

BLUETOOTH LOW ENERGY (BLE) GATT EXAMPLES WITH OBS421 & OLP425 APPLICATION NOTE

Document Revision

Document number: 23035914

Release: Mar 14, 2014 09:57

Document version: 3

Copyright © 2016 u-blox AG. The contents of this document can be changed by u-blox AG without prior notice and do not constitute any binding undertakings from u-blox AG. u-blox AG is not responsible under any circumstances for direct, indirect, unexpected damage or consequent damage

that is caused by this document. All rights reserved. All brand and product names are trademarks or service marks of their respective owners.

1 Abstract

This document will help you get familiar with the GATT commands available in AT mode of the OBS421 by walking through a few sample use cases.



cB-OBS421

2 Table of Content

- [Abstract](#)
- [Table of Content](#)
- [Related Documents & Downloads](#)
- [Related Products](#)
- [OBS421 Configuration](#)
- [Examples](#)
 - [1\) connectBlue OLP425i-14](#)
 - [2\) Bluetooth Low Energy \(BLE\) Polar Heart Rate Monitor](#)
 - [3\) TI SensorTag \(CC2541 based\)](#)

3 Related Documents & Downloads

[Bluetooth Serial Port Adapter AT Commands GATT](#)

[cB-OLP425 Development Kit Getting started](#)

[cB-2138 Serial Port Adapter Toolbox](#)

4 Related Products

[OBS421i or x](#) with Dual Mode [firmware 5.1.1](#) or newer mounted on a [ACC-34](#)

[OLP425i or x](#) with firmware 1.1.0 or newer

[Polar BLE Heart Rate Strap](#)

[TI SensorTag](#)

5 OBS421 Configuration

Starting from a factory default OBS421, enable "Bluetooth Low Energy - Central Role" with the command:

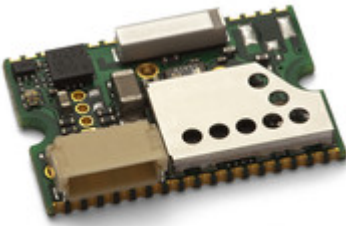
[AT*AGLE=1,1](#) (reset required)

Once the module is configured the following examples will be done with the module in AT Mode.

The Toolbox "User Defined" is the best way to send the commands.

6 Examples

6.1 1) connectBlue OLP425i-14



This Example will show you how to:

- a) Turn On and Off both Red and Green LED's
- b) Single read from the Temperature Sensor
- c) Enable notifications for the Temperature Sensor
- d) Single read from the Battery Level Service
- e) Enable notifications for the 3 axis Accelerometer

Using the GATT commands of an OBS421i-16 and a connectBlue OLP425i-14

6.1.0.0.1 Instructions

-Scan for available unconnected Bluetooth low energy (BLE) devices with the command [AT*AGI=6,4,0](#) to return a list of available devices:

```
AT*AGI=6,4,0
*AGI:78c5e56c8979,0,0,"",53
*AGI:78c5e56c8979,0,1,"OLP425-8979",53
OK
```

-Open a ACL connection to the OLP425 (or Bluetooth low energy (BLE) Peripheral device) with the command [AT*ADCP=78c5e56c8979,17,0,0](#) but replace the Bluetooth address with the address from your OLP module

```
AT*ADCP=78c5e56c8979,17,0,0
*ADCP:0
OK
```

The value the ADCP commands returns is the <ACL Connection Handle> if you have only 1 connection it will always be "0", all the following examples will be using a value of "0"

-Discover Primary Services of the connected device using the command [AT*GATTD=0](#) where 0 is the connection handle reported back when the connection was opened with the AT*ADCP command above.

```
AT*GATTD=0
*GATTD:0,1,11,1800
*GATTD:0,12,15,1801
*GATTD:0,16,26,180A
*GATTD:0,27,31,180F
*GATTD:0,32,50,FFA0
*GATTD:0,51,54,FFE0
*GATTD:0,55,59,FFD0
*GATTD:0,60,65535,2456E1B926E28F83E744F34F01E9D701
OK
```

This list shows the Handle Range and the UUID for each Service.

Most Bluetooth low energy (BLE) devices will have the UUID information documented. For example the [OLP425 Getting Started](#) has the following tables:

```
LED Service (Service UUID: 0xFFD0)
```

The LED service is used to control the red and green LEDs of the cB-OLP425.

