1 - a majority of the settings will be restored to the factory default, but some settings, such as device name, device info, script and private services, stay the same. 2- all parameters are restored to factory default.	SF,<1,2>	SF,1
SM	**Set Timers in µs** - starts one of the application timers. The first parameter is the identifier of the timer to start, and the second parameter is the timer expiration time in microseconds (range between 0x00000001 and 0x7FFFFFFF, outside this range will stop the timer)	SM,<1-3>, <hex32>	SM,1,000f4240
SN	**Set name** - sets the device name	SN,<string>	SN,MyDevice
SP	**Set transmission power** - sets the transmission power. The TX power values for 0 -19.1 1 -15.1 2 -10.9 3 -6.9 4 -2.5 (default) 5 1.6 6 5.8 7 7.5	SP,<0-7>	SP,0
SR	Set features - sets the supported features of module. The input parameter is a 32-bit bitmap that indicates features to be supported. After changing the features, a reboot is necessary to make the changes effective. 0x80000000 - central 0x40000000 - Real-time Read 0x20000000 - Auto Advertise 0x10000000 - Enable MLDP 0x08000000 - Auto MLDP Disable 0x04000000 - No Direct Advertisement 0x02000000 - UART Flow Control 0x01000000 - Run Script After Power On 0x00800000 — Reserved 0x00400000 - Enable Authentication 0x00200000 - Enable Remote Command 0x00100000 - Do not Save Bonding 0x000E0000 - I/O Capabilities 0x00010000 - Block Set (Remote Mode) 0x00004000 - iOS Mode 0x00002000 - Server Only 0x00001000 - Enable UART in Script 0x00000800 - Auto-enter MLDP Mode 0x00000400 - MLDP without Status	SR,<hex32>	SR,20000000 // Set device as peripheral, and // automatically start advertisement

SS	**Set server services -** sets the services supported by the device in a server role 0x80000000 - Blood Pressure, Cycling Speed Cadence, Glucose, Health Thermometer, Heart Rate, Running Speed Cadence 0x40000000 - Battery 0x20000000 - Heart Rate 0x10000000 - Health Thermometer 0x08000000 - Glucose 0x04000000 - Blood Pressure 0x02000000 - Running Speed Cadence 0x01000000 - Cycling Speed Cadence 0x00800000 - Time 0x00400000 - Next DST change 0x00200000 - Reference Time Update 0x00100000 - Link Loss 0x00080000 - Immediate Alert 0x00040000 - TX Power 0x00020000 - Alert Notification 0x00010000 - Phone Alert Status 0x00004000 - Scan Parameters 0x00000001 - User Defined Private Profile	SS,<hex32>	SS,060000 // Support blood pressure and running speed cadence
ST	**Set connection parameters -** sets the initial connection parameters for future connections Connection interval, latency and timeout are often associated with how frequently a peripheral device needs to communicate with central and is therefore closely related to power consumption. Interval - 0x0006-0x0C80, The time interval of communication between two connected devices. (unit: 1.25 ms, Default 0006) Latency - 0x0000-0x01F3, The number of consecutive connection events that the peripheral does not need to communicate with central. Latency must less than: (Timeout * 10 / Interval * 1.25 - 1) Timeout - 0x000A-0x0C80, The maximum time between raw communications before the link is considered lost. (unit: 10 ms, Default 0064)	ST, <interval>, <latency>, <timeout>	ST,0064,0002,0064 // Set the interval to 125 ms, latency to 2, and time-out to 1 second
Get Commands Get commands are same as Set commands. It is start with G. By using get commands we can read the settings of the module. For example GB command returns 4 when the baudrate is set as 115200.			
Action Commands			
+	**Echo -** toggles the local echo on and off	+	+

@O	Output analog signal - set the analog port output. The first parameter can be1, or 2, which specifies the analog port number. The second parameter is only for analog output, which sets the output voltage in mV.	@O,<0-2>, <hex16>	@O,1,03E8 // Set AIO1 output voltage to be 1000 mV
@I	Input analog signal - Get the analog port input	@I,<0-2>	@I,1
|O	**Set PIO's output -** set the output (O) on the digital I/O pins (PIO1, PIO3, and PIO7). 0'b00000001 - PIO1 0'b00000010 - PIO2 0'b00000100 - PIO3 0'b00001000 - PIO7	|O,<hex8>, <hex8>	|O,07,05 // Set PIO1 and PIO3 output to be high and PIO2 low
+I	**Set PIO's input** - get the input (I) on the digital I/O pins (PIO1, PIO3, and PIO7).	|I,<hex8>	|I,06 // Read states of PIO2 and PIO3.
A	**Advertise** - used to start advertisement. When the "A" command is issued without a parameter, by default, the advertisement interval is 100 ms and advertising is indefinite. The "A" command can be followed by two 16-bit hex parameters, which indicates an advertisement interval in milliseconds and total advertisement window time in milliseconds.	A,<hex16>, <hex16>	A,0050,07D0 // Start advertisement with interval of 80 ms for 2 seconds
B	**Bond -** used to secure the connection and bond two connected devices (effective if two devices are already connected) 1 – the connection will be secured and two devices are considered bonded. 0 - the connection is secured and not bonded.	B,<0,1>	B,1
D	**Dump configuration** - displays critical information about the current device over the UART. The information are • Device MAC Address • Device Name • Device Connection Role • Connected Device • Bonded Device • Server Services • Features: • Transmit Power	D	D

