BLE Wireless Power Transmitter and Receiver 1.0

Features

- BLE Wireless Power Transfer Profile in Power Receiver Unit and Power Transmitter Unit role
- Wireless Power Transfer Service
- Deep Sleep mode support
- LED status indication

Development Kit Configuration

1. Default CY8CKIT-042 BLE Pioneer Kit configuration (including BLE Dongle).
2. Connect J2 pin P3[0] to J3 pin VREF.
3. Make sure J16 is set to 3.3V

General Description

This example project demonstrates the Wireless Power Transfer Profile operation of the BLE PSoC Creator Component. The example project requires CY8CKIT-042-BLE Pioneer Kit.

The example consists of two projects: Wireless Power Transmitter (GATT Client) and Wireless Power Receiver (GATT Server).

This example demonstrates communication between Power Receiver Unit (PRU) and Power Transmitter Unit (PTU) in the Wireless Power Transfer systems. PTU central device supports time multiplexing connection with up to 8 PRU peripheral devices. PRU simulates power receiver data and reports the simulated data to a PTU using the Wireless Power Transfer Service (WPTS).

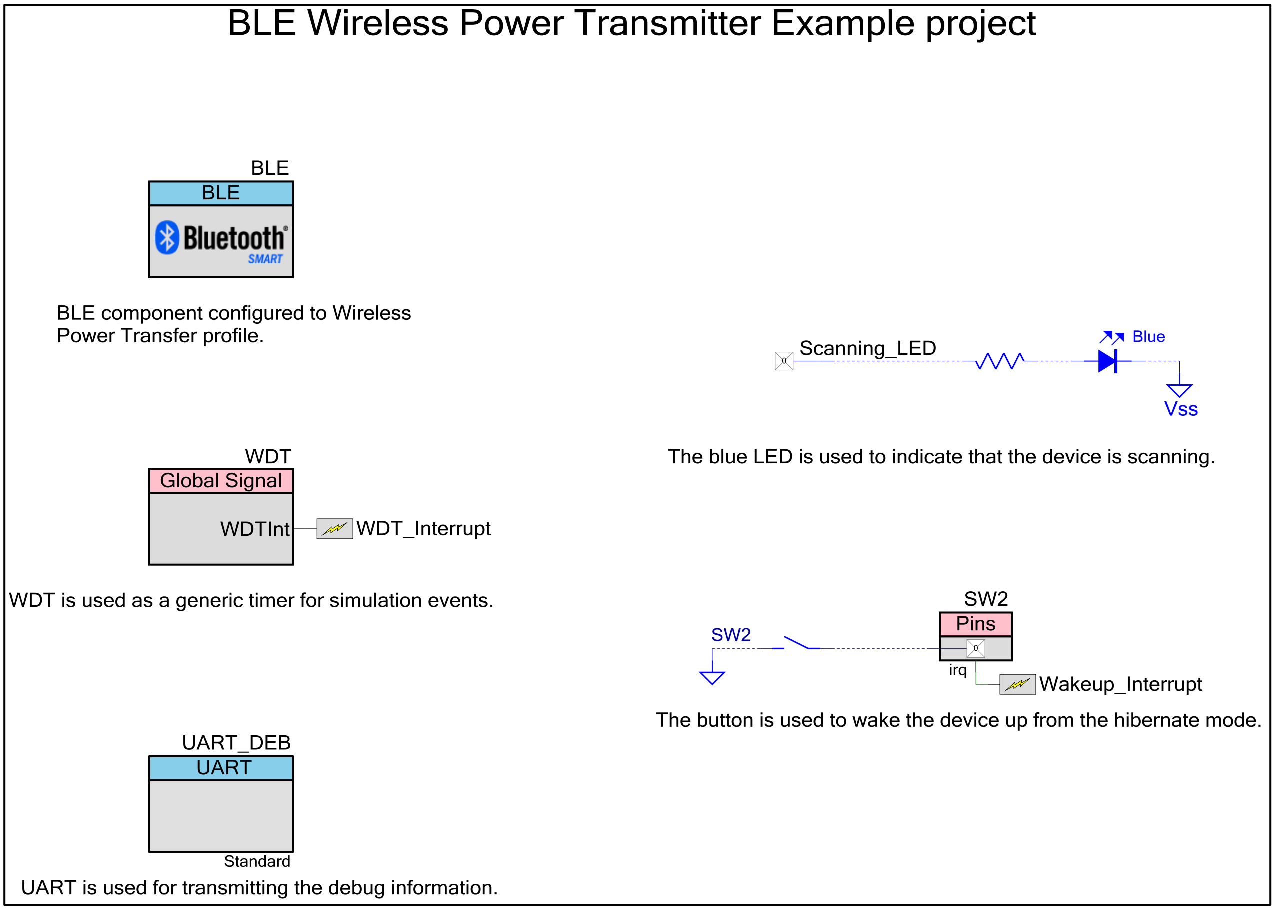
BLE Wireless Power Transmitter Project Configuration

The transmitter example project consists of the following components:

- Bluetooth Low Energy (BLE)
- UART
- Watchdog timer (WDT) that is used as a general timer

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

Figure 1. Top design schematic



BLE component configured to Wireless Power Transfer profile.

WDT is used as a generic timer for simulation events.

UART is used for transmitting the debug information.

The blue LED is used to indicate that the device is scanning.

The button is used to wake the device up from the hibernate mode.

The output pins are used to reflect the line signal output on the LED. The input pin is configured to the resistive pull up mode and is used to wake device from low power hibernate mode.

UART

This component is used for printing debug information and scan commands from terminal.

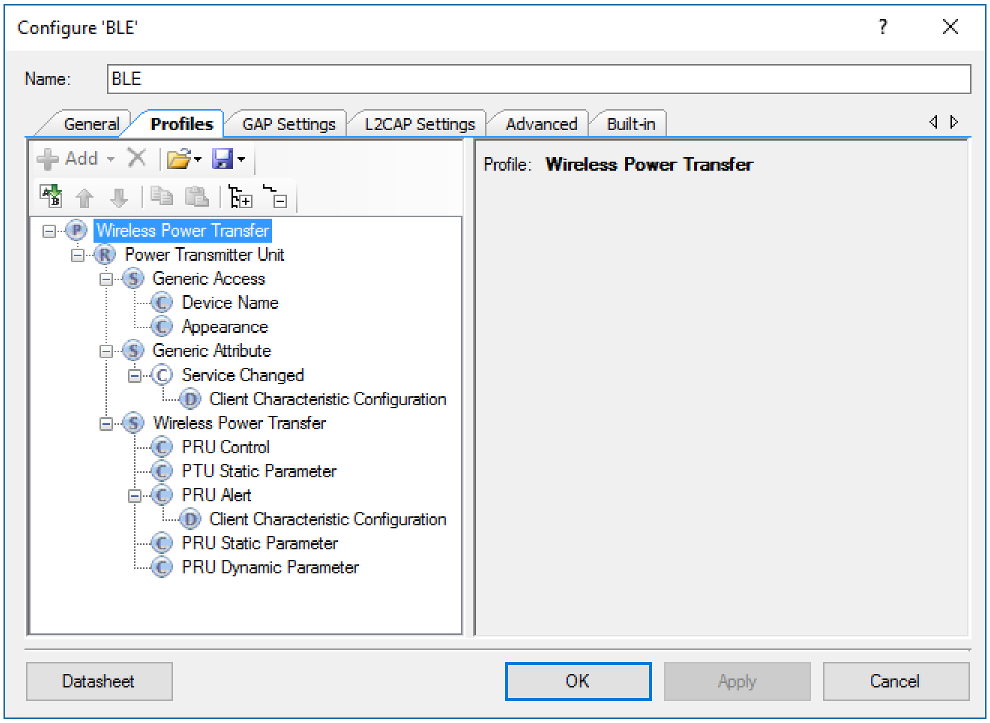
Watch Dog Timer (WDT)

The Global Signal Reference and WDT_Interrupt component is used for the ISR configuration from Watch Dog Timer. This timer works over low power Deep Sleep mode, therefore used as a general timer for LED indication and simulation purpose.

Bluetooth Low Energy (BLE)

The BLE component is configured as Wireless Power Transfer Profile in Power Transmitter Unit role operation.