Characteristic	UUID	Type	Prop	Description
Red LED	0xFFD1	byte	Read Write	0x00: LED off (default) 0x01: LED on
Green LED	0xFFD2	byte	Read Write	0x00: LED off (default) 0x01: LED on

6.1.0.2

Temperature Service (Service UUID: 0xFFE0)

The temperature service is used to read the temperature sensor on the cB-OLP425. Notifications or indications can be enabled.

Characteristic	UUID	Type	Prop	Description
Temperature	0xFFE1	int8	Read Indication Notification	Temperature in Celsius. If enabled indications and notifications will be sent when the temperature changes.

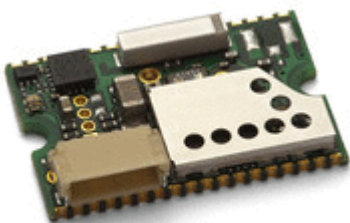
6.1.0.3

Accelerometer Service (Service UUID: 0xFFA0)

The accelerometer service is used to monitor the status of the accelerometer on the cB-OLP425.

Characteristic	UUID	Type		Description
Range	0xFFA2	byte	Read	The range of the accelerometer. 20 = 2G (default) This is a read only attribute.
X	0xFFA3	int8	Indication Notification	Acceleration in X-direction. -2G to 2G. If enable notifications or indications and will be sent when the acceleration is changed.
Y	0xFFA4	int8	Indication Notification	Acceleration in Y-direction. -2G to 2G. If enable notifications or indications and will be sent when the acceleration is changed.
Z	0xFFA5	int8	Indication Notification	Acceleration in Z-direction. -2G to 2G. If enable notifications or indications and will be sent when the acceleration is changed

6.1.0.4 Example 1a - Turn on/off LEDs



From OLP425 documentation, or the table above we know that the LED's are UUID's 0xFFD1 and 0xFFD2. Those will be found as Characteristics of the Primary Service 0xFFD0 as seen in our above scan
 "**GATTDP:0,55,59,FFD0"

-Discover the Characteristics of the LED service by filtering the range to 55 to 59 with the command [AT*GATTDCS=0,55,59](#) where 0 is the connection handle and 55 & 59 are the start and end ranges to display Characteristics.

```
AT*GATTDCS=0,55,59
*GATTDCS:0,56,0A,57,FFD1
*GATTDCS:0,58,0A,59,FFD2
OK
```

This list shows the Characteristic Handle for each of the UUIDs.

-To turn ON the RED LED we will write a Characteristic that doesn't require a response with the command [AT*GATTWN=0,57,01](#) where 0 is the connection handle, 57 is the handle of the characteristic value, and 01 is the hex value to write. 01 for On, 00 for Off in the OLP425 documentation.

```
AT*GATTWN=0,57,01
OK
```

<RED LED will be lit>

```
AT*GATTWN=0,57,00
OK
```

<RED LED will be off>

To then change to the Green LED we simply change the Characteristic value handle from 57(red) to 59(green)

```
AT*GATTWN=0,59,01
OK
```

<GREEN LED will be lit>

```
AT*GATTWN=0,59,00
OK
```

<GREEN LED will be off>

6.1.0.0.5 Example 1b - Read Temperature Sensor

From OLP425 documentation, or the table above we know that the Temperature Sensor UUID is 0xFFE1. That will be found as Characteristics of the Primary Service 0xFFE0 as seen in our above scan
 "**GATTDP:0,51,54,FFE0".

-Discover the Characteristics of the Temperature service by filtering the range to 51 to 54 with the command with the [AT*GATTDCS=0,51,54](#) where 0 is the connection handle and 51 & 54 are the start and end ranges to display Characteristics.

```
AT*GATTDCS=0,51,54
*GATTDCS:0,52,12,53,FFE1
OK
```

This list shows the Characteristic Handle for the only UUID FFE1 to be 53.

-To Read the Characteristic use the Read Characteristic command [AT*GATTR=0,53](#) where 0 is the connection handle and 53 is the Characteristic Handle.

```
AT*GATTR=0,53
*GATTR:0,53,17
OK
```

***An alternate to the AT*GATTR command that requires the Characteristic Handle, is to specify the UUID you

want to read in a specified range.