E	**Establish connection** – starts the process to establish a connection with a peer peripheral device. The first parameter is the MAC address type (0 - for public address or 1 - for a random address), and second parameter is the MAC address of the peripheral device.	E,<0,1>, <mac address>	E,0,00035B0358E6 // Connect to peripheral with public address 00035B0358E6
F	**Start scan** - query the peripheral devices before establishing a connection. If no parameter is provided, the "F" command starts the active scan process with a default scan interval of 375 milliseconds and a scan window of 250 milliseconds. The user has the option to specify the scan interval and scan window as the first and second parameter, respectively, as a 16-bit hex value in milliseconds.	F,<hex16>, <hex16>	F,012C,00C8 // Start inquiry with 300 ms scan interval and 200 ms scan window
H	**Help -** sends a help page to the UART	H	H
J	**Observer role** - places the device into or out of an observer role	J,<0,1>	J,1 // Enter observer mode.
K	**Disconnect -** disconnect the active BTLE link. The "K" command can be used in a central or peripheral role. An error is returned if there is no connection.	K	K
M	**Get RSSI from peer -** obtain the signal strength of the last communication with the peer device. The signal strength can be used to estimate the distance between the device and its peer.	M	M
N	**Enter broadcast information** - place the RN4020 module into a broadcaster role and to set the advertisement content.	N,<hex>	N,11223344 // Place broadcaster role and set advertisement content to be 0x11, 0x22, 0x33, and 0x44.
O	**Enter dormant state -** places the module into a Dormant mode that consumes very little power, and can be issued by either a central or peripheral device	O	O
Q	**Retrieve connection status -** returns the Bluetooth connection status	Q,<1>	Q,1
R	**Reboot** - forces a complete device reboot (similar to a power cycle). It has one mandatory parameter of '1'	R,1	R,1
T	**Change parameter for current connection -** change the connection parameters, interval, latency, and time-out for the current connection	T,<interval>, <latency>, <timeout>	T,0190,0001,03E8 //interval 400 ms, latency 1, and timeout 1000 ms

U	**Unbond** - removes the existing bonding	U	U
V	**Firmware version -** displays the firmware version	V	V
X	**Stop scan** - stops the inquiry process.	X	X
Y	**Stop advertisement** - stops advertisement that was started by an "A" command	Y	Y
Z	**Stop connecting** - stops the connection process that was started by an "E" command	Z	Z
]A	**Enable I2C –** Enables I2C operation Clock is speed of I2C interface: 1 = 100 kHz, 4 = 400 kHz. PIO used to supply power to I2C bus. Valid values are 1, 2, 3, 7 to designate which PIO is used to power I2C.	]A,<clock>, <pio-power>	]A,1,1
]Z	**Disable I2C** – Disables I2C operatoin	]Z	]Z
]ER	**Read EEPROM** – Read the data from the address	]ER,<i2c_addr>, <mem>, <length>	]ER,00A0,0000,8
]EW	**Write EEPROM** – Write data to the EEPROM	]EW,<i2c_addr>, <mem>, <data>	]EW,0xA0,0000,255
]C	**I2C Bus Event -** Generates events on bus: 0 – Start, 1 – Restart, 2 – Stop, 3 – Wait for ACK, 4 – Send ACK, 5 – Send NACK.	]C,<event>	]C,0 //I2C Start function
]R	**I2C Read Data** - Reads data from I2C slave peripheral, max 32 bytes.	]R,<len>	
]W	**I2C Write Data** - Write data to I2C slave peripheral, max 32 bytes.	]W,<data>	
[	**PWM commands** - support up to four HIGH/LOW Pulse Width Modulation (PWM) patterns. The first parameter is used to specify the PWM pin (1,2,3 and 7). Next, are six 8-bit hex value parameters for two PWM patterns. The first three parameters are for the first pattern, and the next three parameters are for the second pattern.	[<pio>,<h1>,<l1>, <d1>, <h2>,<l2>, <d2>	
Characteristic Access Commands			
LC	**List Client Services -** lists the available client services and their characteristics	LC	LC

