BLUETOOTH LOW ENERGY SERIAL PORT ADAPTER - GETTING STARTED

Document Revision
Document number: 10551322
Release: Nov 15, 2013 15:51
Document version: 9

1 Abstract

This document describes how to use the connectBlue *Bluetooth Low Energy Serial Port Adapter* modules. The document does not describe Bluetooth low energy technology in detail and it is required that the user is familiar with the concepts described in the *Bluetooth Serial Port Adapter- Getting started* document (ref[1]).

Not all Bluetooth modules support Bluetooth low energy. Currently module versions OLS425, OLS426 and OBS421 v5.0.1 (and later) support Bluetooth low energy. To get an overview of the differences between Classic Bluetooth and Bluetooth low energy, please review http://www.connectblue.com/technologies/bluetooth-low-energy-technology/.

2 Table of Content

1 Abstract
2 Table of Content
3 Related Documents
4 Bluetooth Low Energy
    4.1 Basic Operation
    4.2 Security
5 Serial Port Adapter Concept and Configuration
    5.1 Basic Configuration
        5.1.1 Client-Central and Server-Peripheral
        5.1.2 Server-Central and Client-Peripheral
    5.2 Security Modes
6 Bluetooth Low Energy Modules
    6.1 OLS425 / OLS426
    6.2 OBS421
7 Basic Connection Setup Configuration
    7.1 Point-to-point OBS421 Client and OLS425/426 Server
    7.2 Multipoint OBS421 Client, OBS421 Server (classic) and OLS425/426 Server
    7.3 iPhone Client, OBS421 (Low Energy) or OLS425/426 Server
8 Optimization OLS425/426
    8.1 Use Case 1 - Peripheral Data; Always Connected
        8.1.1 OLS425/426
        8.1.2 OBS421
    8.2 Use Case 2 - Peripheral Data; Disconnected
        8.2.1 OLS425/426
        8.2.2 OBS421
    8.3 Use Case 3 - Central Data; Disconnected
        8.3.1 OLS425/426
        8.3.2 OBS421
9 AT Commands/Events Summary
    9.1 Standard AT Commands
    9.2 GAP Commands
    9.3 Data Mode Commands
    9.4 Miscellaneous Commands

3 Related Documents

[1] **Bluetooth Serial Port Adapter - Getting Started**: Describes the basic concepts for the Bluetooth Serial Port Adapter. This document is written for the classical Bluetooth Serial Port Adapter but the concepts are the same also for the Bluetooth low energy solutions.

[2] **Bluetooth Serial Port Adapter Security**: Describes the different security modes and gives a short introduction to simple pairing which is introduced in the Bluetooth 2.1 specification.

[3] **Bluetooth Serial Port Adapter AT Commands**: Describes all AT commands, default values for AT commands, and possible constraints for different Serial Port Adapter models

[4] **cB-OLP425 cB-OLS425 cB-OLS426 Electrical Mechanical Data Sheet**: For each module there is an associated Electrical Mechanical Data Sheet describing the electrical and mechanical interface, operating status, antennas solutions, Bluetooth and regulatory information as well as some usage and connection guidelines.

[5] **cB-2138 Serial Port Adapter Toolbox**: A PC program for easy getting started and configuration of the different Serial Port Adapter modules. It supports the most commonly used AT commands. See getting started guide [7] for the toolbox.

[6] **Bluetooth Serial Port Adapter Toolbox - Getting Started**: Describes how to get started with the Toolbox, which is a simple-to-use tool to execute AT commands. The supported AT commands are described in [1].

[7] **cB-2137 Flash Loader**: A Windows PC program to re-program the OBS4xx and OLS42x Serial Port Adapters with new firmware using the serial port of the PC. The OLS425 module requires a custom board where the serial port of the module is accessible and can be connected to the PC.

[8] **connectBlue Low Energy Serial Port Service**: The connectBlue Serial Port Service specification. Useful to implement the serial port service on a different platform.

4 Bluetooth Low Energy

Bluetooth low energy is different from Classic Bluetooth. Even though Bluetooth low energy reuses some Classic Bluetooth components such as radio architecture, part of the protocol, simple pairing and Adaptive Frequency Hopping (AFH), it is a new radio standard. Bluetooth low energy is typically intended for small, low cost, (coin-cell) battery-operated end devices.

Bluetooth low energy single-mode (ie. uses Bluetooth low energy only) devices are also called Bluetooth Smart devices. A dual-mode device, which supports both Classic Bluetooth and Bluetooth low energy, is called a Bluetooth Smart Ready device. Hereafter in this document, the term Classic Bluetooth and Bluetooth low energy will be used.

In Classic Bluetooth, the inquiry and connection procedure uses 32 channels, which means it will take some time to detect and/or setup a connection. During this time a neighboring Wireless LAN network may be disturbed. connectBlue has developed the Low Emission Mode feature for its Classic Bluetooth products in order to avoid this disturbance. For Bluetooth low energy the corresponding inquiry (called advertising) and connection setup uses only three channels (2402, 2426, 2480 MHz). This means it uses less power and has a much faster detection and connection phase (a few milliseconds instead of a number of hundreds milliseconds). Also, the three frequency bands are selected to avoid the standard Wireless LAN channels, which is not the case for the 32 Classic Bluetooth channels.

Once a Bluetooth low energy connection is set up, it uses 37 channels each 2 MHz wide with Adaptive Frequency Hopping (AFH). A Classic Bluetooth connection uses 79 channels each 1 MHz wide. Bluetooth low energy has a lower modulation index, which means it allows for a longer range and more robust connection than a Classic Bluetooth connection. The bandwidth is significantly lower for the Bluetooth low energy link

compared to the Classic Bluetooth link (theoretic limit of 260 kbits/s compared to 2.1 Mbits/s for Classic Bluetooth).

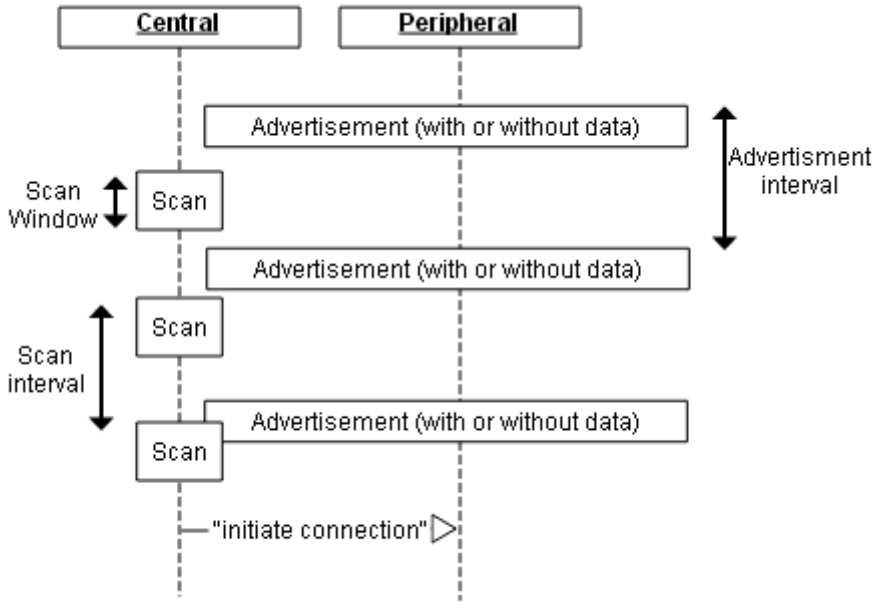
To achieve low power consumption from a Bluetooth low energy module, it is important that the module spends most of its time in low-power mode. A use case where a connection is active and data is continuously transmitted/received does not mean low power consumption. Hence, Bluetooth low energy is really intended for use cases where the module is in low-power mode most of the time and any connection is active only for a short amount of time, which means only limited data transmission is suited.