Using the Read Characteristic by UUID command [AT*GATTRU=0,51,54,FFE1](#) where 0 is the connection handle, and 51 & 54 is the range to scan, and FFE1 is the UUID of the temp sensor we are attempting to read.

```
AT*GATTRU=0,51,54,FFE1
*GATTRU:0,53,17
OK
```

This output displays the hex value 17 -> decimal value 23, current temp is 23C (73F)

6.1.0.0.6 Example 1c - Enable Temperature Sensor Notifications.

Since we have just read the Characteristic Handle "53" and since we know the Characteristic can support Read and Notifications we need to locate the Characteristic Handle. Since the OLP425 document didn't list the order I was able to find the configuration bit to be the next bit "54" it in the source code for the module. It is common practice to have the data bit followed by the configuration bit.

-To Write the Characteristic Configuration we will use the Write Characteristic Configuration command [AT*GATTWC=0,54,01](#) where 0 is the Connection Handle, 54 is the Characteristic Handle, and 01 is the configuration integer to be written.

```
AT*GATTWC=0,54,01
OK
```

```
*GATTN:0,53,1A
*GATTN:0,53,19
*GATTN:0,53,18
*GATTN:0,53,17
*GATTN:0,53,16
*GATTN:0,53,17
*GATTN:0,53,16
*GATTN:0,53,17
```

The notifications were sent based on change of temp, the output is quiet unless there is a change in temperature.

6.1.0.0.7 Example 1d - Read Battery Level

-Find desired UUID from device documentation or Bluetooth SIG listings if standard service.

Battery Service = 180F

https://developer.bluetooth.org/gatt/services/Pages/ServiceViewer.aspx?u=org.bluetooth.service.battery_service.xml

-Discover the Characteristics of the Battery Service by filtering the range to 27 to 31 with the command [AT*GATTDCS=0,27,31](#).

```
AT*GATTDCS=0,27,31
*GATTDCS:0,28,12,29,2A19
OK
```

From online documentation characteristic value starting at 29 is 2A19 "Battery Level Characteristic".

-To Read the Characteristic use the Read Characteristic command [AT*GATTR=0,29](#).

```
AT*GATTR=0,29
*GATTR:0,29,42
OK
```

This output displays the hex value 42 -> decimal value 66, Battery Level is at 66%.

6.1.0.0.8 Example 1e - Enable Accelerometer Sensor Notifications

From OLP425 documentation, or the table above we know that the Accelerometer Sensor UUID's starts at FFA2. That will be found as Characteristics of the Primary Service 0xFFA0 as seen in our above scan [*GATTDG:0,32,50,FFA0](#)".

-Discover the Characteristics of the Temperature service by filtering the range to 32 to 50 with the command with the [AT*GATTDCS=0,32,50](#) where 0 is the connection handle and 51 & 54 are the start and end ranges to display Characteristics.

```
AT*GATTDCS=0,32,50
*GATTDCS:0,33,0A,34,FFA1
*GATTDCS:0,36,02,37,FFA2
*GATTDCS:0,39,12,40,FFA3
*GATTDCS:0,43,12,44,FFA4
*GATTDCS:0,47,12,48,FFA5
OK
```

We now know that 40 is Characteristic Handle for FFA3, the X-Axis, 44 the Y-Axis, 48 the Z-Axis.

To Read the Characteristic of the X-Axis once use the Read Characteristic command [AT*GATTR=0,40](#) where 0 is the connection handle and 40 is the Characteristic Handle.

```
AT*GATTR=0,40
*GATTR:0,40,FD
OK
```

For an accelerometer it would be better to get notifications, so we enable the next bit (41) for X-Axis notifications.

```
AT*GATTWC=0,41,01
OK

*GATTN:0,40,2F
*GATTN:0,40,29
*GATTN:0,40,18
*GATTN:0,40,11
*GATTN:0,40,03
*GATTN:0,40,0A
*GATTN:0,40,FD
```

We can enable more notifications to view all 3 axis:

```
AT*GATTWC=0,45,01
OK
AT*GATTWC=0,49,01
OK

*GATTN:0,40,0A
*GATTN:0,44,F3
*GATTN:0,40,FD
*GATTN:0,44,FC
*GATTN:0,40,04
*GATTN:0,40,17
*GATTN:0,44,02
*GATTN:0,48,4B
*GATTN:0,40,26
*GATTN:0,44,1A
*GATTN:0,48,40
*GATTN:0,44,04
*GATTN:0,48,25
*GATTN:0,40,3F
*GATTN:0,44,F1
*GATTN:0,48,1A
*GATTN:0,40,29
*GATTN:0,44,EB
*GATTN:0,48,0E
*GATTN:0,40,36
```

