BLE Time Sync

1.0

Features

- BLE CTS Service GATT Client role operation
- Sleep mode support
- Reporting the workflow status through UART
- LED status indication

General Description

This example project demonstrates the Time profile operation of the BLE PSoC Creator Component. The Time Sync example utilizes the BLE Time Profile (configured for GAP peripheral role as Time Client) with one instance of Current Time Service to demonstrate capability of time synchronization from the external Time Server. The project also contains one instance of Reference Time Update Service and one instance of Next DST Change Service, however they are not used in the example.
The Time Client operates with other devices, which implement the Time Server Profile role. The device uses Limited Discovery mode during which it is visible for BLE Servers. The device remains in the sleep mode between BLE connection intervals.

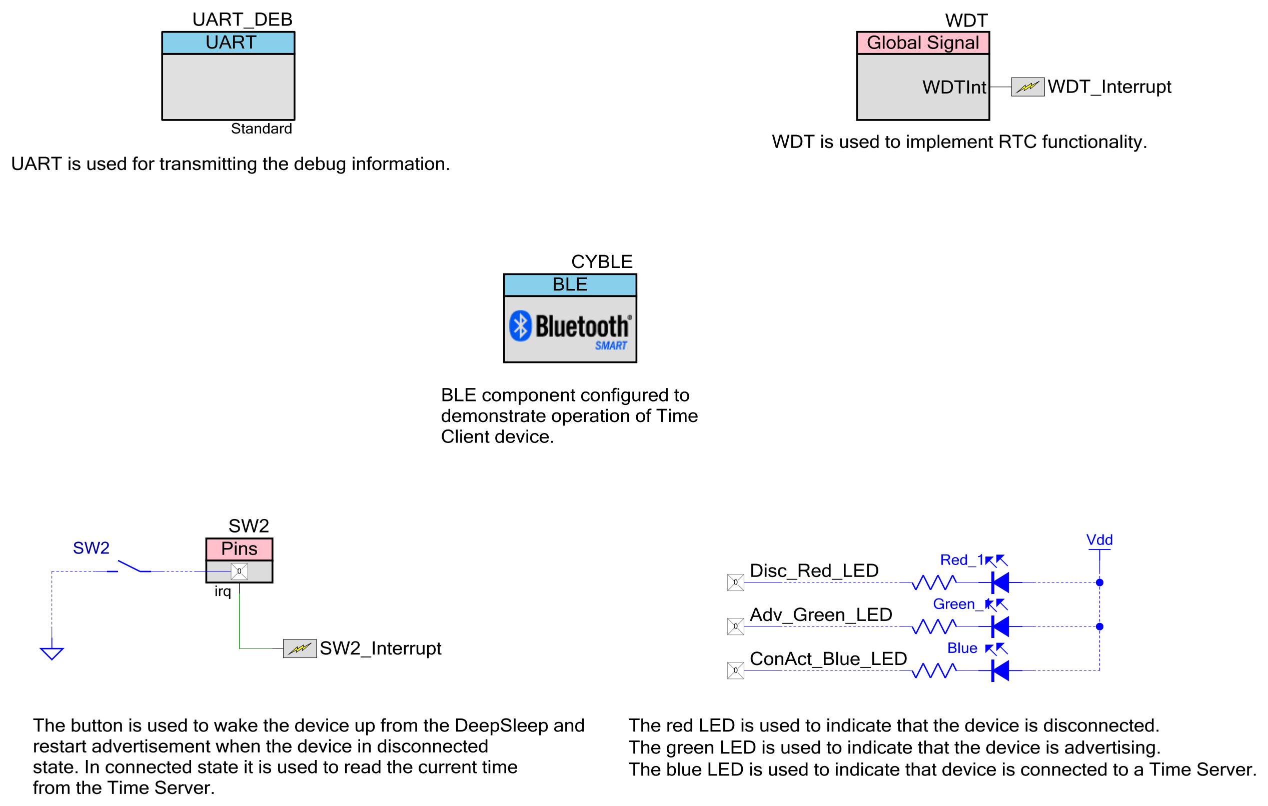
Development Kit Configuration

Configure your device as follows:

- The UART RX pin is connected to port 1 pin 4.
- The UART TX pin is connected to port 1 pin 5.
- A mechanical button (port 2 pin 7) is used to wake up the device and start re-advertising also it allows synchronizing of the current time from Time Server device.
- The red LED (port 2 pin 6) is used to indicate the BLE disconnection state.
- The green LED (port 3 pin 6) is used to indicate the advertising state.
- The blue LED (port 3 pin 7) is used to indicate that device is connected to a Time Server.

Project Configuration

The top design schematic is shown in **Figure 1**.



UART is used for transmitting the debug information.

WDT is used to implement RTC functionality.

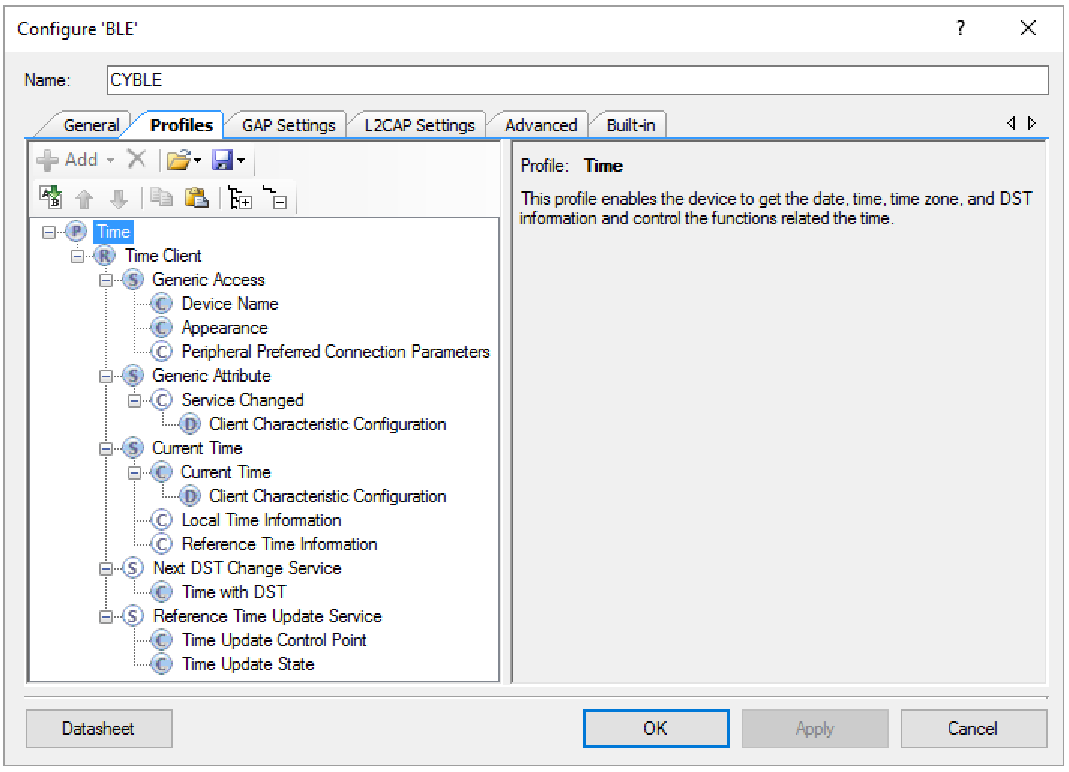
BLE component configured to demonstrate operation of Time Client device.

The button is used to wake the device up from the DeepSleep and restart advertisement when the device in disconnected state. In connected state it is used to read the current time from the Time Server.

The red LED is used to indicate that the device is disconnected.
The green LED is used to indicate that the device is advertising.
The blue LED is used to indicate that device is connected to a Time Server.