LS	**List Server Services -** lists the server services and their characteristics	LS	LS
CHR	**Read Value from Client Handle -** reads the content of the characteristic of the client service from a remote device by addressing its handle	CHR,<hex16>	CHR,001A
CHW	**Write Value to Client Handle -** writes the contents of the characteristic in the client service from a remote device by addressing its handle	CHW,<hex16>,<data>	CHW,001A,64 // Set the value of the characteristic with the handle value 0x001A to be 100 on the remote device
CURC	**Read configuration of client UUID -** reads the configuration of a characteristic in the client service from a remote device by addressing its UUID	CURC, <hex16>	CURC,2A19 // Read the configuration of the characteristic Battery Level with the UUID 0x2A19 from the remote device
CURV	**Read value of client UUID -** reads the value of a characteristic in the client service from a remote device by addressing its UUID	CURV, <hex16>	CURV,2A19 // Read the value of the characteristic
CUWC	**Client UUID notify/indicate start -** writes the configuration of a characteristic in the client service to a remote device by addressing its UUID	CUWC, <hex16>, <data>	CUWC,2A19,1 //Start notification on the remote device
CUWV	**Write value to client UUID -** writes the value of a characteristic in the client service to a remote device by addressing its UUID	CUWV,<hex16>,<data>	CUWV,2A19,64 // Write 100% to the remote device for the characteristic with the UUID 0x2A19
SHR	**Read value of server handle -** reads the contents of the characteristic of the server service on a local device by addressing its handle	SHR,<hex16>	SHR,001A // Read the local content of the characteristic with the handle 0x001A
SHW	**Write value to server handle -** writes the contents of the characteristic in the server service to a local device by addressing its handle	SHW, <hex16>, <data>	SHW,001A,64
SUR	**Read value of server UUID -** reads the value of the characteristic in the server service on a local device by addressing its UUID.	SUR,<hex16>	SUR,2A19
SUW	**Write value to server UUID** – writes the contents of the characteristic in the server service to a local device by addressing its UUID	SUW, <hex16>, <data>	SUW,2A19,64

Private Service Configuration Commands			
PC	**Set private characteristics UUID** – Sets the characteristics of the private service	PC,<hex128>, <8-bitChar>, <data_len>, <flag>	PC,1234567890ABCDEF1234567890ABCDEF,02,02 //Set characteristics of the service of the UUID 1234567890ABCDEF1234567890ABCDEF as read 2 bytes length
PF	**Set primary service UUID filter -** sets the primary UUID filter of the private service	PF,<UUID><Z>	PF,Z // Clear the filter
PS	**Set private service UUID -** sets the UUID of the private service.	PS,<128-bit UUID>	PS,010203040506070809000A0B0C0D0E0F
PZ	**Clear private service -** clears all settings of the private service and the private characteristics	PZ	PZ
MLDP Commands			
I	Enter MLDP mode - places the RN4020 module into MLDP simulation mode	I	I
SE	**Set MLDP security mode -** sets the security mode for MLDP communications 0 - no security 1 - MLDP data over the air will be encrypted 2 - MLDP data over the air will be authenticated	SE,<0-2>	SE,1
Scripting Commands			
LW	**Show script -** lists the current script that is loaded in the RN4020 module	LW	LW
WC	**Clear Script -** clears the script, if any, that is loaded in the RN4020	WC	WC
WP	**Pause script -** stops script execution	WP	WP
WR	**Run Script -** starts script execution	WR,<0-9>	WR,1
WW	**Write Script -** enters Script Input mode	WW	WW

Remote Command			
!	**Enter Remote command mode -** This remote command feature is built on top of MLDP, so it is a prerequisite to support MLDP before using the remote command feature 1 – enter remote command 0 – exit remote command	!,<0,1>	!,1 //Enter into remote command
DFU (Device Firmware Upgrade) Commands			
~	**Device Firmware update** - places the device into Device Firmware Service mode. To use this command, it is mandatory to enable the UART flow control. This command expects one input parameter. If the input parameter 1- DFU mode is set for the upgrade to be handled through the UART. 2 - DFU mode is set for the upgrade to occur OTA.	~,<0,1>	

RN4020 Scripting capability

The RN4020 script engine handles each script line by line. Each line can start with multiple spaces or tabs and end with a return or line feed. Even though spaces are generally not supported between ASCII commands and their parameters, which is the same as commands through the UART, spaces or tabs are supported in assignment and logic expressions, as shown in the following example. Comment lines can be added to the script. A comment line starts with the '#' character and lasts the whole line. The script engine will ignore the comment line and jump to the next script line once a comment line is detected.