Figure 2. GATT settings

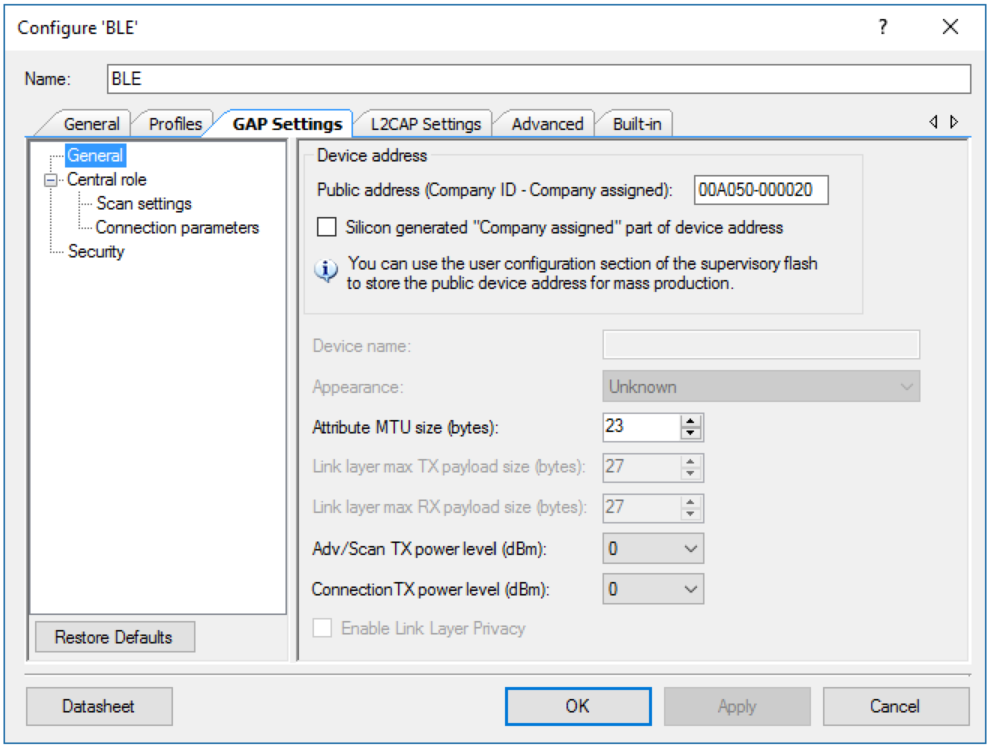


Name: BLE

- Wireless Power Transfer
  - Power Transmitter Unit
    - Generic Access
      - Device Name
      - Appearance
    - Generic Attribute
      - Service Changed
        - Client Characteristic Configuration
    - Wireless Power Transfer
      - PRU Control
      - PTU Static Parameter
      - PRU Alert
        - Client Characteristic Configuration
      - PRU Static Parameter
      - PRU Dynamic Parameter

Profile: **Wireless Power Transfer**

Figure 3. GAP settings



Name: BLE

Device address

Public address (Company ID - Company assigned): 00A050-000020

□ 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:

Appearance: Unknown

Attribute MTU size (bytes): 23

Link layer max TX payload size (bytes): 27

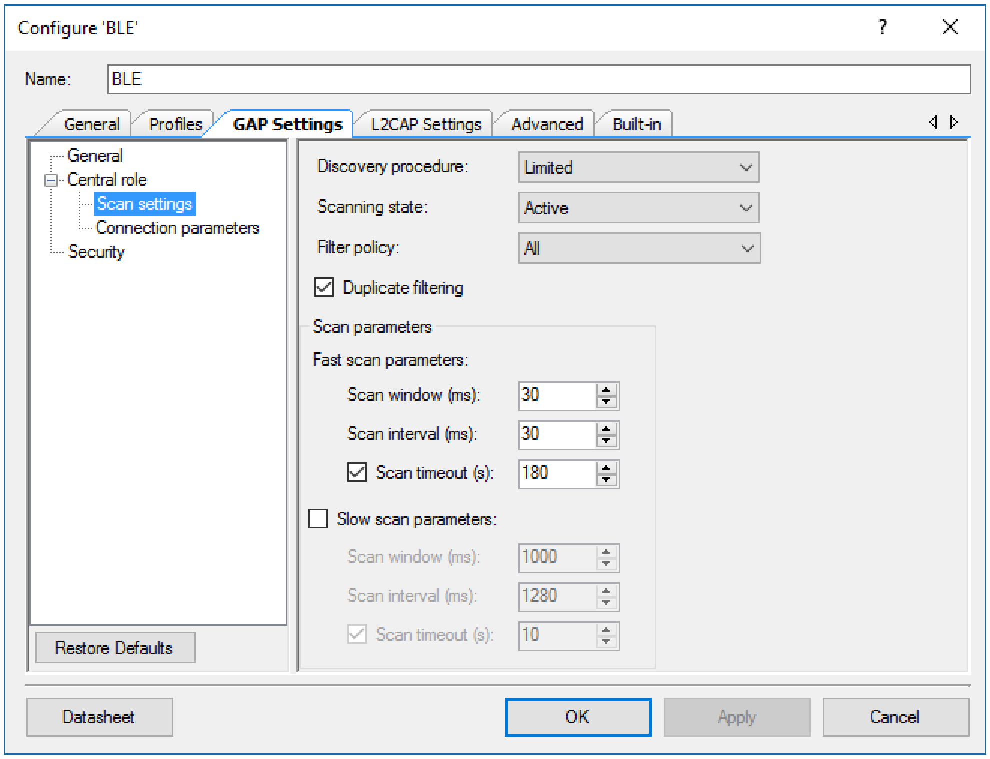
Link layer max RX payload size (bytes): 27

Adv/Scan TX power level (dBm): 0

Connection TX power level (dBm): 0

□ Enable Link Layer Privacy

Figure 4. GAP settings -> Scan settings



Name: BLE

Discovery procedure: Limited

Scanning state: Active

Filter policy: All

☑ Duplicate filtering

Scan parameters

Fast scan parameters:

Scan window (ms): 30

Scan interval (ms): 30

☑ Scan timeout (s): 180

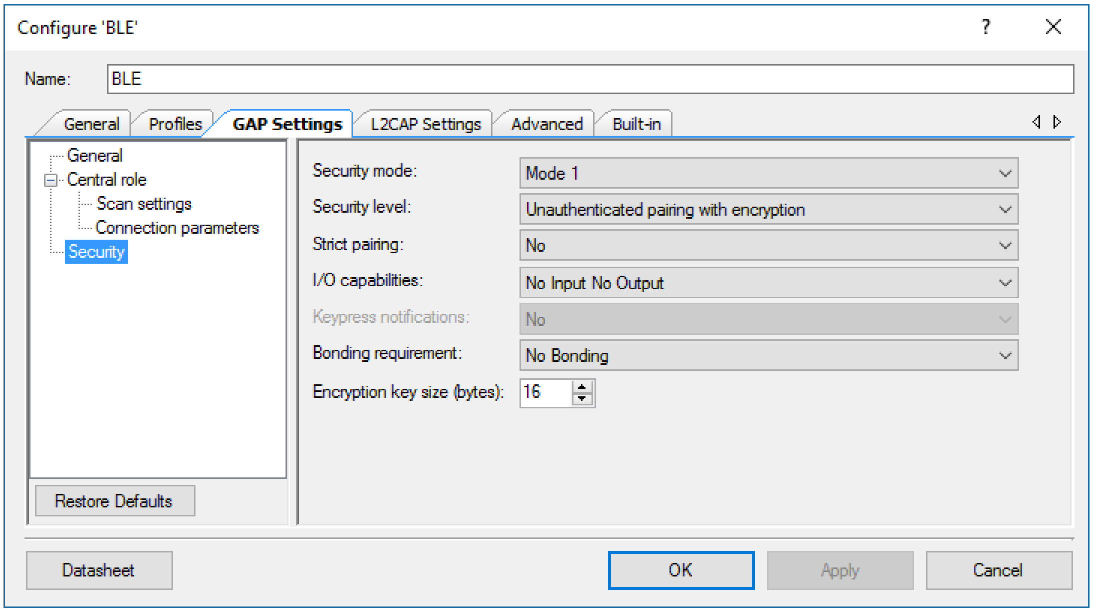
□ Slow scan parameters:

Scan window (ms): 1000

Scan interval (ms): 1280

☑ Scan timeout (s): 10

Figure 5. Security settings



Name: BLE

Security mode: Mode 1

Security level: Unauthenticated pairing with encryption

Strict pairing: No

I/O capabilities: No Input No Output

Keypress notifications: No

Bonding requirement: No Bonding

Encryption key size (bytes): 16

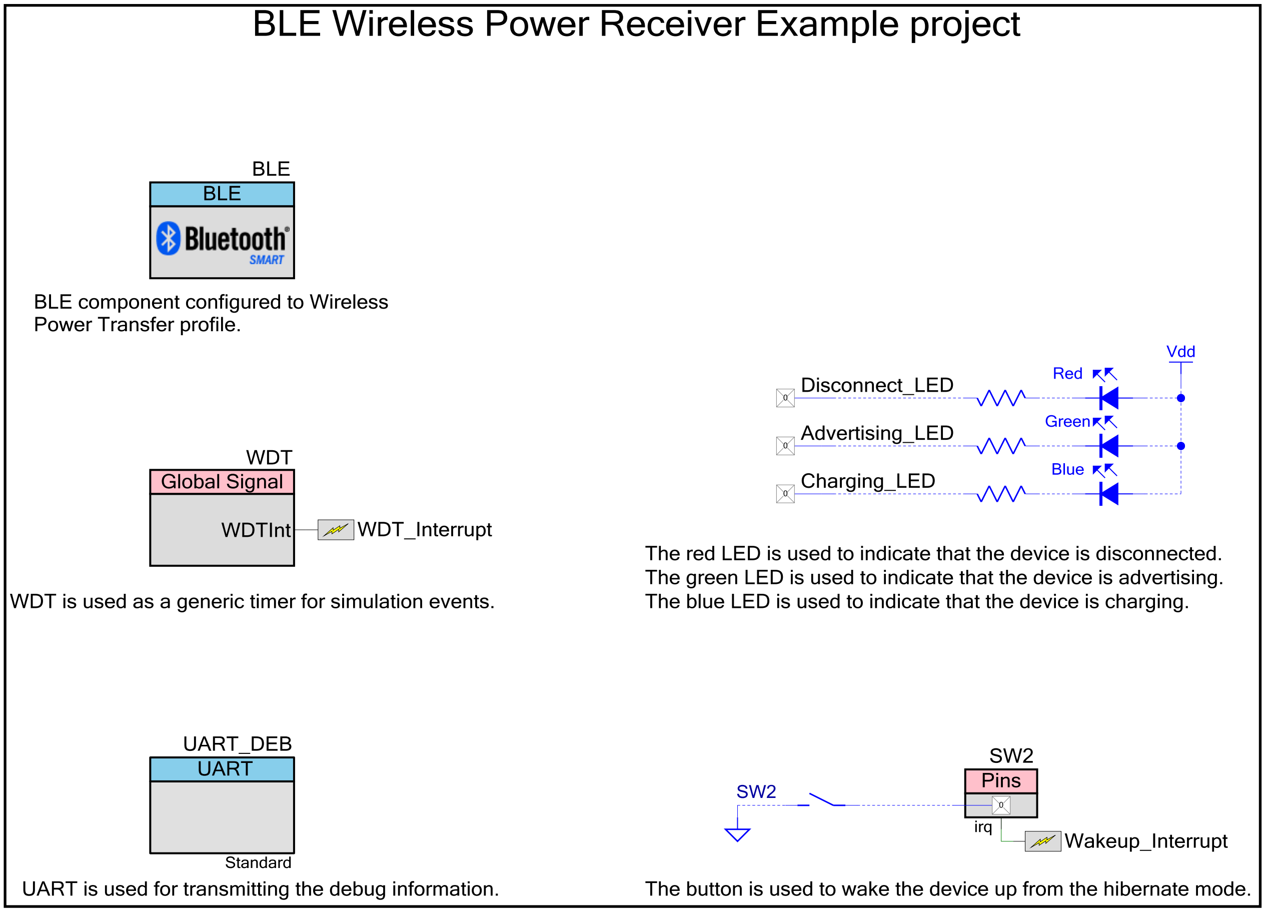
BLE Wireless Power Receiver Project Configuration

The receiver example project consists of the following components:

- Bluetooth Low Energy (BLE)
- UART
- Watchdog timer (WDT) that is used as general timer
- Sequencing SAR ADC, Opamp, DieTemp

The top design schematic is shown in Figure **6** and Figure **7**.

Figure 6. WPRU tab of top design schematic


BLE Wireless Power Receiver Example project
BLE
BLE
Bluetooth SMART
BLE component configured to Wireless Power Transfer profile.
WDT
Global Signal
WDTInt
WDT_Interrupt
WDT is used as a generic timer for simulation events.
UART_DEB
UART
Standard
UART is used for transmitting the debug information.
Vdd
Disconnect_LED
Red
Advertising_LED
Green
Charging_LED
Blue
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 the device is charging.
SW2
SW2
Pins
irq
Wakeup_Interrupt
The button is used to wake the device up from the hibernate mode.

The output pins are used to reflect the line signal output on the LED. The input pin is configured to the resistive pull up mode and is used to wake device from low power hibernate mode.

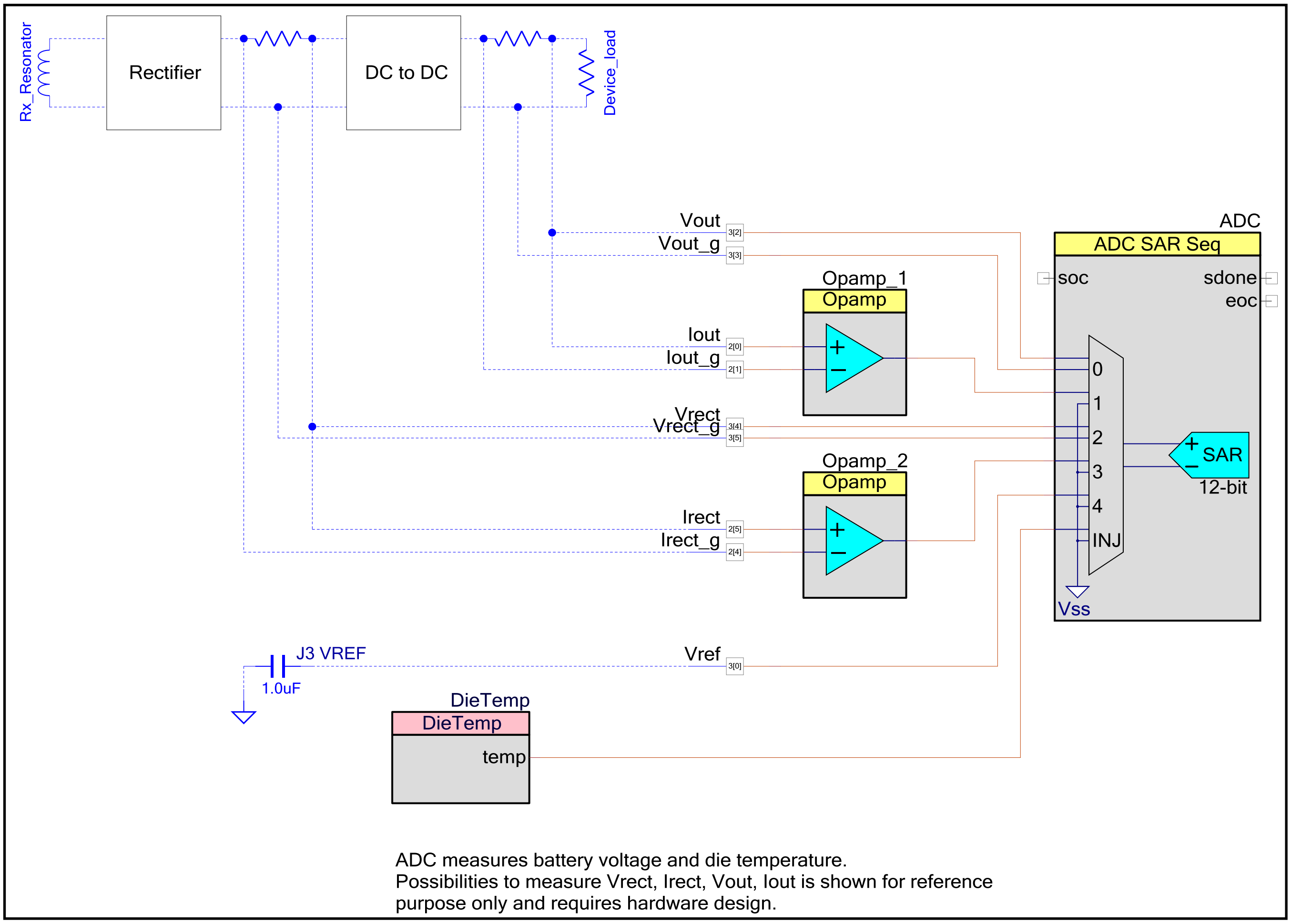
UART

This component is used for printing debug information.

Watch Dog Timer (WDT)

The Global Signal Reference and WDT_Interrupt component is used for the ISR configuration from Watch Dog Timer. This timer works over low power Deep Sleep mode, therefore used as a general timer for LED indication and simulation purpose.

Figure 7. Measure tab of top design schematic



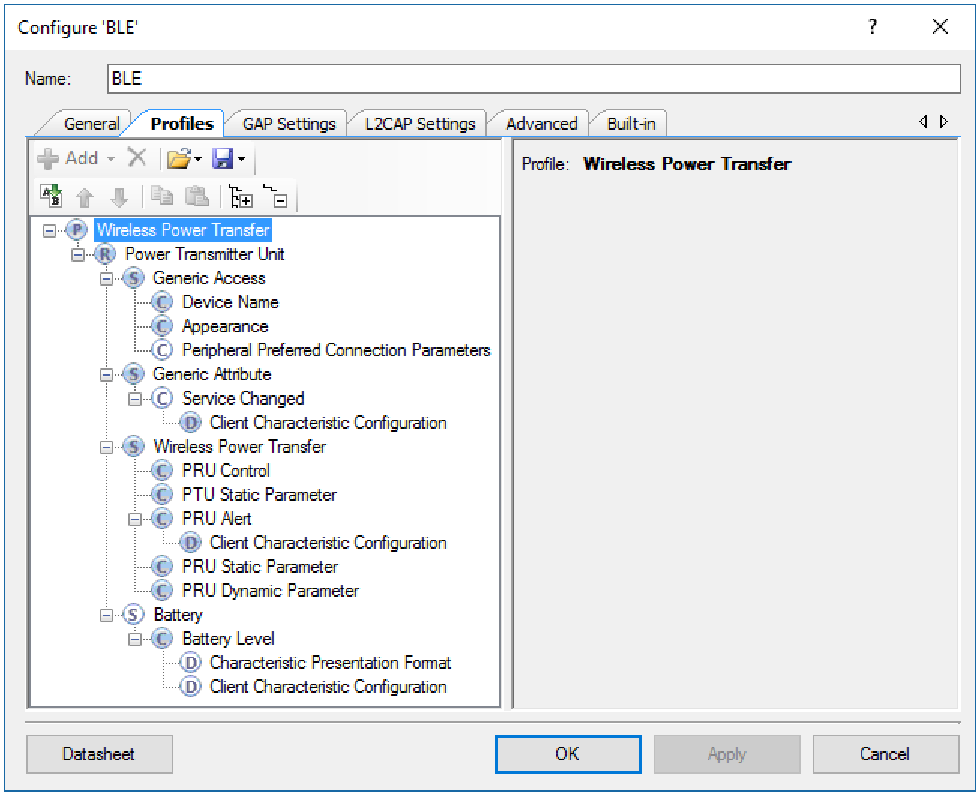
ADC measures battery voltage and die temperature.
Possibilities to measure Vrect, Irect, Vout, Iout is shown for reference purpose only and requires hardware design.

