Getting Started with Bluetooth Low Energy

Anybus Wireless Bolt / Anybus Wireless Bridge II

APPLICATION NOTE

SCM-1202-115 EN 1.0 ENGLISH

HMS Connecting Devices™

1 Introduction

In this application note, we will demonstrate how to communicate with a TICKR X heart rate monitor using an Anybus Wireless Bolt and the Bluetooth Low Energy protocol. (This application note works equally well using an Anybus Wireless Bridge II instead of the Anybus Wireless Bolt).

If the heart rate monitor does not get any input (e.g., heart beats) it will disconnect. For simplicity, wear the heart rate monitor while working with this application note.

If you need more information, the following links will provide helpful guidance:

- Anybus Support

  More specifically, browse your way to the AT Commands Reference. This reference guide is available under the documents and files section for the Anybus Wireless Bolt and the Anybus Wireless Bridge II.

- Bluetooth GATT Specifications

2 Step by Step Guide

1. We will communicate with the Anybus Wireless Bolt using AT commands. To be able to do this, we need a terminal software. We will use PuTTY. Download this software and install it.

   PuTTY download link.

2. Start PuTTY and configure it for a RAW connection to the IP address for the Anybus Wireless Bolt. The default IP address for the Wireless Bolt is 192.168.0.99. Use port 8080.

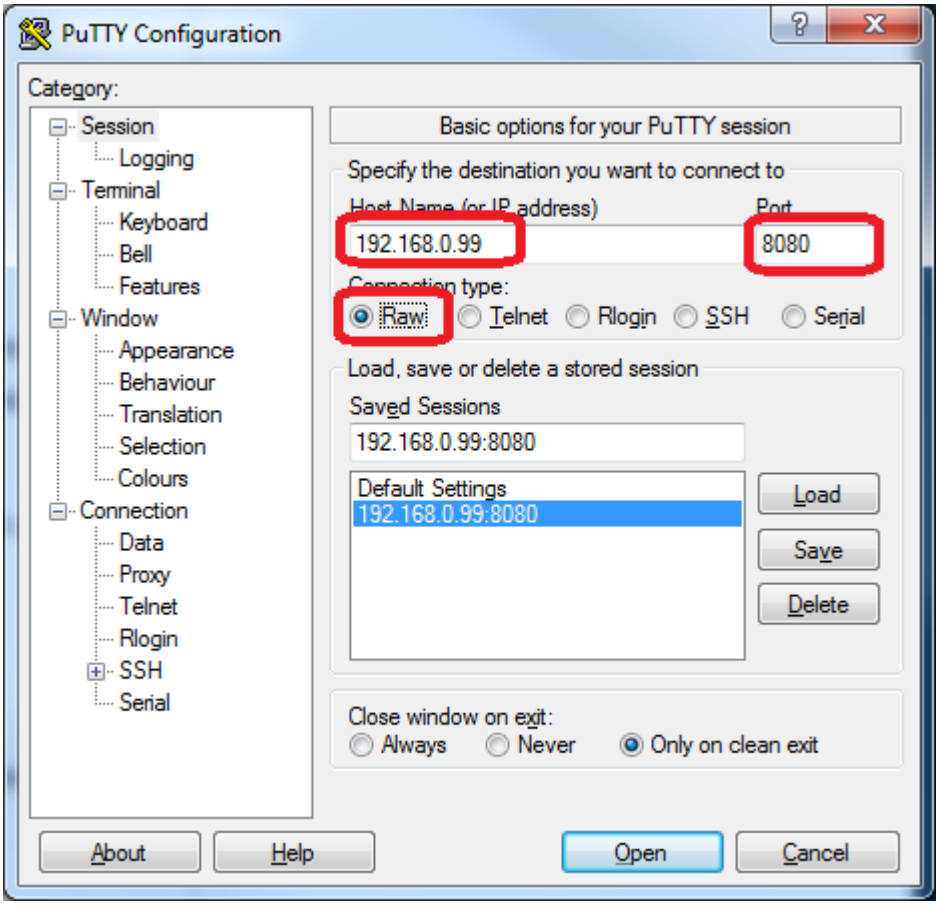


Fig. 1 Putty configuration

   Press **Open** to start the session.

3. Set the Anybus Wireless Bolt to factory default settings by typing in the following command in the PuTTY session window and then pressing **enter**:
   AT&F

4. Set the device to Bluetooth Low Energy mode:
   AT*BLEOM=1,1

5. Reboot the device for the changes to take effect:
   AT*AMreboot

   You need to restart the PuTTY session after the reboot. Left click on the top left corner of the PuTTY window and select **Restart Session**.