4.1 Basic Operation

A Bluetooth low energy connection consists of one *central* and one *peripheral* role. There are more roles than the central and peripheral roles, but they will not be covered in this document.

The *peripheral* contains the "data" and advertise*s* it to inform the environment about its existence. An advertise*ment* is a periodic broadcast that can contain data or is just sent out to allow for incoming connections.

A *central scans* its environment and if an advertise*ment* is detected, it is possible for the *central* to initiate a connection. During the *scan*, only limited radio transmission is done and only when a remote device is found. This means it does not disturb the radio environment more than necessary.



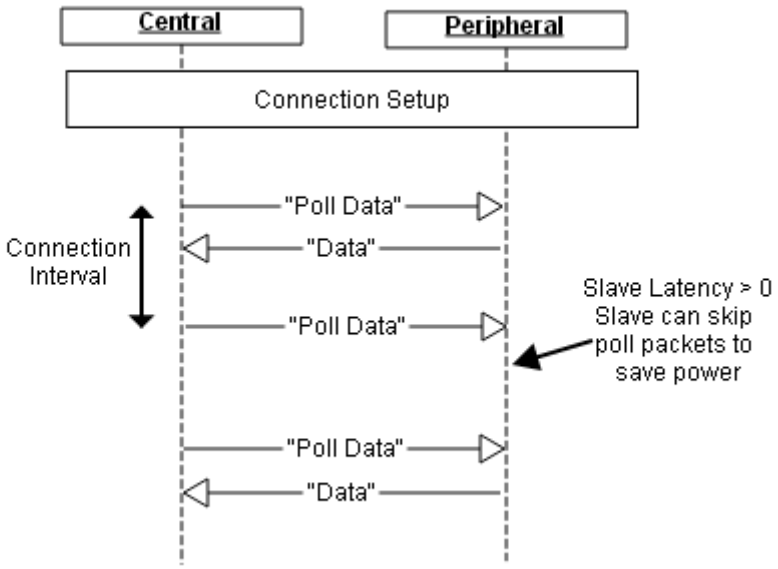
One important advertisement parameter is the advertise*ment interval*. A short interval will allow for faster detection and connections but it also means that the power consumption increases for the peripheral during the advertisement.

For the central, the scan interval and scan window are important parameters that controls how fast the peripheral is found and how active the radio is. Please note that the scan window must always be shorter or equal to the scan interval. The scan duty cycle is the ratio between the scan window and interval. For example, if the scan interval is 100 ms and the scan window is 50 ms, the duty cycle is 50%.

It is also possible for the peripheral to do a directed advertisement, which means that the advertisement is directed to a specific central with a specific Bluetooth address.

If the central sets up a connection to the peripheral, the central will always be the master of the link and the peripheral will be the slave. No master/slave switch is allowed. A central can be connected to more than one peripheral but a peripheral can only be connected to one central. Hence, only star topology is allowed and scatter-nets are not allowed.

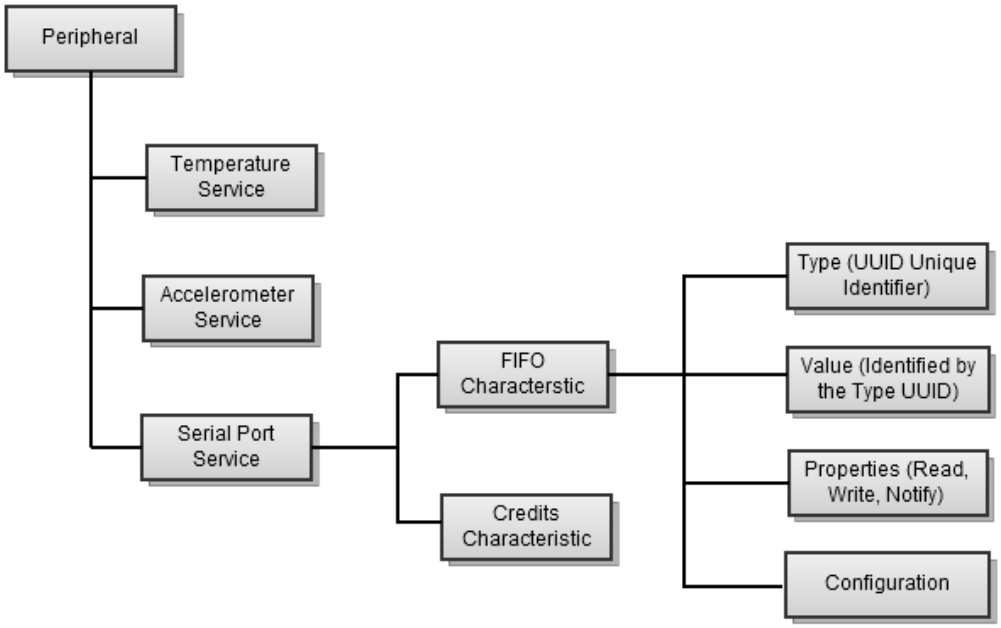
For an ongoing connection, the central (master) polls the peripheral (slave) for data. The connection interval specifies how often this polling is done. A shorter connection interval means lower latency but higher power consumption.



It may be necessary to configure both advertisement and connection intervals to optimize for a specific use case. This may mean compromising between fast connection and response times and low power consumption.

Another way to lower the power consumption on the peripheral side is to specify a *slave latency*, which defines the number of consecutive connection events that may be ignored by the peripheral. This means that when the peripheral doesn't have any data to send it can stay asleep longer before it needs to reply to a connection event. Hence, a way to have short latency and low power consumption for data transmission from peripheral to central.

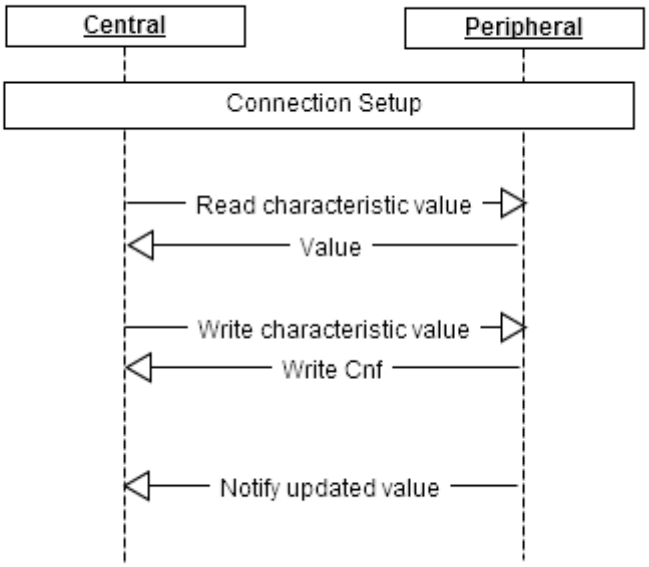
The data of the peripheral is structured in one or more services. There could for example be a battery temperature service, an accelerometer service and/or a serial port service. The Bluetooth SIG defines profiles for services and many standard services can be found at www.bluetooth.org. It is also possible to define proprietary services. Every service contains one or more characteristics. A characteristic consists of a unique identifier (UUID), a value, properties and a configuration. The value (type and size) is defined by the unique identifier. It could e.g. be a string or 16 bit unsigned integer. The properties specifies how the value can be accessed and the configuration is a user defined info often a string describing the characteristic.



There are several ways for a central to access the value of a characteristic.

- Write: The central can write the value of the peripheral
- Read: The central can read the value from the peripheral
- Notify: The peripheral can automatically send updated values to the central. This avoids periodic read operations for the central.