Bluetooth Low Energy (BLE)

The BLE component is configured as Wireless Power Transfer Profile in Power Receiver Unit role operation with the additional Battery Service (BAS). The purpose of the BAS service is to demonstrate battery level measurement possibilities along with measurement of other parameters for wireless power transfer system.

PRU Advertisement packet is configured to connectable undirected advertising type with WPTS specific-service data. Scan response packet includes a complete local name.

Figure 8. GATT settings

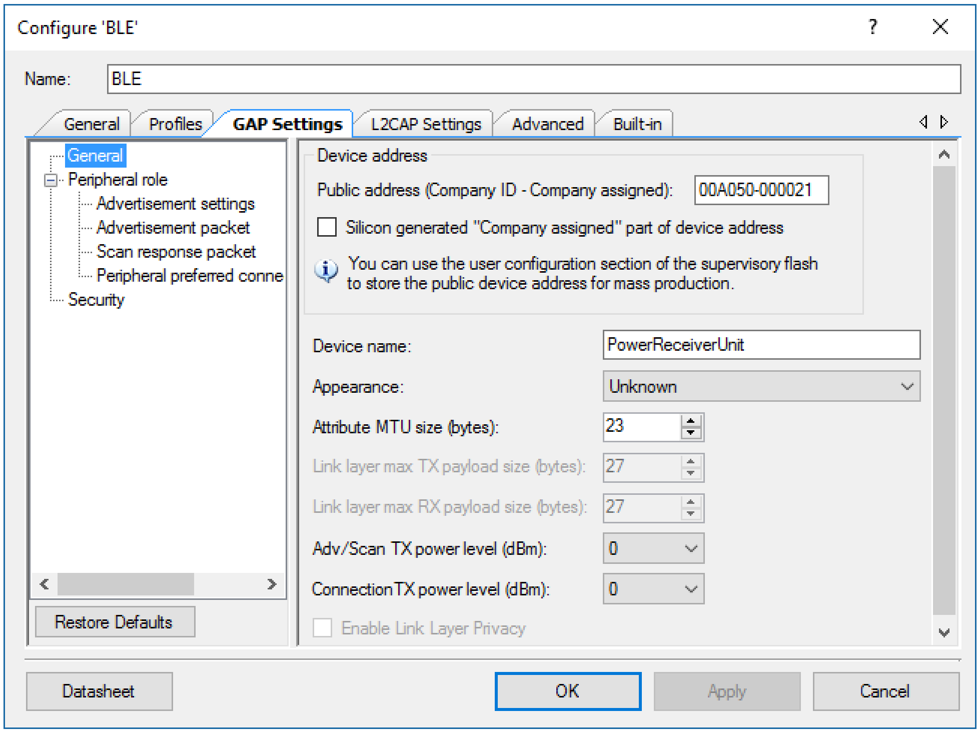


Name: BLE

Profile: **Wireless Power Transfer**

- Wireless Power Transfer
  - Power Transmitter Unit
    - Generic Access
      - Device Name
      - Appearance
      - Peripheral Preferred Connection Parameters
    - Generic Attribute
      - Service Changed
        - Client Characteristic Configuration
    - Wireless Power Transfer
      - PRU Control
      - PTU Static Parameter
      - PRU Alert
        - Client Characteristic Configuration
      - PRU Static Parameter
      - PRU Dynamic Parameter
    - Battery
      - Battery Level
        - Characteristic Presentation Format
        - Client Characteristic Configuration

Figure 9. GAP settings



Name: BLE

Device address

Public address (Company ID - Company assigned): 00A050-000021

□ 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: PowerReceiverUnit

Appearance: Unknown

Attribute MTU size (bytes): 23

Link layer max TX payload size (bytes): 27

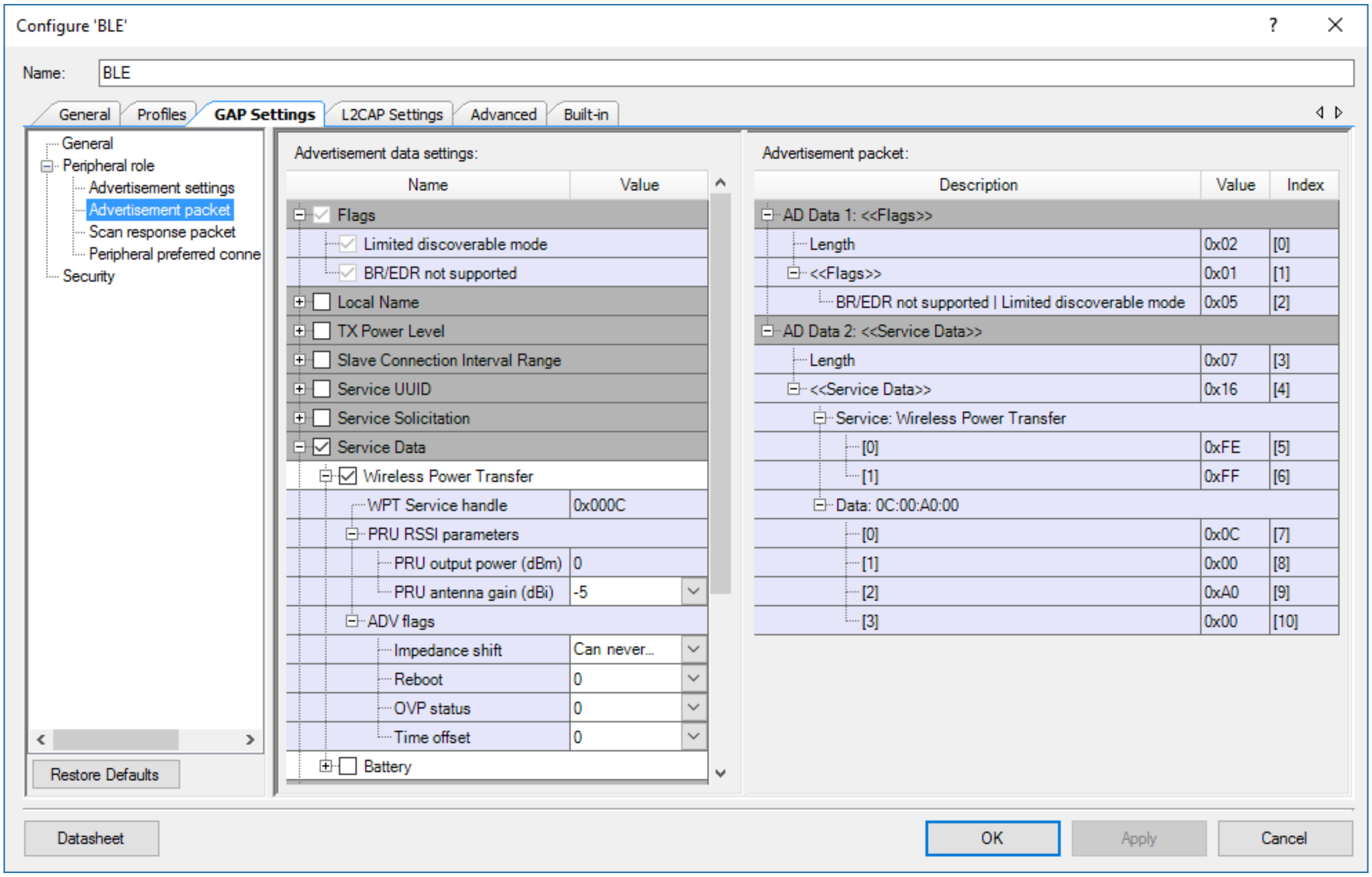
Link layer max RX payload size (bytes): 27