Figure 1. Top design schematic

The BLE component is configured as Time Client in the GAP Peripheral role.



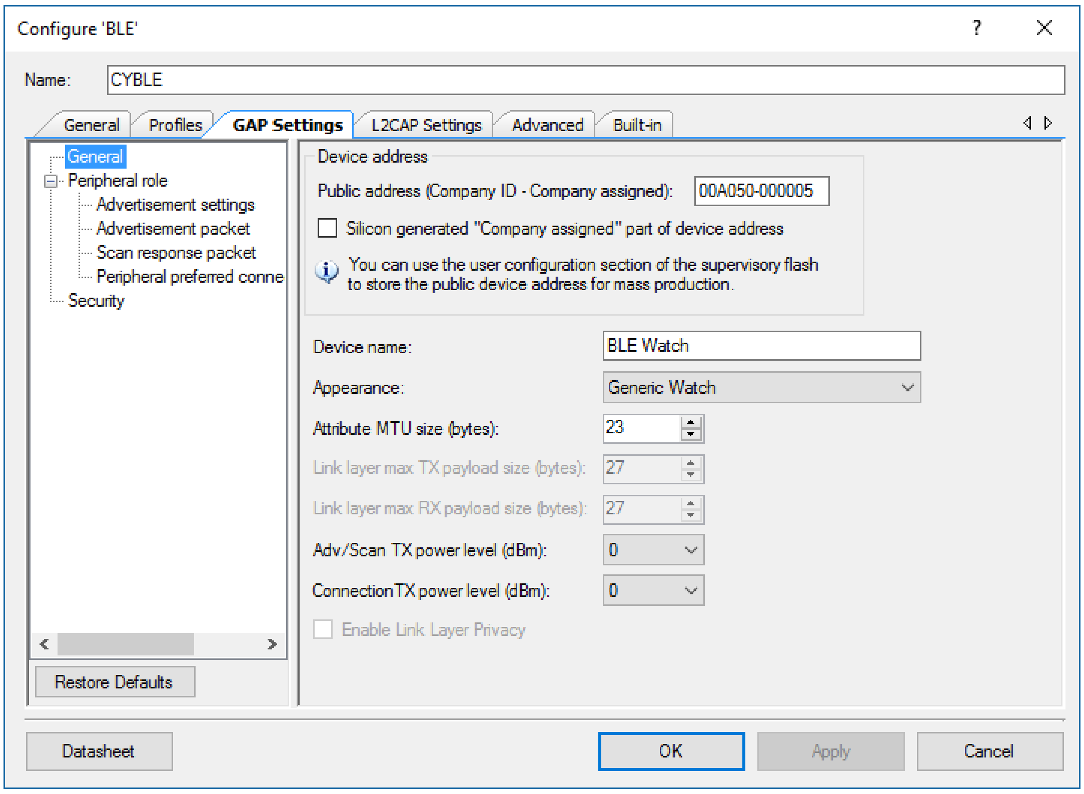
Name: CYBLE

- Time
  - Time Client
    - Generic Access
      - Device Name
      - Appearance
      - Peripheral Preferred Connection Parameters
    - Generic Attribute
      - Service Changed
        - Client Characteristic Configuration
    - Current Time
      - Current Time
        - Client Characteristic Configuration
      - Local Time Information
      - Reference Time Information
    - Next DST Change Service
      - Time with DST
    - Reference Time Update Service
      - Time Update Control Point
      - Time Update State

Profile: **Time**

This profile enables the device to get the date, time, time zone, and DST information and control the functions related the time.