If you want to close the connection clean use the Close Connection, where 0 is the Connection Handle.

```
AT*ADCC=0,1
```

6.2 2) Bluetooth Low Energy (BLE) Polar Heart Rate Monitor

This Example will show you quickly how to enable notifications for the Heart Rate Profile over GATT using the GATT commands of an OBS421i-16 and a Polar H7 BLE Heart Rate Strap. The commands are not explained

as I use them, see Example 1 - OLP425 for more detailed steps.

6.2.0.0.1 Example 2 - Enable Heart Rate Notifications

-Put on Heart Rate Monitor

-Scan for Bluetooth low energy (BLE) devices:

```
AT*AGI=6,4,0
*AGI:0022d0262f71,0,0,"",59
*AGI:0022d0262f71,0,1,"Polar H7 262F71",58
```

-Connect to Peripheral:

```
AT*ADCP=0022d0262f71,17,0,0
*ADCP:0
OK
```

-Discover Primary Services:

```
AT*GATTDP=0
*GATTDP:0,1,11,1800
*GATTDP:0,12,14,1801
*GATTDP:0,15,20,180D
*GATTDP:0,21,35,180A
*GATTDP:0,36,38,180F
*GATTDP:0,39,65535,6217FF49AC7B547EEECF016A06970BA9
OK
```

-Find desired UUID from device documentation or Bluetooth SIG listings if standard.

Heartrate Profile = 180D

https://developer.bluetooth.org/gatt/services/Pages/ServiceViewer.aspx?u=org.bluetooth.service.heart_rate.xml

-Find Included Services:

```
AT*GATTDCS=0,15,20
*GATTDCS:0,16,10,17,2A37
*GATTDCS:0,19,02,20,2A38
OK
```

From online documentation characteristic value starting at 17 is 2A37 "Heart Rate Characteristic" (read bit 0, enable bit 1).

-Write the Characteristic to enable notifications:

```
AT*GATTWC=0,18,01
OK
```

-Notifications are sent 1 per second:

```
*GATTN:0,17,164E0D03
*GATTN:0,17,1651F502F002
*GATTN:0,17,1650FF02
*GATTN:0,17,164F0F03
*GATTN:0,17,164B29034703
*GATTN:0,17,164A5403
*GATTN:0,17,16495203
*GATTN:0,17,16493B03
*GATTN:0,17,164E1203F302
*GATTN:0,17,1650EC02
*GATTN:0,17,1651EC02
*GATTN:0,17,1654ED02CB02
*GATTN:0,17,1653EF02
```

When I removed the Heart Rate strap the device powered down, the OBS421 correctly reported back a connection link lost:

```
*ADCCO:0,1,1
```

If you noticed that some of the data is longer than others it is because multiple RR values and optional Energy

Expended values can be included with each notification.

6.3 3) TI SensorTag (CC2541 based)

This Example will show you quickly how to enable notifications and read sensor values on the [TI SensorTag](#) using the GATT commands of an OBS421i-16.

The commands are not explained in detail here as I use them, see the GATT AT Commands Document or the Example 1 - OLP425 for more details.

http://processors.wiki.ti.com/index.php/Bluetooth_SensorTag

This Example will show you how to:

- a) Notifications from Button Inputs
- b) One time IR Temperature Sensor read
- c) Notification from IR Temperature Sensor

6.3.0.1 Directions

-Assemble and Power Up the SensorTag:

-Scan for Bluetooth low energy (BLE) devices:

```
AT*AGI=6,4,0
*AGI:bc6a29ac4549,0,0,"",56
*AGI:bc6a29ac4549,0,1,"SensorTag",56
OK
```

-Connect to Peripheral:

```
AT*ADCP=bc6a29ac4549,17,0,0
*ADCP:0
OK
```

-Discover Primary Services:

```
AT*GATTDP=0
*GATTDP:0,1,11,1800
*GATTDP:0,12,15,1801
*GATTDP:0,16,34,180A
*GATTDP:0,35,42,F000AA0004514000B000000000000000
*GATTDP:0,43,53,F000AA1004514000B000000000000000
*GATTDP:0,54,61,F000AA2004514000B000000000000000
*GATTDP:0,62,72,F000AA3004514000B000000000000000
*GATTDP:0,73,84,F000AA4004514000B000000000000000
*GATTDP:0,85,92,F000AA5004514000B000000000000000
*GATTDP:0,93,97,FFE0
*GATTDP:0,98,104,F000AA6004514000B000000000000000
*GATTDP:0,105,115,F000CCC004514000B000000000000000
*GATTDP:0,116,65535,F000FFC004514000B000000000000000
OK
```

This list shows the Handle Range and the UUID for each Service.