Adv/Scan TX power level (dBm): 0

Connection TX power level (dBm): 0

□ Enable Link Layer Privacy

Figure 10. GAP settings -> Advertisement settings



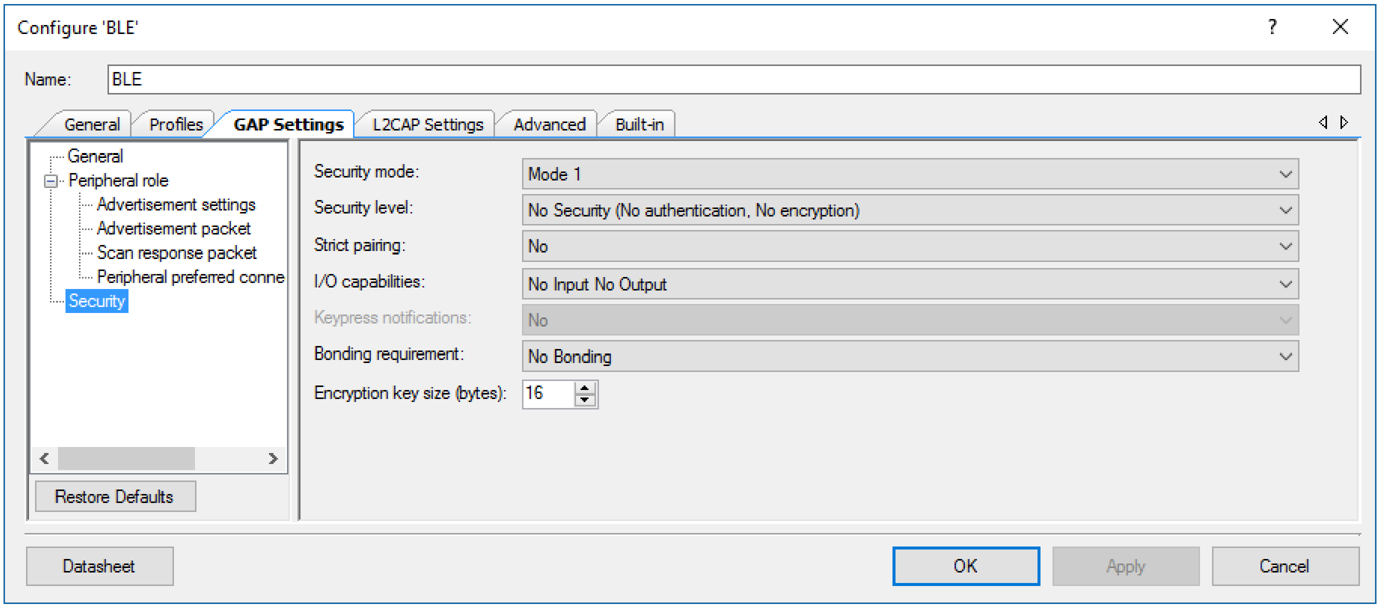
Advertisement data settings:

Name	Value
☑ Flags	
☑ Limited discoverable mode	
☑ BR/EDR not supported	
□ Local Name	
□ TX Power Level	
□ Slave Connection Interval Range	
□ Service UUID	
□ Service Solicitation	
☑ Service Data	
☑ Wireless Power Transfer	
WPT Service handle	0x000C
PRU RSSI parameters	
PRU output power (dBm)	0
PRU antenna gain (dBi)	-5
ADV flags	
Impedance shift	Can never...
Reboot	0
OVP status	0
Time offset	0
□ Battery	

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: <<Service Data>>		
Length	0x07	[3]
<<Service Data>>	0x16	[4]
Service: Wireless Power Transfer		
[0]	0xFE	[5]
[1]	0xFF	[6]
Data: 0C:00:A0:00		
[0]	0x0C	[7]
[1]	0x00	[8]
[2]	0xA0	[9]
[3]	0x00	[10]

Figure 11. Security settings



Security mode: Mode 1

Security level: No Security (No authentication, No encryption)

Strict pairing: No

I/O capabilities: No Input No Output

Keypress notifications: No

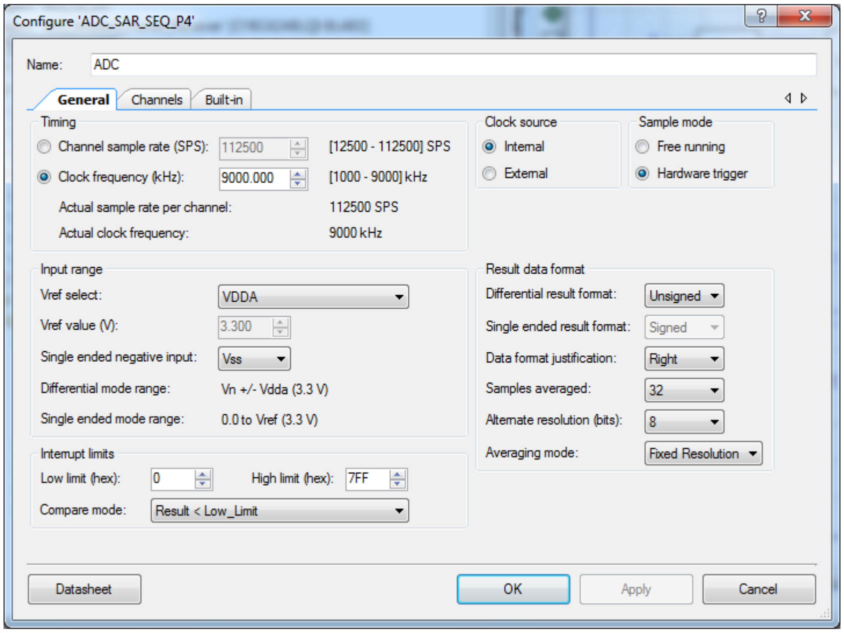
Bonding requirement: No Bonding

Encryption key size (bytes): 16

Sequencing SAR ADC

ADC is configured with five sequenced channels and injection channel measurement. Clock frequency is configured to the maximum possible (9 MHz) with Vref equal to VDDA. Acquisition time for injection channel is configured to 50 clks to have required at least 5 µs temperature sensor settling time.

Figure 12. ADC General settings



Name: ADC

Timing

○ Channel sample rate (SPS): 112500 [12500 - 112500] SPS

● Clock frequency (kHz): 9000.000 [1000 - 9000] kHz

Actual sample rate per channel: 112500 SPS

Actual clock frequency: 9000 kHz

Clock source

● Internal

○ External

Sample mode

○ Free running

● Hardware trigger

Input range

Vref select: VDDA

Vref value (V): 3.300

Single ended negative input: Vss

Differential mode range: Vn +/- Vdda (3.3 V)

Single ended mode range: 0.0 to Vref (3.3 V)

Interrupt limits

Low limit (hex): 0 High limit (hex): 7FF

Compare mode: Result < Low_Limit

Result data format

Differential result format: Unsigned

Single ended result format: Signed

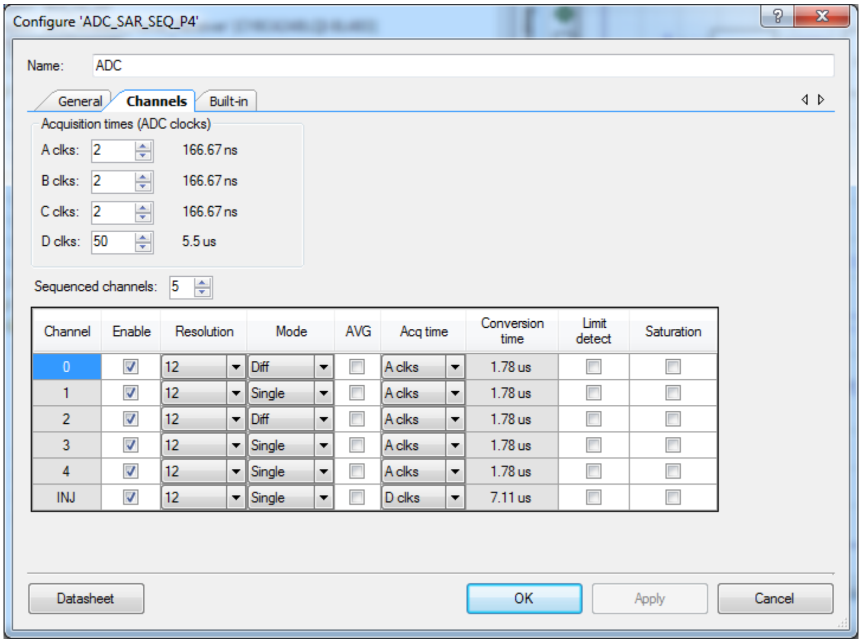
Data format justification: Right

Samples averaged: 32

Alternate resolution (bits): 8

Averaging mode: Fixed Resolution

Figure 13. ADC channels settings



Name: ADC

Acquisition times (ADC clocks)

A clks: 2 166.67 ns

B clks: 2 166.67 ns

C clks: 2 166.67 ns

D clks: 50 5.5 us

Sequenced channels: 5

Channel	Enable	Resolution	Mode	AVG	Acq time	Conversion time	Limit detect	Saturation
0	☑	12	Diff	□	A clks	1.78 us	□	□
1	☑	12	Single	□	A clks	1.78 us	□	□
2	☑	12	Diff	□	A clks	1.78 us	□	□
3	☑	12	Single	□	A clks	1.78 us	□	□
4	☑	12	Single	□	A clks	1.78 us	□	□
INJ	☑	12	Single	□	D clks	7.11 us	□	□

BLE Wireless Power Transmitter Project Description

The project demonstrates the BLE component functionality configured as PTU.
For operation the example project uses two callback functions - AppCallBack() and WptsCallBack(). One callback function (AppCallBack()) is required for receiving the generic events from the BLE Stack and the second (WptsCallBack()) is required for receiving the events from the WPTS Service.