The kind of operation that a peripheral allows for a specific value is decided by the characteristics properties. Hence, it is not always possible to write, read or get notifications.



Actually there are some more operations that can be done. It is possible to make a write with or without a confirm. It is also possible to get an indication, which is basically a notification that requires a confirm. These will not be explained further in this document.

4.2 Security

To the user, security for Bluetooth low energy is similar to secure simple pairing used for Classic Bluetooth. See ref [2] for details.

Bluetooth low energy supports three use cases:

- Just Works
- Passkey Entry
- Out of Band

The *Numeric Comparison* use case in Classic Bluetooth is not supported in Bluetooth low energy. Another difference is that the *Just Works* and *Passkey Entry* use cases supports only very limited protection for passive eavesdropping compared to Classic Bluetooth.

The encryption algorithm used in Bluetooth low energy is AES-CCM cryptography.

Bluetooth low energy also specifies a privacy feature to make it more difficult to track a Bluetooth low energy device by changing its Bluetooth address frequently.

The privacy feature is only used for connection setup and not for discoverability.

5 Serial Port Adapter Concept and Configuration

Classic Bluetooth has a determined set of “profiles” that in essence are application defined behaviors that Bluetooth devices use to communicate with other each other. Bluetooth low energy profiles are different from those used in Classic Bluetooth and are based on the Generic Attribute Profile (GATT). GATT is used for service discovery as well as read/write values on a device. Unlike Classic Bluetooth, product developers can develop their own profiles and services to add to those from the Bluetooth SIG.

The basic operation of the Serial Port Adapter is to transmit application data between the serial port and a Bluetooth connection. Contrary to Classic Bluetooth, there is no Serial Port Profile (SPP) for Bluetooth low energy. Therefore, connectBlue has defined the proprietary "connectBlue Low Energy Serial Port Service" to offer transparent serial communication and simulate the Classic Bluetooth SPP.

Many features and functions will be the same for a Classic Bluetooth module and a Bluetooth low energy module. In this document, only Bluetooth low energy features are covered. For details of general operation of the Serial Port Adapter, see ref [1].

5.1 Basic Configuration

For Bluetooth classic there is a Client and Server profile. The client initiates the connection attempt to the server and the server accepts incoming connections. This is similar for Bluetooth Low Energy except that there are two client and two server roles depending on the Low Energy role (central/peripheral).

- Client - Central
- Server - Peripheral
- Client - Peripheral (implemented using directed advertisement)
- Server - Central (implemented using directed advertisement)

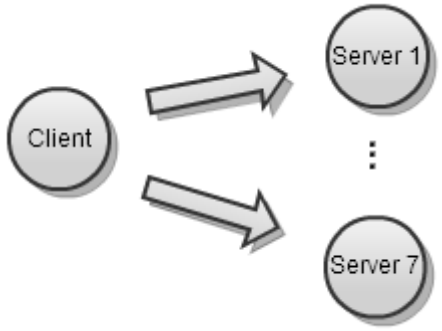
The central always makes the scan (looking for advertisements) and the peripheral always makes the advertisement. The basic idea is that the peripheral is in low-power mode most of the time (to save power) and when it is awakened, it starts advertising. The central is always scanning and when a "correct" peripheral is found, a connection is initiated. Since the scan does involves only limited radio transmission when a remote device is found, it does not disturb other radios in the vicinity more than necessary.

A Client-Central can setup a connection to a Server-Peripheral and the Client-Peripheral can "setup" a connection to a Server-Central. However, two centrals or two peripherals cannot connect to each other.

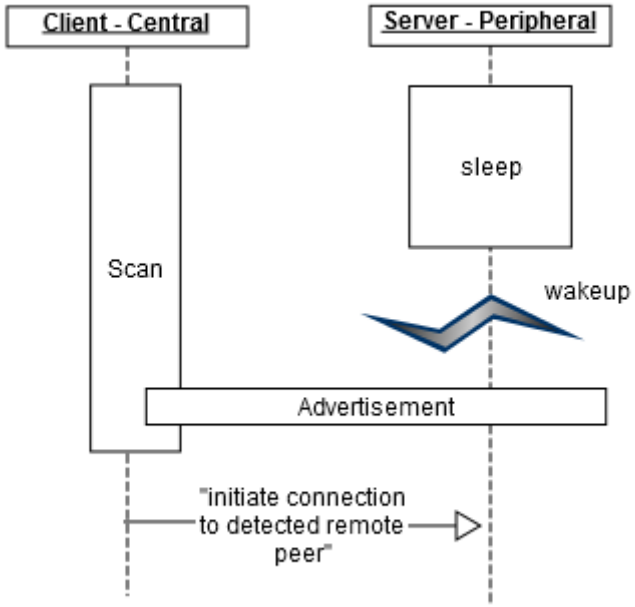
The following sections describes how the Serial Port Adapter concept (client, server, remote peer) is mapped to the Bluetooth low energy concepts (central, peripheral, scan, advertisement).

5.1.1 Client-Central and Server-Peripheral

The more straightforward case is when the client is the central and the peripheral is the server. The client sets up the connection to the server similar to the Classic Bluetooth definition.



At Low Energy level, the Client-Central is scanning for the Server-Peripheral. When the Server-Peripheral starts advertising the Client-Central finds the Server-Peripheral and initiates the connection.



The connection attempt can (for some modules) be initiated with the connect (AT*ADCP) AT command.

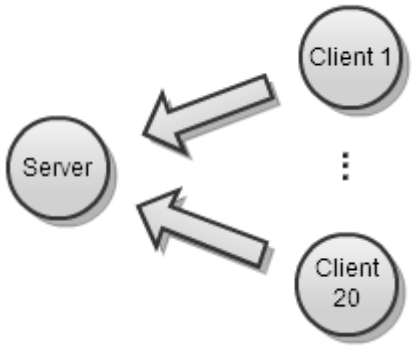
It is also possible to configure the module to automatically setup the connection. For an automatic connection, the default client profile (AT*ADDCP), the write remote peer (AT*ADWDRP) and the number of remote peers (AT*ADNRP) AT commands must be used. The remote peer is the server, a client is configured to setup a connection to and it is normally possible to configure a maximum of 1, 3 or 7 remote peers depending on module and use case.

The server is configured with the default server profile (AT*ADDSP) AT command.

5.1.2 Server-Central and Client-Peripheral

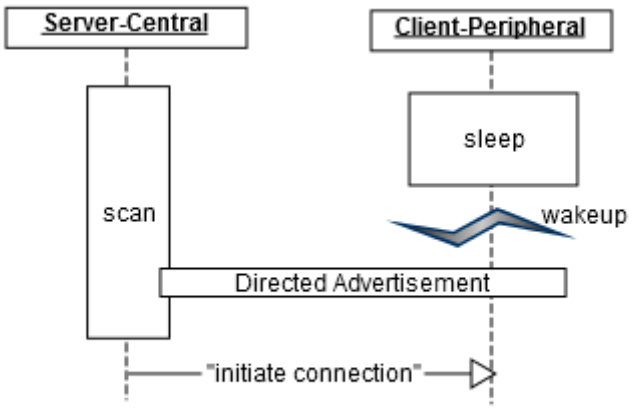
With the Client-Central and Server-Peripheral configuration, the Bluetooth address of the remote peer (server) is configured in the client. This is not the best solution if there are many servers. There is a limit of max number of peers (normally 1, 3 or 7), which may not be enough for some use cases. The Client-Central cannot be configured to automatically connect to more than 7 remote peers (Server-Peripheral).

To handle a use case where a central must communicate with a large number of sensors, the Client-Peripheral and Server-Central roles are added.