Here is the link that I used to for my testing: _

http://processors.wiki.ti.com/index.php/SensorTag_User_Guide#Gatt_Server

6.3.0.1.1 Example 3a - Notification from "Simple Keys" Button Press

From the TI documentation I found the following table:

Simple Key Service

Type	UUID	Handle	Read/Write	Format
------	------	--------	------------	--------

<Data>	F000FFE1	0x5F	Read Only	Bit 2: side key, Bit 1- right key, Bit 0 –left key
--------	----------	------	-----------	--

-Find Included Services

```
AT+GATTDCS=0,93,97
*GATTDCS:0,94,10,95,FFE1
OK
```

-Write the Configuration Characteristic to enable notifications:

```
AT+GATTWC=0,96,01
OK
```

-Notifications are sent every time the state of a button has changed:

```
*GATTN:0,95,02 (Left Down, Right Up)
*GATTN:0,95,00 (Both Up)
*GATTN:0,95,01 (Left Up, Right Down)
*GATTN:0,95,00
*GATTN:0,95,02
*GATTN:0,95,03 (Both Down)
*GATTN:0,95,01
*GATTN:0,95,00
*GATTN:0,95,02
*GATTN:0,95,03
*GATTN:0,95,00
```

To stop notifications

-Write the Configuration Characteristic to disable notifications:

```
AT+GATTWC=0,96,00
OK
```

6.3.0.1.2 Example 3b - Read IR Temp Sensor

Texas Instruments TMP006 @ U5 Two types of data are obtained from the IR Temperature sensor: object temperature and ambient temperature.

6.3.0.1.3 IR Temperature Sensor

Type	UUID	Handle	Read/Write	Format
<Data>	F000AA01 *	0x25	Read/Notify	ObjLSB ObjMSB AmbLSB AmbMSB (4 bytes)
<Data Notification>		0x26	R/W	2 bytes
<Configuration>	F000AA02 *	0x2A	R/W	1 byte

When the enable command is issued, the sensor starts to perform measurements each second (average over four measurements) and the data is stored in the <Data> each second as well. When the disable command is issued, the sensor is put in stand-by mode. To obtain data OTA either use notifications or read the data directly.

For more information please refer to [TI TMP006 User's Guide](#)

-Find Included Services or read from the Handle List

```
AT+GATTDCS=0,35,42
*GATTDCS:0,36,12,37,F000AA0104514000B0000000000000000
*GATTDCS:0,40,0A,41,F000AA0204514000B0000000000000000
OK
```

-Write the Characteristic for the Configuration to enable the sensor, if not enabled the read will report back all zeros

```
AT+GATTW=0,41,01
OK
```

-Read the Characteristic with a single read.

```
AT+GATTR=0,37
*GATTR:0,37,A1FE7C0B
OK
```

Based on the TI TMP006 documentation the bits need flipped and converted for the proper value lets take the ambient temp 7C 0B -> 0x 0B73 -> 2931 (dec)

```
//-- calculate die temperature [°C] -- m_tmpAmb = (double)((qint16)rawT)/128.0// (From TI Documentation)
```

2931 / 128.0 = **22.89 °C**

6.3.0.1.4 Example 3c - Enable Notifications for IR Temp Sensor

```
AT+GATTWC=0,38,01
OK

*GATTN:0,37,A7FE800B
*GATTN:0,37,A0FE800B
*GATTN:0,37,9FFE800B
*GATTN:0,37,A7FE800B
*GATTN:0,37,A5FE800B
*GATTN:0,37,A3FE800B
*GATTN:0,37,A4FE800B
*GATTN:0,37,A1FE800B
*GATTN:0,37,A1FE800B
```

The notification are sent every second regardless of change in value.

-Close the connection

```
AT+ADCC=0,1
OK
```

These same steps can be applied to other sensors on the SensorTag, or other BLE devices.