To start the example project operation, build it and program into development kit BLE Dongle. Right after the startup, the BLE, UART, and ISR components are initialized. After the initialization the BLE component begins its operation that can be seen on the user LED which starts blinking with a blue color. This indicates that the device has started scanning. After 180 seconds timeout, if no peripheral device has been connected, the PTU stops scanning. User LED is turned off indicating the disconnection state and the system enters into the hibernate mode. Press the user button on BLE Dongle (SW2) to wake up the system and start re-scanning.

Advertising packets, received during scanning procedure from peripheral devices, are parsed and filtered. Only packets with WPT service-specific data are handled and showed in the debug terminal with the device sequence number as a candidate to connect.

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

Command	Description
'z'+'#'	Select specific peer device, where '#' – sequence number from advertising packet (0-7). Default value: 0.
'c'	Send connect request to selected peer device.
'd'	Send disconnect request to peer device.
'v'	Cancel connection request.
's'	Start discovery procedure.
'1'	Enable notifications for Alert characteristic.
'2'	Enable indications for Alert characteristic.
'3'	Disable notifications and indications for Alert characteristic.
'4'	Send Read request for PRU Static Parameter characteristic.
'5'	Send Read request for PRU Dynamic Parameter characteristic.
'6'	Send "Enable Charging" command to PRU control characteristic.
'7'	Send "Disable Charging" command to PRU control characteristic.
'8'	Enable sequential read of PRU Dynamic Parameter characteristic.
'9'	Disable sequential read of PRU Dynamic Parameter characteristic.

The above list is prompted to Terminal emulator when 'h' is entered in the app.

To connect to the PRU device, select a specific device by pressing 'z' + device number listed along with the advertising report in the terminal and send a connection request to the device ('c') when the device is advertising.
PTU uses WPT service handle from advertising packet for quick PRU discovery instead of the classic device discovery procedure.

When connection is established, PTU automatically initiates a basic information exchange procedure, i.e. sends a read request to PRU static parameter characteristic, write request to PTU static parameter characteristic and enables notification of Alert characteristic. Each second a read request to PRU Dynamic Parameter characteristic is sent. User LED is turned on indicating the connection state.

Use ‘6’ and ‘7’ commands to send Enable and Disable Charging commands to the PRU control characteristic. PRU device will indicate charging with a blue led.

BLE Wireless Power Receiver Project Description

The project demonstrates the BLE component functionality configured as PRU.

For operation the example project uses two callback functions: AppCallBack() and WptsCallBack(). One callback function (AppCallBack()) is required to receive generic events from BLE Stack. CyBle_GappStartAdvertisement() API is called after CYBLE_EVT_STACK_ON event to start advertising with the packet shown in **Figure 10**. WptsCallBack() callback is required for receiving the events from the WPTS Service.

To start the example project operation, build it and program into baseboard of development kit. Right after the startup, the BLE, ADC, UART, and ISR components are initialized. After the initialization the BLE component begins its operation that can be seen on the RGB LED which starts blinking with a green color (blinking with blue of white color when charging is enabled). This indicates that the device has started advertising. After 180 seconds timeout, if no central device has been connected and charging is not enabled by PTU, the PRU stops advertising, a red LED is turned on indicating the disconnection state and the system enters into the hibernate mode. Press the mechanical button on CY8CKIT-042 BLE (SW2) to wake up the system and start re-advertising.

When a Client has connected successfully, both red and green LEDs are turned off. When a Client enables charging, a blue LED turns on. Blinking blue LED indicates charging with parallel advertising in the disconnected state.

The project measures DC voltage and current at the output of the rectifier, voltage and current at charge/battery port, die temperature and writes the result to the PRU Dynamic Parameter characteristic. Possibilities to measure Vrect, Irect, Vout, Iout are shown for reference purpose only and require hardware design. Meanwhile these inputs could be connected to resistive dividers to simulate charging hardware. White LED color indicates Over Voltage (when Vrect is greater than PRU_VRECT_MAX), Over Current (when Irect is greater than PRU_IRECT_MAX) or Over Temperature (when measured temperature is greater than PRU_OVER_TEMP_LEVEL) alert condition.

The measurement interval value is set to 1 second for voltage and currents. Temperature is measured rarely.

Additionally, this project implements Battery Service. For battery level measurements J2 pin P3[0] should be connected to J3 pin VREF. For measurements details and instructions on how to use this service, refer to BLE_Battery_Level example project datasheet.

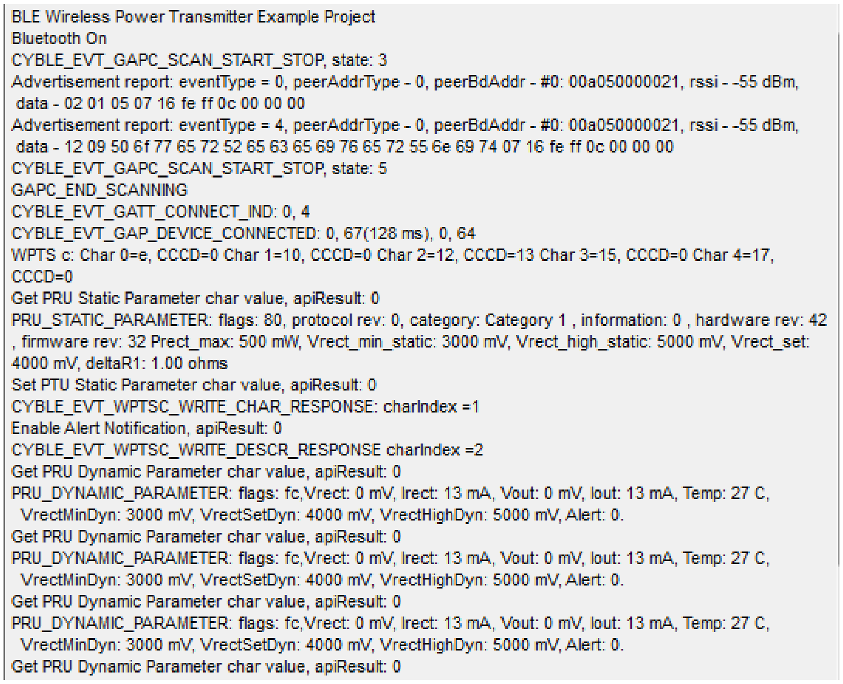
Expected Results

The BLE Wireless Power Transmitter project is intended to work in pair with BLE Wireless Power Receiver. Two projects send log messages through UART. After starting, the PTU project logs Advertising and Scan response reports from PRU, for example:

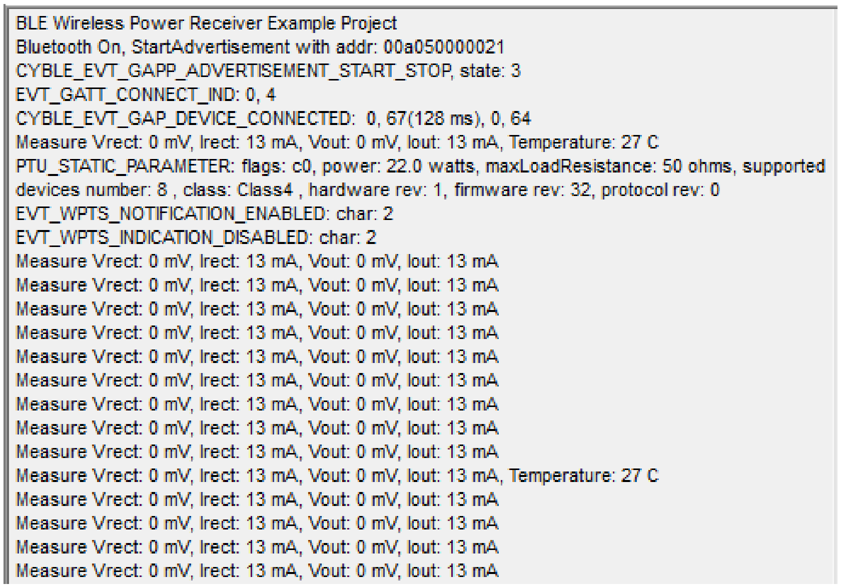
Advertisement report: eventType = 0, peerAddrType - 0, peerBdAddr - #0: 00a050000021, rssi - -58 dBm, data - 02 01 05 07 16 fe ff 0c 00 00 00

where #0 is a sequence number of the PRU device. Use this number after ‘z’ command to select required PRU if multiple PRU devices are available. Press ‘c’ to connect to PRU. Observe initialization procedure. Press ‘6’ to start charging. Blue LED on PRU indicates charging. After 100 sec PRU sends simulated Alert Notification with Complete Charge status and switches off the blue LED.