Figure 2. GATT settings



Configure 'BLE'

Name: CYBLE

Device address

Public address (Company ID - Company assigned): 00A050-000005

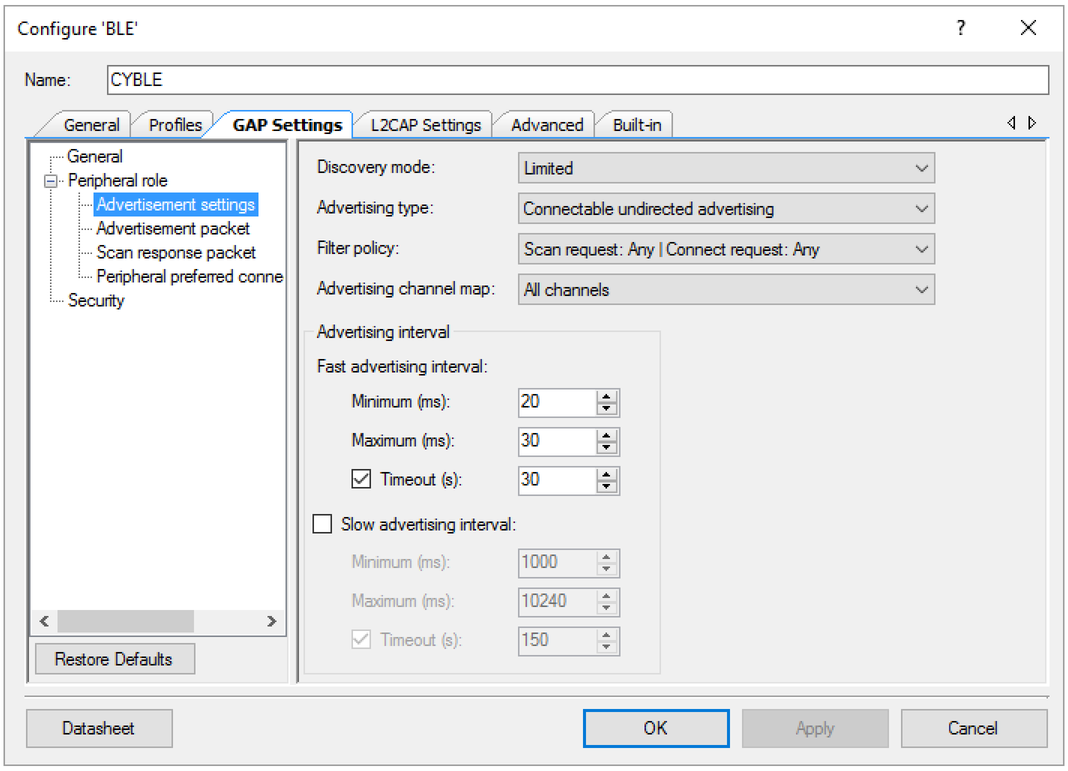
☐ Silicon generated "Company assigned" part of device address

You can use the user configuration section of the supervisory flash to store the public device address for mass production.

Device name:	BLE Watch
Appearance:	Generic Watch
Attribute MTU size (bytes):	23
Link layer max TX payload size (bytes):	27
Link layer max RX payload size (bytes):	27
Adv/Scan TX power level (dBm):	0
Connection TX power level (dBm):	0

☐ Enable Link Layer Privacy

Figure 3. GAP settings



Configure 'BLE'

Name: CYBLE

Discovery mode:	Limited
Advertising type:	Connectable undirected advertising
Filter policy:	Scan request: Any | Connect request: Any
Advertising channel map:	All channels