Variables: The RN4020 script engine defines two variables: $VAR1 and $VAR2. Variable names are case sensitive. Using the '=' operator, the value of the variables can be assigned to a constant value, or a value that is returned by an ASCII command. For instance, the following script line assigns the value 0x1234 to the variable $VAR1:
$VAR1 = "1234"

List of events and event labels: A script is driven by events. Currently, there are 11 events defined. All event scripts start with an event label, which is then followed by one or more logic operations or ASCII commands. Once an event is triggered, if an event label is defined, control is passed over to the script engine. The script engine begins executing the commands that are listed following the event label until the end of script or until another event label is encountered. The events and event labels are

Power On	@PW_ON
Timer1 expired	@TMR1
Timer2 expired	@TMR2
Timer3 expired	@TMR3
Connected	@CONN
Disconnected	@DISCON
PIO4 Input Change to Low	@PIOL

PIO4 Input Change to High	@PIOH
High Priority Alert	@ALERTH
Low Priority Alert	@ALERTL
Alert Off	@ALERTO

Handle Association: An I/O port can be associated with the handle of a server characteristic. Once the handle receives requests from a peer device to read or write, the I/O port is read or written, respectively, without further instruction. The analog port and four digital ports can be associated with a handle. The associated handle can be identified by the proceeding identifier "%". For instance, the following script line associates server characteristic handle 0x0021 with a read operation of analog port AIO2, so whenever the peer device wants to read handle 0x0021, AIO2 is read and the value will be returned to the peer device.
%0021 = @I,2

Procedure

1. Supply power to the module. If you provide +5V adapter supply to the module, keep J1 in +3.3VA side, else if you provide battery power keep J1 in +3.3VB side.
2. Open WinXtalk or Hyperterminal with the baudrate of 115200
3. Enter the following code
4. Now the module is ready to act as a peripheral. For the master use your smart phone.
5. Switch on Bluetooth in the smart phone
6. The device will be available in the device list in the Bluetooth settings
7. Install nRF connect master control panel in your phone
8. Open nRF connect master control panel. The paired devices with the services and characteristics are displayed in the phone.
9. Open the service by clicking over it
10. Read the ADC readings in your phone.

Example Programs:

1. **Program to read the ADC values in the smart phone**

```
+                   //Echo On
SF,1                //Factory Reset
SS,00000001         //Set user-defined private services
SR,00000000         //Set the module as peripheral
PZ                  //Clear Previous Services
PS,1234567890ABCDEF1234567890ABCDEF      //Set 128-bit UUID for the services
PC,1234567890ABCDEF1234567890ABCDEF,02,02        //Set the Characteristic of the
                                                 //service (Read, 2 Bytes long)
LS                  //List of Services
R,1                 //Reboot
WC                  //Clear the previous Script
WW                  //Enter Script mode
@PW_ON              //At power-on event
A                   //Advertise
@CONN               //At connection event
SM,1,00100000       //Set Timer for 1 second
```

```
@TMR1          //At Timer1 Event
$VAR1=@I,2 //Read analog Input from Channel 2 and store the value in the variable
SHW,000B,$VAR1    //Move the value to the User-defined attribute
A
SM,1,00100000          //Reload the timer
@DISCON                //At Disconnect Event
K                      //Disconnect from the master
```

Press Esc key to Exit from the script mode. To execute the script which is stored in the module enter the following in the Hyper Terminal.

```
LW                     //READ THE SCRIPT
SR,01000000            //RUN THE SCRIPT
R,1                    //REBOOT
```

2. Program to read the ADC readings and to control the LED's in the module through phone

```
+                      //Echo ON
SF,1                   //Factory Reset
SS,00000001            //Set user-defined services
SR,00000000            //Set the module as peripheral
PZ                     //Clear Previous Services
PS,1234567890ABCDEF1234567890ABCDEF      //Set 128-bit UUID
PC,1234567890ABCDEF1234567890ABCDEF,0A,02 //Set the characteristics of the services
                                          //Read and Write, 2 Bytes long
LS                     //List the services
R,1                    //Reboot
WC                     //Clear the previous script
WW                     //Enter Script mode
@PW_ON                 //At power-on event
A                      //Advertise
@CONN                  //At connection event
SM,1,00100000          //Load the Timer for 1 second
@TMR1                  //At Timer1 event
$VAR1=@I,2             //Read analog input from channel 2 and store it in VAR1
SHW,000B,$VAR1         //Move the value to the user-defined attribute
|O,07,%000B            //Read the value of the handle and move it to PIO where LEDs are
                       //connected
A                      //Advertise
SM,1,00100000          //Reload the Timer for 1 second
@DISCON                //At disconnect event
K                      //Disconnect from the master
```

Press Esc to exit the script mode. To run the script, enter the following code

```
LW                     //Read the script
SR,01000000            //Run the script
R,1                    //Reboot
```