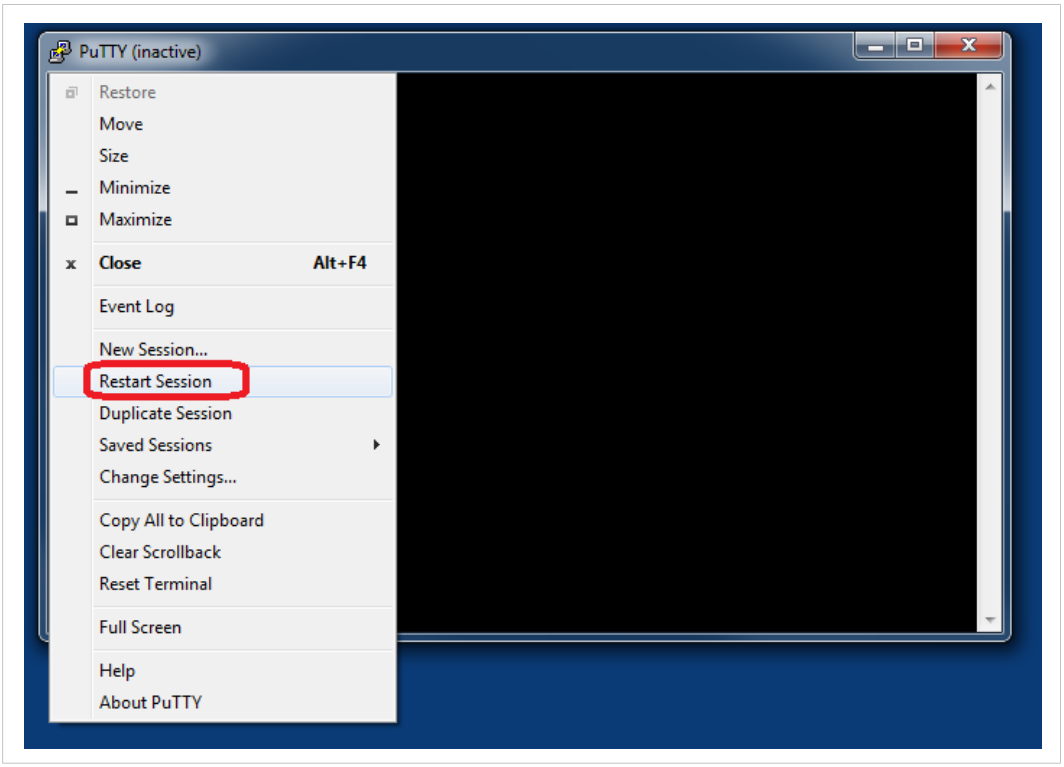


Fig. 2 Restart PuTTY session

6. To be able to receive unsolicited events from the heart rate sensor, the device must be set to Raw Bluetooth Low Energy Mode:
   AT*BLERM=1

7. To find your device, scan the surroundings using this command:
   AT*BLEDD=0,5000,0
   You will receive a list (for example like below) of Bluetooth Low Energy devices nearby.

   *BLEDD:6F-EF-33-CF-A0-B5r,-81,,
   *BLEDD:FF-E3-36-32-35-F1r,-85,,11078BEB9F0F50F1FA97B34A7D0A04EE26A0
   *BLEDD:FF-02-3C-18-89-8Fr,-77,"CREATIVE MUVO 2
   BLE",1409435245415449564520 4D55564F203220424C45
   *BLEDD:6F-EF-33-CF-A0-B5r,-80,,
   *BLEDD:FF-02-3C-18-89-8Fr,-76,"CREATIVE MUVO 2
   BLE",1409435245415449564520 4D55564F203220424C45
   *BLEDD:6F-EF-33-CF-A0-B5r,-81,,
   OK

8. Products are identified by their MAC numbers. For the heart rate monitor in this example, the MAC number is “FF-E3-36-32-35-F1r”.

   To find the name of a device, and to make sure the MAC number matches the right product, make a name discovery (using the MAC number):
   AT*BLEND=FF-E3-36-32-35-F1r

   *BLEND:"TICKR X 9630"
   OK

9. Great! We found the heart rate monitor. Let's connect to it:
   AT*BLEC=FF-E3-36-32-35-F1r
   OK

   *BLEC:0,FF-E3-36-32-35-F1r

   The zero (0) in the beginning is the connection handle. It is possible to connect to several devices, where each connection gets its own separate handle.