The example log is shown below:



```
BLE Wireless Power Transmitter Example Project
Bluetooth On
CYBLE_EVT_GAPC_SCAN_START_STOP, state: 3
Advertisement report: eventType = 0, peerAddrType - 0, peerBdAddr - #0: 00a050000021, rssi - -55 dBm,
 data - 02 01 05 07 16 fe ff 0c 00 00 00
Advertisement report: eventType = 4, peerAddrType - 0, peerBdAddr - #0: 00a050000021, rssi - -55 dBm,
 data - 12 09 50 6f 77 65 72 52 65 63 65 69 76 65 72 55 6e 69 74 07 16 fe ff 0c 00 00 00
CYBLE_EVT_GAPC_SCAN_START_STOP, state: 5
GAPC_END_SCANNING
CYBLE_EVT_GATT_CONNECT_IND: 0, 4
CYBLE_EVT_GAP_DEVICE_CONNECTED: 0, 67(128 ms), 0, 64
WPTS c: Char 0=e, CCCD=0 Char 1=10, CCCD=0 Char 2=12, CCCD=13 Char 3=15, CCCD=0 Char 4=17,
CCCD=0
Get PRU Static Parameter char value, apiResult: 0
PRU_STATIC_PARAMETER: flags: 80, protocol rev: 0, category: Category 1 , information: 0 , hardware rev: 42
, firmware rev: 32 Prect_max: 500 mW, Vrect_min_static: 3000 mV, Vrect_high_static: 5000 mV, Vrect_set:
4000 mV, deltaR1: 1.00 ohms
Set PTU Static Parameter char value, apiResult: 0
CYBLE_EVT_WPTSC_WRITE_CHAR_RESPONSE: charIndex =1
Enable Alert Notification, apiResult: 0
CYBLE_EVT_WPTSC_WRITE_DESCR_RESPONSE charIndex =2
Get PRU Dynamic Parameter char value, apiResult: 0
PRU_DYNAMIC_PARAMETER: flags: fc,Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA, Temp: 27 C,
  VrectMinDyn: 3000 mV, VrectSetDyn: 4000 mV, VrectHighDyn: 5000 mV, Alert: 0.
Get PRU Dynamic Parameter char value, apiResult: 0
PRU_DYNAMIC_PARAMETER: flags: fc,Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA, Temp: 27 C,
  VrectMinDyn: 3000 mV, VrectSetDyn: 4000 mV, VrectHighDyn: 5000 mV, Alert: 0.
Get PRU Dynamic Parameter char value, apiResult: 0
PRU_DYNAMIC_PARAMETER: flags: fc,Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA, Temp: 27 C,
  VrectMinDyn: 3000 mV, VrectSetDyn: 4000 mV, VrectHighDyn: 5000 mV, Alert: 0.
Get PRU Dynamic Parameter char value, apiResult: 0
```

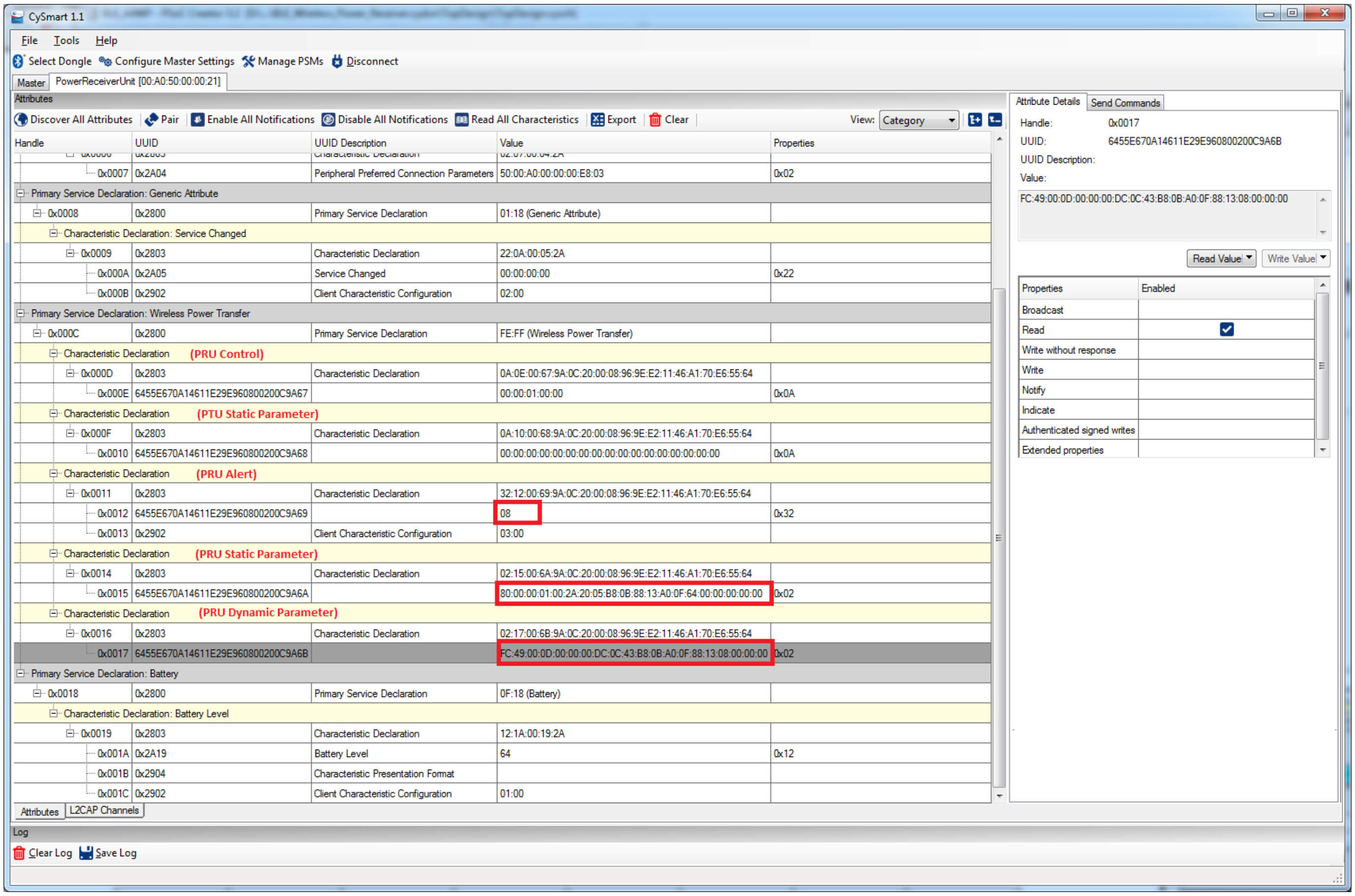


```
BLE Wireless Power Receiver Example Project
Bluetooth On, StartAdvertisement with addr: 00a050000021
CYBLE_EVT_GAPP_ADVERTISEMENT_START_STOP, state: 3
EVT_GATT_CONNECT_IND: 0, 4
CYBLE_EVT_GAP_DEVICE_CONNECTED:  0, 67(128 ms), 0, 64
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA, Temperature: 27 C
PTU_STATIC_PARAMETER: flags: c0, power: 22.0 watts, maxLoadResistance: 50 ohms, supported
devices number: 8 , class: Class4 , hardware rev: 1, firmware rev: 32, protocol rev: 0
EVT_WPTS_NOTIFICATION_ENABLED: char: 2
EVT_WPTS_INDICATION_DISABLED: char: 2
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA, Temperature: 27 C
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
Measure Vrect: 0 mV, Irect: 13 mA, Vout: 0 mV, Iout: 13 mA
```

You can use CySmart app on a Windows PC, Android or iOS BLE-compatible device as Client for connection to PRU.

To use CySmart Windows application as Client:

- Connect the CySmart BLE dongle to a USB port on the PC.
- Launch CySmart app and select connected dongle in dialog window.
- Reset development kit to start advertising by pressing SW1 button.
- Click **Start Scan** button to discover available devices.
- Select **PowerReceiverUnit** in the list of available devices and connect to it.
- Click **Discover All Attributes**, then **Enable All Notifications,** and finally **Read All Characteristics**. Observe received characteristic values.
- Select **PRU Dynamic Parameter** characteristic and press **Read Value** to observe measured values. Refer to A4WP Wireless Power Transfer System Baseline System Specification for details on characteristic structure.