The Client-Peripheral then makes a directed advertisement which is targeted to a specific Server-Central. If that specific Server-Central detects the advertisement, it will initiate a connection. Hence, it is still the central that initiate connections but at a Serial Port Adapter concept level, the peripheral will be configured as the client and the central as the server.

It is possible to have any number of Client-Peripherals, which are all configured to connect to the same Server-Central. Of course it does not mean that all of them can be connected at the same time since there is a limit on how many parallel links a module supports. However, many Bluetooth low energy use cases do not need a large number of parallel connections. Instead the Client-Peripheral modules spend almost all time in low-power mode and wake up only periodically to setup a connection and transfer data.



The client module is configured to automatically "setup the connection". The default client profile (AT*ADDCP), the write remote peer (AT*ADWDRP) and the number of remote peers (AT*ADNRP) AT commands must be used for the Client-Peripheral. The remote peer is the server that the client is configured to setup a connection to.

The Server-Central is configured with the default server profile (AT*ADDSP) AT command.

5.2 Security Modes

The Bluetooth Low Energy Serial Port Adapters supports five security modes (see ref [2] for details).

- Security Disabled (mode 1)
- Fixed Pin (mode 3)
- Just Works (mode 4)
- Display Only (mode 5)
- Keyboard Only (mode 7)

The security mode is configured using the AT*AGSM command.

Please note that not all Bluetooth low energy modules will support all of the modes.

For the *Security Disabled* mode, no pairing is needed and the module will accept all connections.

The *Fixed Pin* and *Keyboard Only* modes are similar that a passkey will be used to authenticate. For the *Fixed Pin* mode, the passkey configured by the AT*AGFP2 command will be used and for the *Keyboard Only* mode, the user must enter the passkey with the AT*AGUPE command when the *AGUPE event has been received (same as for Bluetooth classic).

The *Just Works* mode basically just means that pairing is by default disabled. To pair, pairing must be enabled and during this time all pairing attempts will succeed. See later reference to the security mode command (AT*AGSM) for details on how to enable pairing. Please note that if the *Just Works* mode is set, pairing is automatically disabled. This means that even if security is then set another mode, pairing is still disabled.

The *Display Only* mode is used by a device if the other device uses *Keyboard Only*. The display side (e.g. PC) displays a passkey that must be entered on the keyboard side. The host of the display side get the *AGUPD event when it is time to display a passkey.

6 Bluetooth Low Energy Modules

There are currently three connectBlue modules that support Bluetooth low energy. The OBS421 v5.0.1 (and later), which is a dual-mode module, the OLS425 and OLS426, which are both single-mode modules.

6.1 OLS425 / OLS426

The OLS425 is available with an adapter board (cB-0958) to get the same footprint as the OBS421 module. This means it is possible to use any of the existing connectBlue carrier and development boards. An OLS425 mounted on the cB-0958 board is named OLS426. Hence, the OLS425 and OLS426 contains the same firmware but have a different form factor.

cB-OLS426i-04

cB-OLS426x-04

cB-OLS425i-04

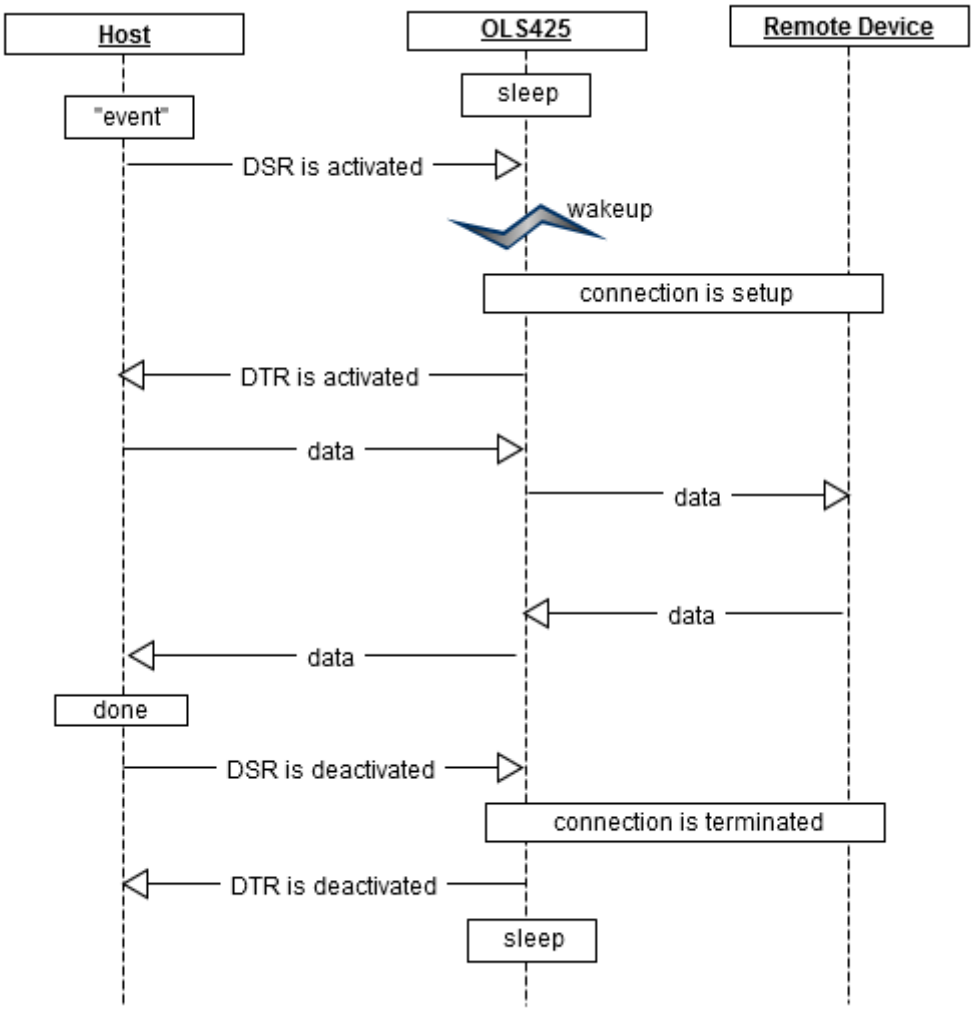
cB-OLS425x-04

Currently the Client-Central and Server-Central roles are not implemented for the OLS425/426. This means it is not possible to setup a connection between two OLS425/426 modules. The OLS425/426 is mainly intended for the low power peripheral side and the OBS421 (or other solution) should be used for the central (remote) side.

6.1.0.1 Operating modes

The OLS425/426 may be configured to suit specific operating requirements in terms of fast connection and response times versus low power consumption. First, you can set the power save mode using AT*AMPM, which decides whether low-power mode is allowed or not. This basically means that under certain conditions, the module will be asleep and the power consumption is very low. Second, the module may be configured with AT*AMDS2 to let the host control the module operation by activating/deactivating the UART DSR pin of the OLS425/426, and AT*AMPSF to achieve a customized behavior. This can be for example to disable advertising when DSR is deactivated in order to lower the power consumption, with the cost of temporarily disabled discoverability and connectability. The UART DTR pin of the OLS425/426 is by default used by the module to indicate when there is an active connection.

One example is to configure the module so that the host will wake the module up by activating DSR, wait for an active connection (DTR is activated), transmit/receive data and when done, disconnect and put the module back in low-power mode by deactivating DSR.



One advantage with Bluetooth low energy and the OLS425/426 is that the described procedure can be done in a very short time (compared to Classic Bluetooth) and that the OLS425/426 uses only a few µA in low-power mode. Hence, it is possible to have a use case where the module can live for a very long time on a coin-cell battery. Of course this assumes that the module is in low-power mode most of the time and that only limited amount of data is transmitted. For more information see Power save optimizations OLS425/426.