10. Now we want to find all services supported by the heart rate monitor. We use the connection handle to communicate directly with the heart rate monitor:

    AT*BGCPSD=0

    *BGCPSD:1,7,1800
    *BGCPSD:8,11,1801
    *BGCPSD:12,15,180f
    *BGCPSD:16,22,180a
    *BGCPSD:23,29,a026ee01-0a7d-4ab3-97fa-f1500f9feb8b
    *BGCPSD:23,29,a026ee01-0a7d-4ab3-97fa-f1500f9feb8b
    *BGCPSD:30,33,a026ee03-0a7d-4ab3-97fa-f1500f9feb8b
    *BGCPSD:34,45,180d
    *BGCPSD:46,51,1814
    *BGCPSD:52,57,1816
    *BGCPSD:58,65535,a026ee04-0a7d-4ab3-97fa-f1500f9feb8b
    OK

    The response lists all available services in the heart rate monitor. The services in the 1800-1900 range are GATT services . Read more about GATT and GATT services here. The other services listed in the response are manufacturer specific.

11. We are specifically interested in GATT service 180D, which is in the list. This service exposes heart rate from a Heart Rate Sensor.

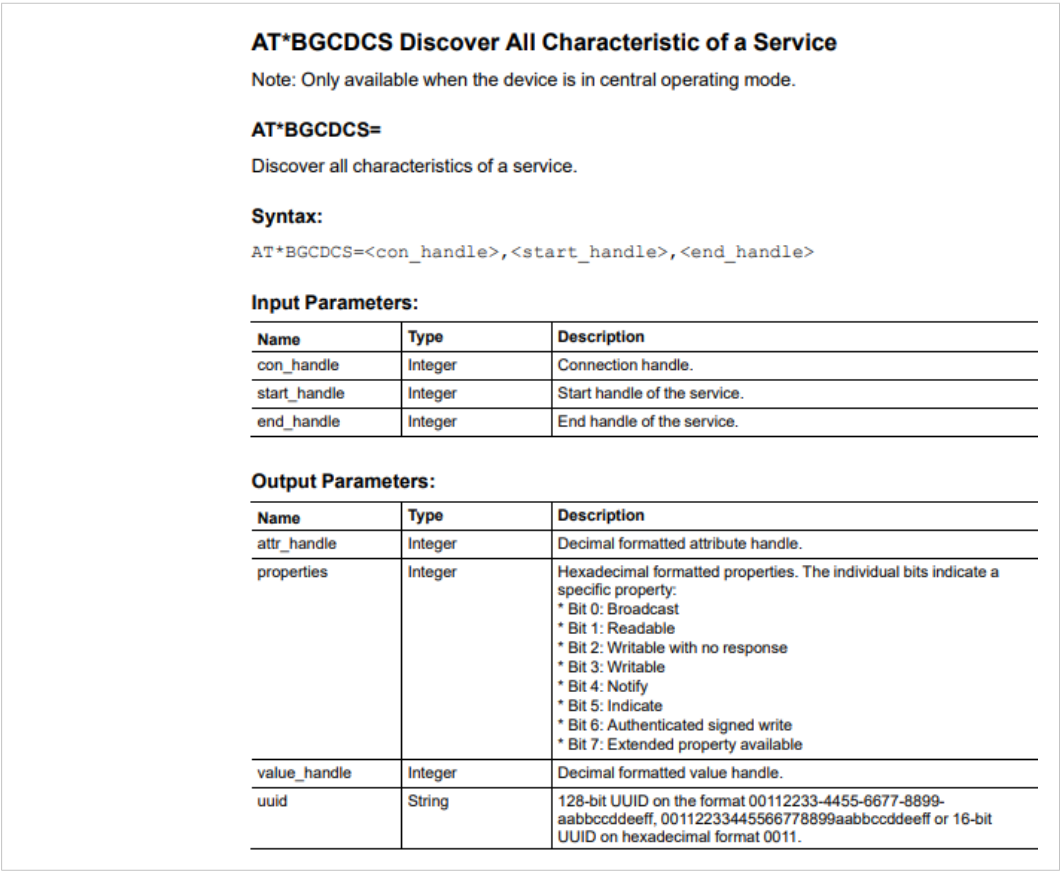
    The next step is to find all characteristics of the 180D service. This means that we want to find out what data we can get from the device. In the table above, the 180D service has start handle 34 and end handle 45, so we search in that area.