Advertising interval

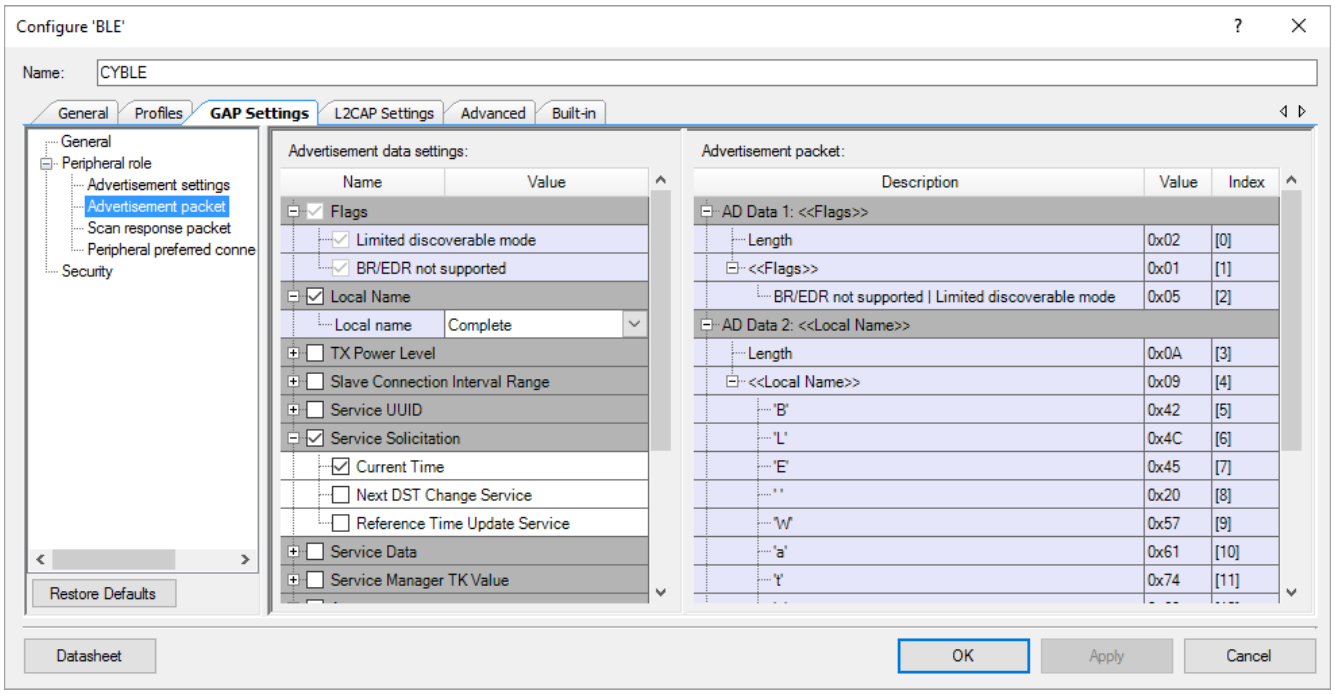
Fast advertising interval:

Minimum (ms):	20
Maximum (ms):	30
☑ Timeout (s):	30

☐ Slow advertising interval:

Minimum (ms):	1000
Maximum (ms):	10240
☑ Timeout (s):	150

Figure 4. GAP settings -> Advertisement settings


Configure 'BLE'

Name: CYBLE

General | Profiles | **GAP Settings** | L2CAP Settings | Advanced | Built-in

- General
- Peripheral role
  - Advertisement settings
  - Advertisement packet
  - Scan response packet
  - Peripheral preferred conne
- Security

Advertisement data settings:

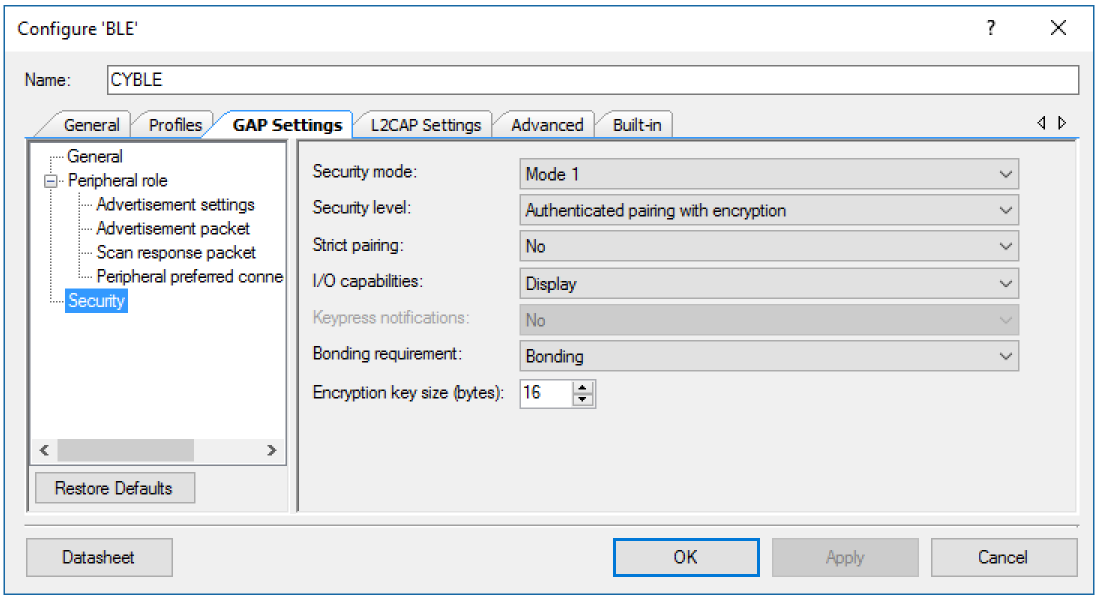
Name	Value
☑ Flags	
☑ Limited discoverable mode	
☑ BR/EDR not supported	
☑ Local Name	
Local name	Complete
☐ TX Power Level	
☐ Slave Connection Interval Range	
☐ Service UUID	
☑ Service Solicitation	
☑ Current Time	
☐ Next DST Change Service	
☐ Reference Time Update Service	
☐ Service Data	
☐ Service Manager TK Value	

Advertisement packet:

Description	Value	Index
AD Data 1: <<Flags>>		
Length	0x02	[0]
<<Flags>>	0x01	[1]
BR/EDR not supported | Limited discoverable mode	0x05	[2]
AD Data 2: <<Local Name>>		
Length	0x0A	[3]
<<Local Name>>	0x09	[4]
'B'	0x42	[5]
'L'	0x4C	[6]
'E'	0x45	[7]
' '	0x20	[8]
'W'	0x57	[9]
'a'	0x61	[10]
't'	0x74	[11]

Restore Defaults | Datasheet | OK | Apply | Cancel

Figure 5. GAP settings -> Advertisement packet


Configure 'BLE'

Name: CYBLE

General | Profiles | **GAP Settings** | L2CAP Settings | Advanced | Built-in

- General
- Peripheral role
  - Advertisement settings
  - Advertisement packet
  - Scan response packet
  - Peripheral preferred conne
- Security

Security mode:	Mode 1
Security level:	Authenticated pairing with encryption
Strict pairing:	No
I/O capabilities:	Display
Keypress notifications:	No
Bonding requirement:	Bonding
Encryption key size (bytes):	16

Restore Defaults | Datasheet | OK | Apply | Cancel

Figure 6. Security settings

Project Description

The example project demonstrates the core functionality of the BLE component configured as a Time Client. For operation the example project requires the Time Server device configured in the GAP Central role. Currently the Time Server in the GAP Central role only supported by iOS (iPod, iPhone). Please note that CySmart app. also doesn't support Time Server in the GAP Central role so the standard iOS menu should be used to connect to the Time Client.

To start the example project operation, build it and program onto PSoC 4100-BL or PSoC 4200-BL device. Right after the startup, the device performs BLE component initialization as well as initialization of UART, WDT and ISR components. In this project three callback functions are required for the BLE operation. One callback function (AppCallBack()) is required for receiving generic events from BLE Stack and the CtsAppEventHandler() required for receiving events from Current Time Service. There are also two additional callback functions – RtusAppEventHandler() and NdcsAppEventHandler() which are not actually used in the example project as they are optional in the Time profile, but are present for user reference.
The CYBLE_EVT_STACK_ON event indicates a successful initialization of the BLE Stack. After this event is received the component starts advertising with the packet structure as configured in the BLE component customizer (see **Figure 5**). As the BLE component is configured in the Limited Discovery mode, it stops advertising once the 30 seconds advertising period expires. To resume advertising, press the mechanical button, this will resume advertising for another 30 seconds. The blinking green LED indicates that the device is advertising. Glowing red LED indicates that no Central device was connected to the Client within of 30 seconds Limited Discovery period.