In a scenario where a server provides sensor data with fairly low update rate and where the amount of data is not too large, a possible solution is to use customized advertise data. In that way data is transferred from the peripheral to the central without establishing a connection, which in turn means no security. Typically the OLS425/426 is held in AT mode constantly and fed with new advertise data from the host through an AT command, see AT*AGAD and AT*AGSD in ref [3].

6.2 OBS421

Bluetooth low energy is supported by the OBS421 in firmware version 5.0.1 and later.

cB-OBS421i-14

cB-OBS421x-14

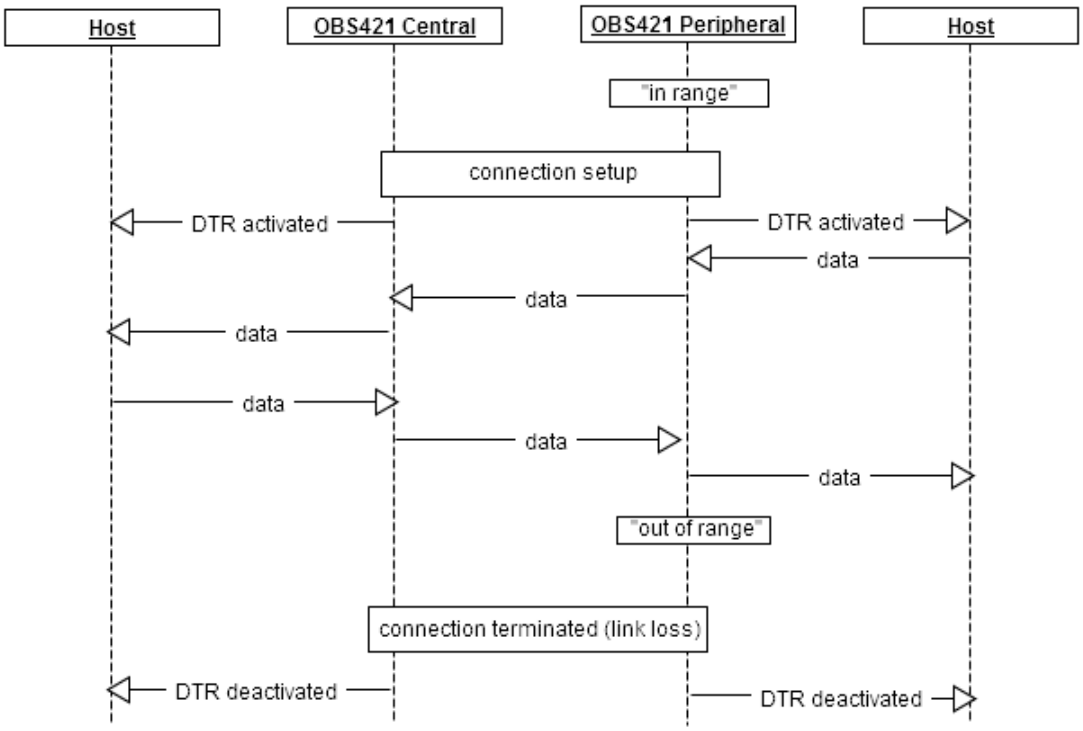
By default, Bluetooth low energy is disabled for the OBS421 module and it then supports up to 7 parallel Classic Bluetooth links. To enable Bluetooth low energy, the AT*AGLE command is used. If Bluetooth low energy is enabled, the following link configurations are possible.

- 3 Classic Bluetooth links and 0 Bluetooth low energy links
- 2 Classic Bluetooth links in parallel with 1 Bluetooth low energy link
- 1 Classic Bluetooth link in parallel with 2 Bluetooth low energy links
- 0 Classic Bluetooth links and 3 Bluetooth low energy links

Hence, it is possible, for the OBS421, to have Classic Bluetooth links and Bluetooth low energy links active in parallel. The host transmits data without caring what kind of link it is (except that the throughput is normally much lower for the low energy link). Also note that for the OBS421 module, the Extended Data Mode works both for Classic Bluetooth and for Bluetooth low energy.

An OBS421 module can setup a Bluetooth low energy connection to another OBS421 module. This is not a typical scenario but there are use cases where it could be interesting. It is more common to have a Bluetooth low energy connection between an OBS421 and an OLS425/426, where the OLS425/426 is the low power side.

In the figure below the DTR pin for both OBS421 modules are configured to show the connection status. The peripheral side (client or server) is moved in range where a connection is setup and data transmitted and then out of range.



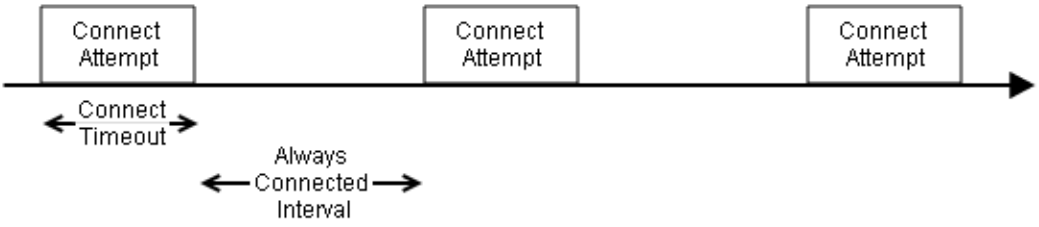
The OBS421 DTR pin is configured to show the connection status. Hence, it can be used like a traditional CD (Carrier Detect).

7 Basic Connection Setup Configuration

It is possible to configure the OBS421 and OLS425/426 to automatically setup a connection between each other. Typically the client and server profile must be configured as well as one or more remote peers for the client side. A client will setup a connection to a server and a server accepts incoming connections. A remote peer is a remote server a client shall setup a connection to.

An OBS421 module can setup multiple and mixed Classic Bluetooth and Bluetooth Low Energy connections.

In the samples, the *always connected* scheme is used when configuring the remote peer. By default, the module will then periodically try to setup a connection for a specific time before giving up.



By default, the connect timeout is 5 seconds and the always connected interval is 10 seconds. The connect timeout and the always connected interval can be configured using the ATS6007 and ATS6010 S-register AT commands. Please note that for Bluetooth Low Energy, the module will not do a scan between the connect attempts. Scanning is only performed during a connect attempt.

7.1 Point-to-point OBS421 Client and OLS425/426 Server

Client device (OBS421) is configured to always attempt to connect to a specific Bluetooth Low Energy server device (OLS425/426).

OBS421 - Client Configuration

AT Command	Description
AT*AGLE=1,1	Enable Bluetooth low energy central role.
AT*ADDSP=255,1	Server profile to none.
AT*ADDCP=14,1	Client profile to serial port for both classic and low energy.
AT*ADNRP=1,1	One remote peer.
AT*ADWDRP=0,112233445566,0x0102,0,"",1	Remote peer with BD address 0x112233445566. Connection scheme is "Always connected" and LE peer bit is set. See AT Command specification for details.

Please note that the module must be reset for the settings to be activated.

OLS425/426 - Server Configuration

Default configuration.

7.2 Multipoint OBS421 Client, OBS421 Server (classic) and OLS425/426 Server

The client (OBS421) sets up a Classic Bluetooth connection to one server (OBS421) and a Bluetooth Low Energy connection to another server (OLS425/426).