Handle	UUID	UUID Description	Value	Properties
0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A	
0x0007	0x2A04	Peripheral Preferred Connection Parameters	50:00:A0:00:00:00:E8:03	0x02
Primary Service Declaration: Generic Attribute				
0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)	
Characteristic Declaration: Service Changed				
0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A	
0x000A	0x2A05	Service Changed	00:00:00:00	0x22
0x000B	0x2902	Client Characteristic Configuration	02:00	
Primary Service Declaration: Wireless Power Transfer				
0x000C	0x2800	Primary Service Declaration	FE:FF (Wireless Power Transfer)	
Characteristic Declaration (PRU Control)				
0x000D	0x2803	Characteristic Declaration	0A:0E:00:67:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x000E	6455E670A14611E29E960800200C9A67		00:00:01:00:00	0x0A
Characteristic Declaration (PTU Static Parameter)				
0x000F	0x2803	Characteristic Declaration	0A:10:00:68:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x0010	6455E670A14611E29E960800200C9A68		00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00	0x0A
Characteristic Declaration (PRU Alert)				
0x0011	0x2803	Characteristic Declaration	32:12:00:69:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x0012	6455E670A14611E29E960800200C9A69		08	0x32
0x0013	0x2902	Client Characteristic Configuration	03:00	
Characteristic Declaration (PRU Static Parameter)				
0x0014	0x2803	Characteristic Declaration	02:15:00:6A:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x0015	6455E670A14611E29E960800200C9A6A		80:00:00:01:00:2A:20:05:B8:0B:88:13:A0:0F:64:00:00:00:00:00	0x02
Characteristic Declaration (PRU Dynamic Parameter)				
0x0016	0x2803	Characteristic Declaration	02:17:00:6B:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x0017	6455E670A14611E29E960800200C9A6B		FC:49:00:0D:00:00:00:DC:0C:43:B8:0B:A0:0F:88:13:08:00:00:00	0x02
Primary Service Declaration: Battery				
0x0018	0x2800	Primary Service Declaration	0F:18 (Battery)	
Characteristic Declaration: Battery Level				
0x0019	0x2803	Characteristic Declaration	12:1A:00:19:2A	
0x001A	0x2A19	Battery Level	64	0x12
0x001B	0x2904	Characteristic Presentation Format		
0x001C	0x2902	Client Characteristic Configuration	01:00	

Handle: 0x0017

UUID: 6455E670A14611E29E960800200C9A6B

UUID Description:

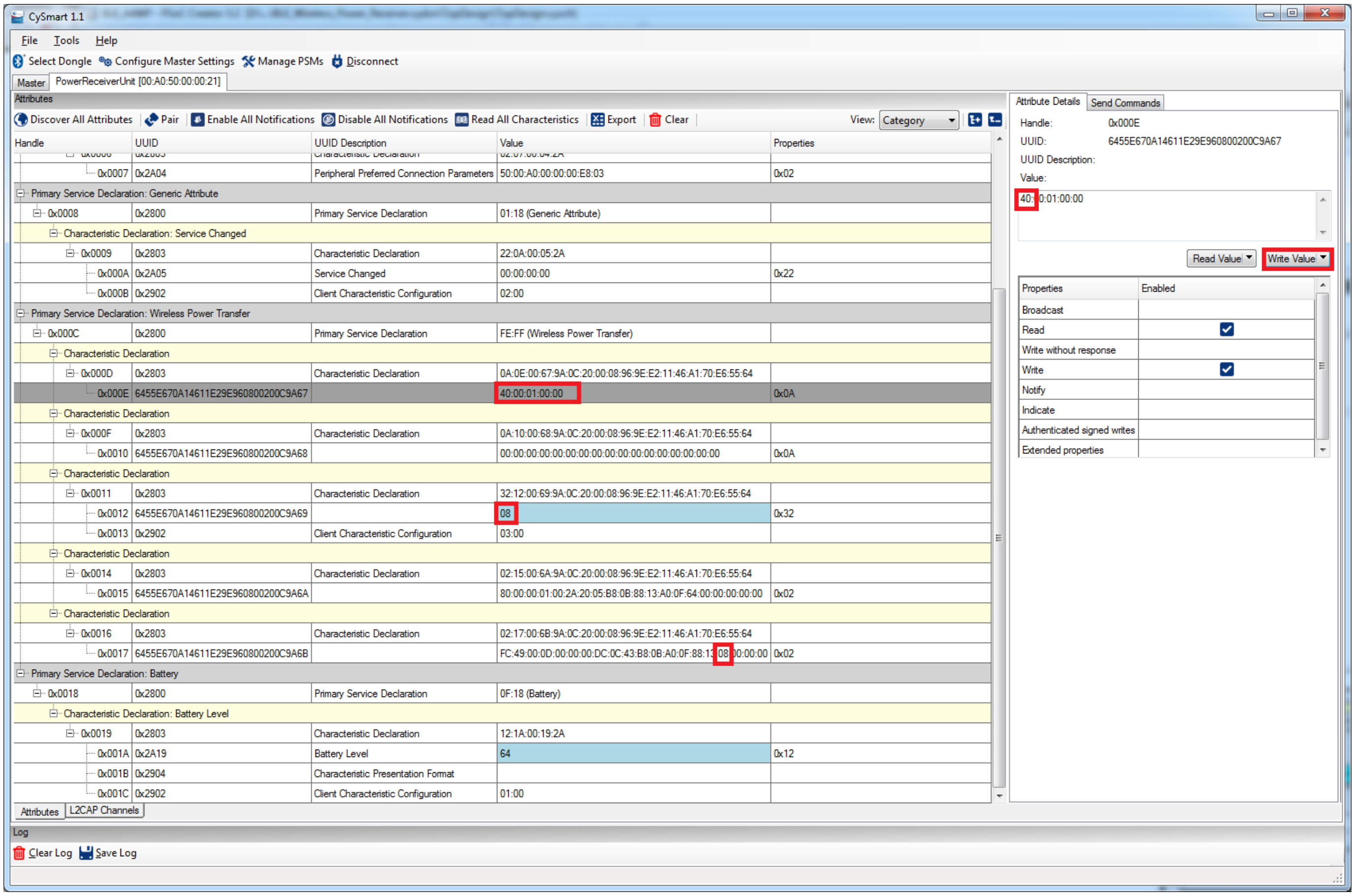
Value:

FC:49:00:0D:00:00:00:DC:0C:43:B8:0B:A0:0F:88:13:08:00:00:00

Properties	Enabled
Broadcast	
Read	☑
Write without response	
Write	
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

Figure 14. CySmart Windows app

- Select **PRU Control** characteristic, write 0x40 value (charge enable) to the first byte and press **Write Byte**. Observe that a blue LED indicates that the command is received in PRU device.
- After 100 sec observe Notification of **Alert** characteristic received with a value 0x08 (Charge Complete). Read **PRU Dynamic Parameter** characteristic and observe the same value in **PRU Alert** position (17th byte):



Handle	UUID	UUID Description	Value	Properties
0x0006	0x2803	Characteristic Declaration	02:07:00:04:2A	
0x0007	0x2A04	Peripheral Preferred Connection Parameters	50:00:A0:00:00:00:E8:03	0x02
Primary Service Declaration: Generic Attribute				
0x0008	0x2800	Primary Service Declaration	01:18 (Generic Attribute)	
Characteristic Declaration: Service Changed				
0x0009	0x2803	Characteristic Declaration	22:0A:00:05:2A	
0x000A	0x2A05	Service Changed	00:00:00:00	0x22
0x000B	0x2902	Client Characteristic Configuration	02:00	
Primary Service Declaration: Wireless Power Transfer				
0x000C	0x2800	Primary Service Declaration	FE:FF (Wireless Power Transfer)	
Characteristic Declaration				
0x000D	0x2803	Characteristic Declaration	0A:0E:00:67:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x000E	6455E670A14611E29E960800200C9A67		40:00:01:00:00	0x0A
Characteristic Declaration				
0x000F	0x2803	Characteristic Declaration	0A:10:00:68:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x0010	6455E670A14611E29E960800200C9A68		00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00	0x0A
Characteristic Declaration				
0x0011	0x2803	Characteristic Declaration	32:12:00:69:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x0012	6455E670A14611E29E960800200C9A69		08	0x32
0x0013	0x2902	Client Characteristic Configuration	03:00	
Characteristic Declaration				
0x0014	0x2803	Characteristic Declaration	02:15:00:6A:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x0015	6455E670A14611E29E960800200C9A6A		80:00:00:01:00:2A:20:05:B8:0B:88:13:A0:0F:64:00:00:00:00:00	0x02
Characteristic Declaration				
0x0016	0x2803	Characteristic Declaration	02:17:00:6B:9A:0C:20:00:08:96:9E:E2:11:46:A1:70:E6:55:64	
0x0017	6455E670A14611E29E960800200C9A6B		FC:49:00:0D:00:00:00:DC:0C:43:B8:0B:A0:0F:88:13:08:00:00:00	0x02
Primary Service Declaration: Battery				
0x0018	0x2800	Primary Service Declaration	0F:18 (Battery)	
Characteristic Declaration: Battery Level				
0x0019	0x2803	Characteristic Declaration	12:1A:00:19:2A	
0x001A	0x2A19	Battery Level	64	0x12
0x001B	0x2904	Characteristic Presentation Format		
0x001C	0x2902	Client Characteristic Configuration	01:00	

Handle: 0x000E
UUID: 6455E670A14611E29E960800200C9A67
UUID Description:
Value:
40:00:01:00:00

Properties	Enabled
Broadcast	
Read	☑
Write without response	
Write	☑
Notify	
Indicate	
Authenticated signed writes	
Extended properties	

Figure 15. CySmart Windows app

If you have problems with usage of CySmart app, refer to CySmart User Guide.

CySmart mobile app does not have WPTS service imlementation, but it still could be used in the GATT DB mode.

- Launch CySmart mobile app (Android/iOS), and swipe down to refresh the list of found BLE devices.
- Connect to “PowerReceiverUnit” device and open Unknown Service in GATT DB.
- Read the last unknown characteristic (**PRU Dynamic Parameter**) and notice the measured values.

Cypress Semiconductor Corporation, 2009-2016. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.