While connected to a Client and between connection intervals, the device is put into the sleep mode.

The example project uses UART component for displaying debug information as also for entering commands through the Terminal emulator app. Commands are the procedures which user can perform. The list of the commands is shown below:

Command	Description
'd'	Send disconnect request to the peer device.
'1'	Enable notifications for the Current Time Characteristic.
'o'	Display local device's current time.

After the Time Client is performing connection to the Time Server device, it will prompt a passkey, which should be entered on the Server's side to make secure connection. When the passkey is successfully entered, the Time Server is ready for time synchronization. Prior to the synchronization with the Server, the Client is initialized with the following current time value:

Current time: 00:00:00: Unknown day of week 1.1.1900

The Watchdog Timer component is used to simulate the Real Time clock which updates the current time every second. In this example project the WDT is configured to update only seconds, minutes and hours. To see the current time on the Client, type ‘o’ in the Terminal emulator app, and it will display it. To synchronize the time from the Time Server, press the mechanical button and it will send a read request to the Server to read Current Time Characteristic. When the Client receives the response, it will display the updated time on the Terminal. Also with this example project it is possible to enable notifications from the Time Client. For this purpose enter ‘1’ in the Terminal. After that the current time will be updated each time the notification with a new time value is received.

Note that to send the time notifications from the iOS the time on the iOS device should be changes for that value greater than 1 minute.

Expected Results

The example project uses UART to display debug information. The example of a log captured on UART with Time Client is shown below:

```
Bluetooth On. Start advertisement with addr:00a050000005.
Device is entered Limited Discovery mode.

Advertisement is enabled

CYBLE_EVT_GATT_CONNECT_IND: 0

CYBLE_EVT_DEVICE_CONNECTED: 4
Start Discovery

Discovery complete.
Discovered services:
Service with UUID 0x1800 has handle range from 0x1 to 0x5
Service with UUID 0x1801 has handle range from 0x6 to 0x9
Service with UUID 0x1805 has handle range from 0x31 to 0x36
Service with UUID 0x1807 has handle range from 0x0 to 0x0
Service with UUID 0x1806 has handle range from 0x0 to 0x0

CYBLE_EVT_GAP_AUTH_REQ: security=0x2, bonding=0x1, ekeySize=0x10,
err=0x0

CYBLE_EVT_GAP_PASSKEY_DISPLAY_REQUEST. Passkey is: 010946.
Please enter the passkey on your Server device.

CYBLE_EVT_GAP_ENCRYPT_CHANGE: 1

CYBLE_EVT_GAP_KEYINFO_EXCHNGE_CMPLT

CYBLE_EVT_GAP_AUTH_COMPLETE: security: 0x2, bonding: 0x1, ekeySize:
0x10, authErr 0x0
Bonding complete.

CyBle_CtscSetCharacteristicDescriptor() routine Success
CTS Current Time CCCD was written successfully
```

```
Current time: 00:01:15: Unknown day of week  1.1.1900

CyBle_CtscGetCharacteristicValue() routine Success
CTS characteristic read response received
Current time: 19:41:42: Wed  12.3.2014

Current Time Characteristic notification received
Current time: 19:50:00: Wed  12.3.2014


Disconnect Device routine Success.

CYBLE_EVT_GATT_DISCONNECT_IND:

CYBLE_EVT_DEVICE_DISCONNECTED
Current time: 19:51:13: Wed  12.3.2014
```