OBS421 - Client Configuration

AT Command	Description
AT*AGLE=1,1	Enable Bluetooth low energy central role.
AT*ADWM=2,0,1	Enables two links.
AT*ADDSP=255,1	Server profile to none.
AT*ADDCP=14,1	Client profile to serial port for both classic and low energy.
AT*ADNRP=2,1	Two remote peers.
AT*ADWDRP=0,112233445566,0x02,0,"",1	First remote peer with BD address 0x112233445566. Connection scheme is "Always connected" and LE peer bit is not set (hence classic remote peer).
AT*ADWDRP=1,223344556677,0x0102,0,"",1	Second remote peer with BD address 0x223344556677. Connection scheme is "Always connected" and LE peer bit is set (hence LE remote peer).

Please note that the module must be reset for the settings to be activated.

OBS421 - Server Configuration

Default configuration.

OLS425/426 - Server Configuration Peer

Default configuration.

7.3 iPhone Client, OBS421 (Low Energy) or OLS425/426 Server

A connectBlue demo application for iOS can be downloaded from iTunes App Store: https://itunes.apple.com/app/cb-olp425/id575523395. The demo application cB-OLP425 can be used to search for different low energy peripheral devices, perform connections and display services. This sample describes how the different models are configured to be discovered by the demo application.

OBS421 - Server Configuration

AT Command	Description
AT*AGLE=2,1	Enable Bluetooth low energy peripheral role
AT*ADDSP=14,1	Set the server profile to SPP low energy.

OLS425/426 - Server Configuration

Default configuration.

8 Optimization OLS425/426

This section describes some important parameters to optimize behavior of mainly the OLS425/426. It also describes some common use cases and the configuration to consider. For basic connection setup configuration see the Basic Connection Setup Configuration section.

Regarding optimizing power consumption, the OBS421 has a much higher power consumption than the OLS425/426 and most of the optimization to reduce power consumption refers to the OLS425/425 device.

There are some very important parameters in Bluetooth Low Energy that affect the connection time, response time and power consumption.

- *Advertisement Interval*: Time between two consecutive advertisements for a peripheral.
- *Scan Window and Interval*: The window is the length of the scan and the interval is the time between two consecutive scans. Hence, "window time" is less or equal to "interval time".
- *Connection Interval*: Time between two consecutive poll intervals from central to peripheral.
- *Connection Latency*: When a connection is active, the peripheral must respond to poll packets from the central (even if there is no data to transmit). If the connection latency is >0, the peripheral is allowed to skip poll packets if there is no data to transmit. Hence, even with a short connection interval, it is possible to get a low power consumption for the peripheral.

See section Basic Operation for details. The above parameters are set with the S-Register AT command (ATS). See AT command specification for details.

For a *quick connection setup*, the *advertisement interval* (ATS6001 and ATS6002) must be short for the peripheral and the scan duty cycle (ratio between the scan *interval* and *scan window*) must be high (see AT commands ATS6008 and ATS6009). This increases power consumption for both sides since the radios will be much more active.

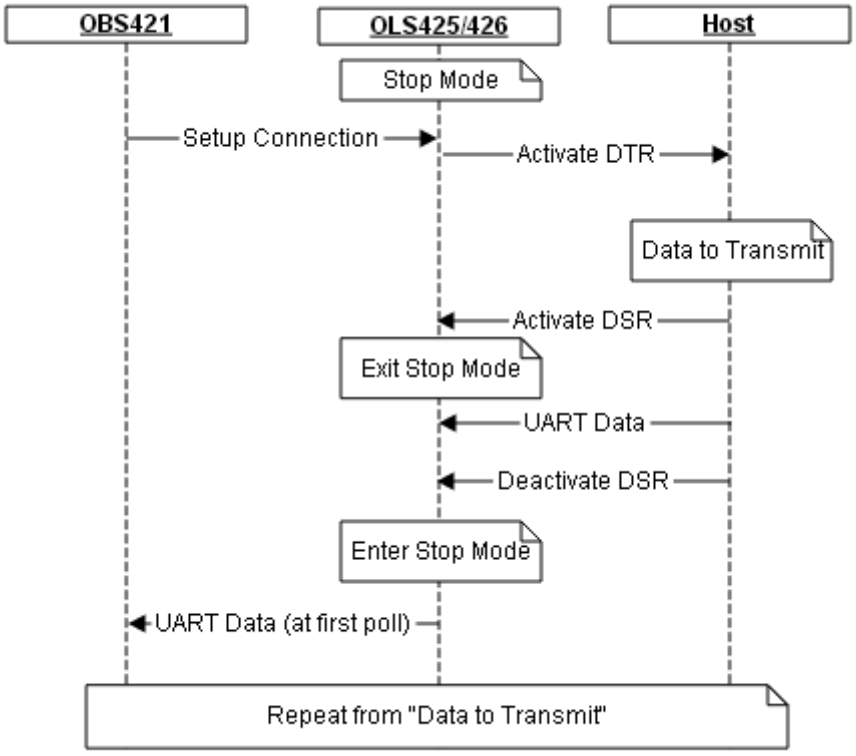
For a *short response time* when a connection is active, the *connection interval* (ATS6004 and ATS6005) must be short, which increases power consumption since the radios will be more active. It is possible to lower the power consumption on the peripheral side setting the central *connection latency*, which means that the peripheral does not need to respond to all poll packets from the central. The *connection latency* (ATS6006) is configured at the central side (OBS421).

To get a low power consumption for the OLS425/426 peripheral, it is important to keep the module in stop mode as much of the time as possible. The OLS425/426 can be in stop mode no matter if the connection is active or not or if the module is advertising or not. However, for the UART to be active, the OLS425/426 must be awake.

Stop mode can be controlled by the host CPU of the OLS425/426 using the DSR pin or it can be controlled internally by the LE connection. See the *AT*AMDS2* and *AT*AMPSF* commands for details.

8.1 Use Case 1 - Peripheral Data; Always Connected

The use case describes a setup where the peripheral (typically sensor side and OLP425/426) transmits data and where the connection is always active. To optimize power consumption for the peripheral side, it is important that it enters stop mode when there is no data to transmit (even if the connection is always active).



The OLS425/426 is configured as a server-peripheral and the OBS421 as a client-central. For basic connection setup, see section Basic Connection Setup Configuration

Please note that it is also possible to transmit data from the OBS421 to the OLS425/426. However, some form of hand-shaking is then required for the OLS425/426 side to know when all data has been transmitted.

8.1.1 OLS425/426

At Command	Description
AT*AMPM=3,1	Stop Mode The module is allowed to enter stop mode.
AT*AMDS2=0,0x03,1	Do not allow stop mode when DSR is activate. Allow sleep mode when DSR is inactive.

This configuration is the default configuration of the OLS425/426 module.