    AT*BGCDCS=0,34,45

    *BGCDCS:35,10,36,2a37
    *BGCDCS:38,02,39,2a38
    *BGCDCS:40,10,41,a026e010-0a7d-4ab3-97fa-f1500f9feb8b
    *BGCDCS:43,14,44,a026e011-0a7d-4ab3-97fa-f1500f9feb8b
    OK

    All 16 bit attributes (e.g., 2a37 and 2a38) are GATT attributes and documented on bluetooth.com. The 128 bit characteristics are manufacturer dependent.

    Attribute 2a37 represents heart rate, which is what we are looking for. To see what we can do with this attribute, we need to interpret the response in greater detail. Here is a screenshot from the AT Command Reference Manual:



AT*BGCDCS Discover All Characteristic of a Service

Note: Only available when the device is in central operating mode.

AT*BGCDCS=

Discover all characteristics of a service.

Syntax:

`AT*BGCDCS=<con_handle>,<start_handle>,<end_handle>`

Input Parameters:

Name	Type	Description
con_handle	Integer	Connection handle.
start_handle	Integer	Start handle of the service.
end_handle	Integer	End handle of the service.

Output Parameters:

Name	Type	Description
attr_handle	Integer	Decimal formatted attribute handle.
properties	Integer	Hexadecimal formatted properties. The individual bits indicate a specific property: * Bit 0: Broadcast * Bit 1: Readable * Bit 2: Writable with no response * Bit 3: Writable * Bit 4: Notify * Bit 5: Indicate * Bit 6: Authenticated signed write * Bit 7: Extended property available
value_handle	Integer	Decimal formatted value handle.
uuid	String	128-bit UUID on the format 00112233-4455-6677-8899-aabbccddeeff, 00112233445566778899aabbccddeeff or 16-bit UUID on hexadecimal format 0011.

Fig. 3 BGCDCS AT command details

According to this, the hexadecimal value 10 (10000 in binary) in the response means that attribute 2a37 has a **notify** property. No other properties are available, so this is the property we must use to get readings from the heart rate monitor.

12. To proceed, we need to somehow subscribe to notifications from the **notify** property. GATT properties has associated characteristics. To find these we need to dig deeper:

    AT*BGCDCD=0,35,38

    *BGCDCD:35,36,2a37
    *BGCDCD:35,37,2902
    OK

    The 16 bit identifier 2902 is a GATT characteristic descriptor. It is used to configure the property it is associated with. In this case the property is 2a37. According to the GATT specification, bit 0 of descriptor 2902 enables/disables notifications and bit 1 enables/disables indications.

13. To enable notifications, we will write a 1 to bit 0 of 2902.

    AT*BGCWC=0,37,0100 (byte swapped word)

14. Voila! We are now receiving values from the heart rate monitor, as unsolicited events.

    *BLENR:0,36,1648D303
    *BLENR:0,36,1647CC03
    *BLENR:0,36,16479203
    *BLENR:0,36,1642A703
    *BLENR:0,36,1641CF03

    Byte 2 represents the pulse value. The value is in hexadecimal, so 42 equals 66 bpm.

3 Additional Examples

In the above guide we used the **notify** service to receive values from the heart rate monitor. Let's extend the example by using another service to read the battery level from the monitor.

1. There is a GATT Battery Service, and it has identifier 180F. Let's find the handles:

   AT*BGCPSDU=0,180f

   *BGCPSDU:12,15
   OK

   The start and end handles are 12 and 15, respectively.

2. To find all characteristics associated with this service:

   AT*BGCDCS=0,13,15

   *BGCDCS:13,12,14,2a19
   OK

   2A19 represents a GATT characteristic called Battery Level. The property value is 12 in hexadecimal, which is 10010 in binary. Bit 1 is set, which means that this characteristic supports the **readable** service.

3. This means that we can read the Battery Level value directly (note that this value will vary from time to time):

   AT*BGCRC=0,14,0

   *BGCRC:5F
   OK

   The hexadecimal value 5F is 95 in decimal. 95%! Plenty of charge left on that battery!

Box 4126
300 04 Halmstad, Sweden

info@hms.se

SCM-1202-115 EN 1.0 / 2018-11-01 / 10404