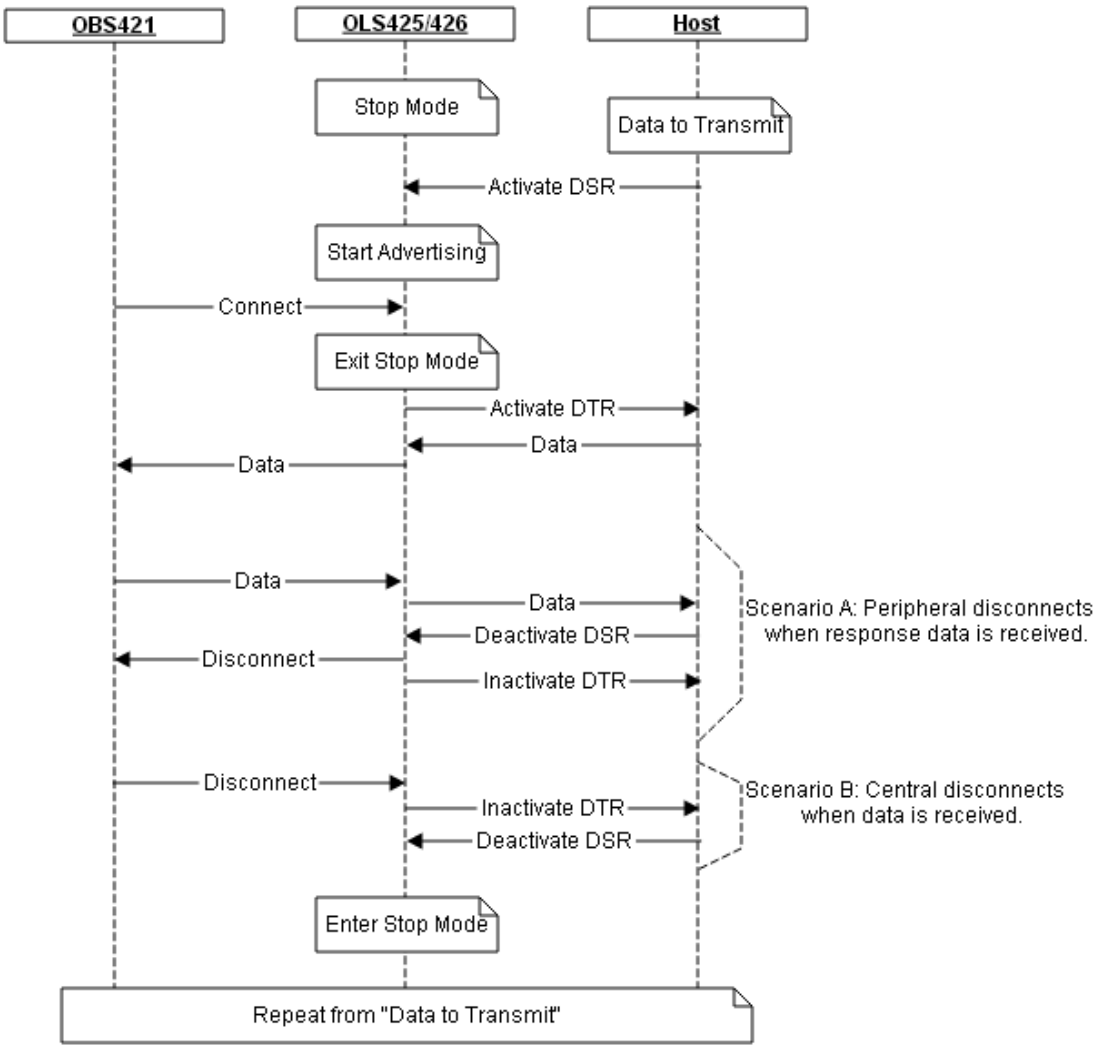
8.1.2 OBS421

At Command	Description
ATS6000=4000	Link loss timeout (Default value 2000ms) Longer connection intervals and slave latency means the link loss timeout needs to be increased.
ATS6004=80	Min connection interval (Default value 6) 80 * 1.25ms = 100ms
ATS6005=80	Max connection interval (Default value 8) 80 * 1.25ms = 100ms
ATS6006=9	Slave latency (Default value 0) Value 9 means that the peripheral must respond to at least every 10 poll packets. With the connection interval of 0.1s the module must respond at least once every second. Hence, if there is data to transmit the latency is 0.1s for the peripheral to transmit data but if there is no data to transmit the peripheral will respond only once a second.

Note that if the connection interval and slave latency means that the slave might sleep for longer than the link supervision timeout (default 2s), it must be increased as well (ATS6000).

8.2 Use Case 2 - Peripheral Data; Disconnected

The use case describes a setup where the central (OBS421) sets up the connection, the peripheral (typically sensor side and OLS425/426) transmits data and where the connection is active only during data transmission. Hence, every time the peripheral has data to transmit, the connection must be setup.



There are two ways to terminate the connection. In scenario A, the central must send a response and when received by the host of the peripheral, it disconnects by deactivating DSR. Another option, scenario B, is for the central to disconnect once the expected data is received.

The OLS425/426 is configured as a server-peripheral and the OBS421 as a client-central. For basic connection setup, see section Basic Connection Setup Configuration

Note that there is a trade-off between low power consumption and short connection setup. The advertisement interval and scan window/interval should be configured to match the requirements of the specific use case.

8.2.1 OLS425/426

At Command	Description
AT*AMPM=3,1	Stop Mode The module is allowed to enter stop mode.
ATS6001=32	Min advertisement interval (Default value is 1600) 32 * 0.625 ms = 20ms
ATS6002=40	Max advertisement interval (Default value is 2000) 40 * 0.625ms = 25ms
Scenario A: AT*AMDS2=0,0x2C,1 Scenario B: AT*AMDS2=0,0x0C,1	Enable connectable and discoverable (advertising active) when DSR is active. Disable connectable and discoverable (no advertisement) when DSR is inactive. Scenario A: When DSR is deactivated, the LE connection is terminated. Hence, 0x**2**C.
AT*AMPSF=0x03,1	Do not allow stop mode when an LE connection is active. Allow stop mode when there is no LE connection.

Note that the power consumption when there is no active connection will be quite high. Hence, the OLS425/426 must have a strategy on how long it will keep DSR active if there is no connection. Some timeout is needed not to drain the battery.

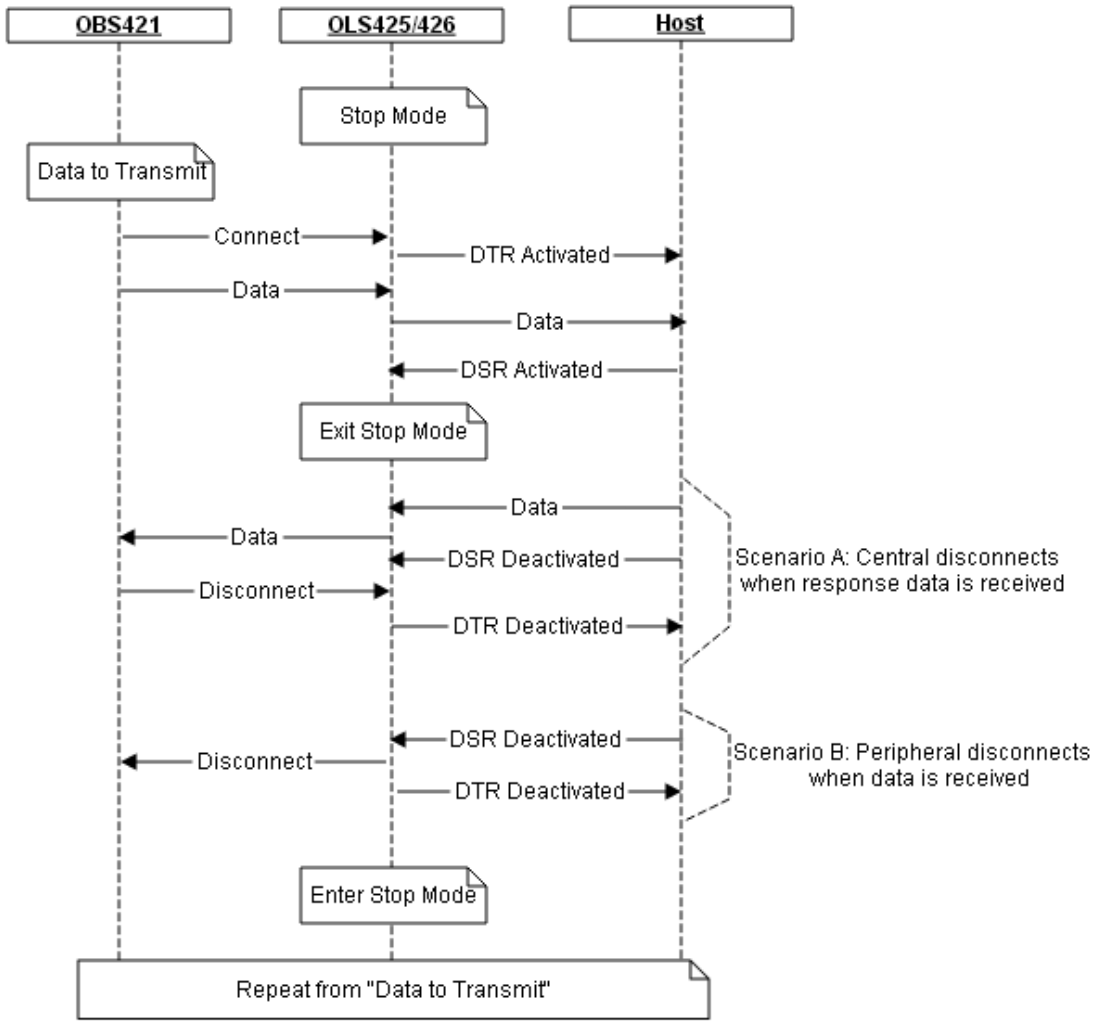
8.2.2 OBS421

At Command	Description
ATS6004=6	Min connection interval (Default value 6) 6 * 1.25ms = 7.5ms
ATS6005=8	Max connection interval (Default value 8) 8 * 1.25ms = 10ms
ATS6008=16	Scan window (Default value 16) 16 * 0.625ms = 10ms
ATS6009=16	Scan interval (Default value 16) 16 * 0.625ms = 10ms
ATS6010=0	Always connect interval (default 10000) 0 * 1 ms = 0 ms

Note that with the above configuration of the scan parameters, the OBS421 will continuously be scanning.

8.3 Use Case 3 - Central Data; Disconnected

The use case describes a setup where the central (OBS421) sets up the connection, transmits data and where the connection is active only when there is data to transmit. Hence, every time the central has data to transmit, the connection is setup.



There are two ways to terminate the connection. In scenario A, the host of the peripheral must send a response and when received by the central, it disconnects. Another option, scenario B, is for the host of the peripheral to disconnect once the expected data is received.

The OLS425/426 is configured as a server-peripheral and the OBS421 as a client-central. For basic connection setup, see section Basic Connection Setup Configuration

Note that there is a trade-off between low power consumption and short connection setup. The advertisement interval and scan window/interval should be configured to match the requirements of the specific use case.

8.3.1 OLS425/426

At Command	Description
AT*AMPM=3,1	Stop Mode The module is allowed to enter stop mode.
ATS6001=160	Min advertisement interval (Default value is 1600) 160 * 0.625 ms = 100 ms
ATS6002=160	Max advertisement interval (Default value is 2000) 160 * 0.625ms = 100 ms
Scenario A: AT*AMDS2=0,0x03,1 Scenario B: AT*AMDS2=0,0x23,1	Stop mode is controlled by DSR. Note that even with DSR inactive, it is possible for the central to transmit data and for the peripheral to forward received data to the host on the UART. Scenario B: When DSR is deactivated, the LE connection is terminated.
AT*AMPSF=0x00,1	Stop mode is not affected by connection (Default).

8.3.2 OBS421

At Command	Description
ATS6004=6	Min connection interval (Default value 6) 6 * 1.25 ms = 7.5 ms
ATS6005=8	Max connection interval (Default value 8) 8 * 1.25 ms = 10 ms
ATS6008=16	Scan window (Default value 16) 16 * 0.625 ms = 10 ms
ATS6009=16	Scan interval (Default value 16) 16 * 0.625 ms = 10 ms

Note that with the above configuration of the scan parameters, the OBS421 will continuously be scanning.

9 AT Commands/Events Summary

This section is a description of the AT commands that are added/changed or extended to support also Bluetooth low energy. See ref [3] for all AT commands.

9.1 Standard AT Commands

Command	Comment
S-Register (ATS)	Bluetooth low energy connection parameters are set using the S-Register command. Currently the following parameters can be set: • Link Loss timeout • Advertising Interval Min • Advertising Interval Max • Advertising Channel Map Bitmap • Connection Interval Min • Connection Interval Max • Slave latency • Create Connection Timeout • Scan Window • Scan Interval • Always connected interval • UART receive timeout • Packet length • Tx power • Rx gain • Initial online period

9.2 GAP Commands

Command	Comment
Security Mode (AT*AGSM)	Security mode "Display Yes/No" is not supported for low energy modules.
Name Discovery (AT*AGND)	To retrieve the device name of a remote low energy device two additional parameters has to be passed to the AT*AGND command. The first is a timeout and the second is mode (distinguishing Bluetooth low energy from Classic Bluetooth). Example: AT*AGND=3c2db784e0345,0,1
Inquiry (AT*AGI)	New inquiry types for Bluetooth Low Energy has been added, applicable to central role units. Both an LE inquiry type corresponding to the normal general extended inquiry, and also inquiry types suitable for reading customized advertise data and scan response data (AT*AGAD / AT*AGSD).
Bond (AT*AGB)	To bond with a low energy device an additional mode-parameter has to be passed to the AT*AGB command. Example: AT*AGB=3c2db784e0345,1
Read Bonded Devices (AT*AGBD)	To read bonded low energy devices an additional mode-parameter has to be passed to the AT*AGBD command. Example "Read all bonded BLE devices": AT*AGBD=1
Bluetooth Low Energy Role (AT*AGLE)	The Bluetooth Low Energy Role command is used to enable Bluetooth low energy functionality in the OBS421 module. This command also the defines if the device shall act as a central or as a peripheral device. Note that Bluetooth low energy by default is disabled on the OBS421. Use this command to enable it. For the OLS425 Bluetooth low energy is always enabled. Exa mple: "Enable Central functionality": AT*AGLE=1,1
Low Energy Advertise Data (AT*AGAD)	Command for use of custom advertise data. This allows for small data packages with low update frequency to be transferred without the need of establishing a connection.
Low Energy Scan Response Data (AT*AGSD)	Command for use of custom scan response data. This allows for small data packages with low update frequency to be transferred without the need of establishing a connection.

9.3 Data Mode Commands

Commands	Comment
Default Client Profile (AT*ADDCP)	Client profile "Serial Port Service Bluetooth Low Energy" is added for low energy modules.
Default Server Profile (AT*ADDSP)	Server profile "Serial Port Service Bluetooth Low Energy" is added for low energy modules.
Default Remote Peer (AT*ADRDRP/AT*ADWDRP)	A bit for Bluetooth Low Energy has been added to the connect scheme parameter. For low energy modules acting as peripherals like the OLS425/OLS426 client functionality is implemented using directed advertisement. Directed advertisements may jam the advertisement channels and if this functionality is used care must taken not have situation where the device tries to connect no a non existing device or a device that will not accept the connection.
Wireless Multidrop Configuration (AT*ADWM)	OLS425/OLS426 only supports the peripheral role and one active connection. The OBS421 supports up to three connected low energy peripherals.

9.4 Miscellaneous Commands

Command	Comment
RS232 Settings (AT*AMRS)	The OLS425/OLS426 only support a limited number of baudrates compared to the OLS421. Baudrates from 1200 to 57600 are supported.
DTR DSR Settings Low Energy (AT*AMDS2)	Used to configure the OLS425/OLS426 to let the host control the module operation using the DTR and DSR pins of the UART interface to achieve a customized behavior.
Power Save Feature Low Energy (AT*AMPSF)	Configure the OLS425/OLS426 to use different power save features, like waking up (exit low-power mode) when a connection is established.