# RW BLE Wireless Power Transfer System Profile Interface Specification

Interface Specification

RW-BLE-PRF-WPTP-IS

Version 1.0

2016-07-12

RivieraWaves
http://www.rivierawaves.com/

# Revision History

| Version | Date | Revision Description | Author |
|---|---|---|---|
| 0.1 | November 14th 2016 | Initial draft | VF |
| 0.9 | December 5th 2016 | Major re-write to include MSCs, API parameter description, add diagrams, Device Discovery and clean-up | GF |
| 1.0 | December 7th 2016 | First Release Version – Updated to match new Cmd structure for the API. | GF |

## Table of Contents

# 1 Overview

## 1.1 Document Overview

This document describes the non-standard interface of the RivieraWaves (RW) Bluetooth Low Energy (BLE) Wireless Power Transfer System Profile (WPTP) implementation. Throughout this document, the interface messages will be referred to as API messages for the profile block(s).

Their description will include their usage and reason for implementation to provide the user and developer a better understanding, to assist in the API in a full profile compliant higher application.

## 1.2 Introduction to Wireless Power Transfer System

This document describes the BLE profile implementation to support flexibly coupled wireless power transfer (WPT) systems. These systems are used for charging mobile devices. The stationary device usually have a mains connection and serve as power source called in this document the **Power Transmitting Unit** (PTU). The mobile device is usually a recipient of the charge power and is called **Power Receiving Unit** (PTU). The Profile provides a communication protocol to ensure interoperability.

The Alliance for Wireless Power (A4WP) WPT system was standardized to transfers power from a single Power Transmitter Unit (PTU) to one or more Power Receiver Units (PRU's.). Up to eight devices can be powered from a single PTU depending on transmitter and receiver geometry and power levels. The Bluetooth Low Energy (BLE) link in the A4WP WPT system is intended for control of power levels, identification of valid loads and protection of non-compliant devices.

Figure 1 below show one possibly configuration of a WPT system consisting if a single PTU and 5 PTUs, each of which is being actively charged and also maintaining an active BT LE link.

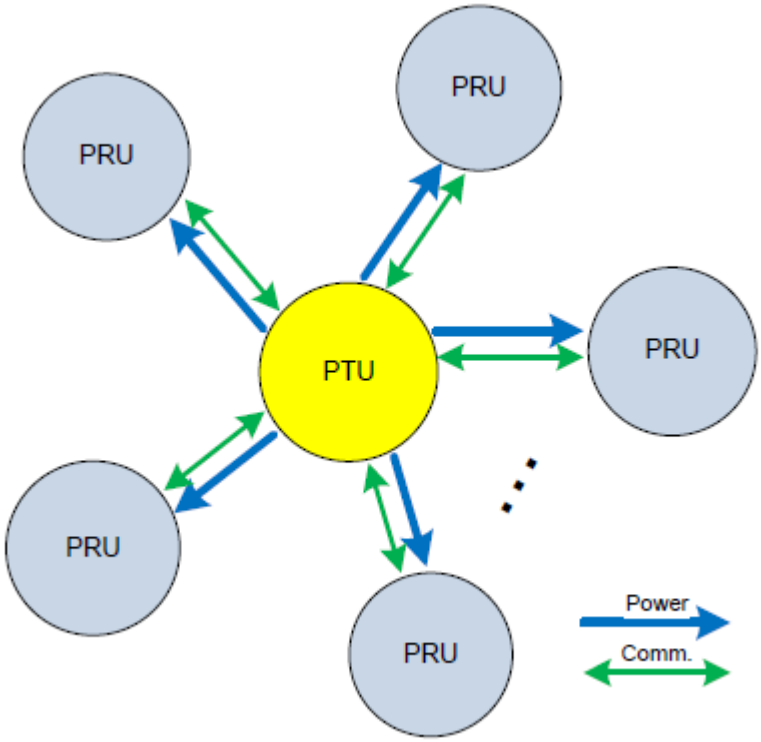


**Figure 1 - A4WP WPTP Profile PTU / PRU Network ( with 5 PRUs connected)**

## 1.3 BLE Wireless Power Transfer System Profile Overview

The WPTP enables a collector device to connect and interact with Wireless Power Transfer System Device sharing its sensor's data.

This service has been implemented as a profile. Within this profile, two roles can be supported: Server role (WPTS) and Client role (WPTC). The Client must support the GAP Central Role and the Server must support the GAP Peripheral role. The profile requires a connection to be established between the two devices for its functionality.

The Alliance for Wireless Power (A4WP) documents present different use cases for this profile, their GATT, GAP and security, mandatory and optional requirements. The Wireless Power Transfer System Profile specifications have been adopted by the Rezence Alliance for Wireless Power on November 5th 2014 ([1]). Related Test Specifications have been created and are referenced in ([2]).

The profile is implemented in the RW-BLE software stack as two tasks, one for each role. Each task has an API determined after the study of the profile specifications and test specifications, and it is considered to be minimalistic and designed for a future application which would combine the profile functionality with the device connectivity and security procedures.

The main focus of the Profile APIs is to abstract the user from the GATT interface in the system. It is expected that the final application will use the existing GAP interface to establish and manage device discovery and connections.

# 2 Wireless Power Transfer System Characteristics

The Wireless Power Transfer System (WPTS) supports the set of characteristics shown below in the Table 1. The details of these characteristics are detailed in the following sub-sections. The values for most of these characteristic are assumed to be held in the Application (not present in the ATT-DB). However, the PRU_STATIC characteristic value and the CCC for the Alert Characteristic and maintained in the ATT-DB.

| Characteristic Name | Data Direction | Characteristic Enumerator | Properties | Description |
|---|---|---|---|---|
| **PRU Static Parameter** | **PTU<-PRU** | **WPTS_IDX_PRU_STATIC_CHAR** | Read | Contains static characteristics of the PRU. PTU initiates read when device connects (can be more) |
| **PTU Static Parameter** | **PTU->PRU** | **WPTS_IDX_PTU_STATIC_CHAR** | Write/Read | Contains static characteristics of the PTU. PTU initiates write when new device connects. |
| **PRU Dynamic Parameter** | **PTU<-PRU** | **WPTS_IDX_PRU_DYNAMIC_CHAR** | Read | Contains dynamic characteristics of the PRU. PTU initiates read from each device. |
| **PRU Control** | **PTU->PRU** | **WPTS_IDX_PRU_CONTROL_CHAR** | Write/Read | PRU ON/OFF control. PTU initiates write when command needs to be sent |
| **PRU Alert** | **PTU<-PRU** | **WPTS_IDX_PRU_ALERT_CHAR** | Read/Notify/ Indicate | Notifies the PTU of overvoltage, over-current, over-temperature and self-protection conditions of the PRU. |

**Table 1 – Enumerated list of WPTS Characteristics - *enum wpts_att_list***

### 2.1.1 WPTS_IDX_PTU_STATIC_CHAR

This characteristic contains the static information about the PTU. This information is written by the PTU to the characteristic in the PRU, after connection. It contains the following elements, each of which is a single octet.

- ptu power
- maximum source impedance
- maximum load resistance
- class
- hardware revision
- firmware revision
- protocol revision
- number of devices supported

The detailed encoding of these fields in the characteristics is described in A4WP specification [1].

### 2.1.2 WPTS_IDX_PRU_STATIC_CHAR

This characteristic contains the static information about the PRU. This information can only be read by the PTU, and is normally read after connection is established. It contains the following elements:

- category
- pru information
- hardware revision
- firmware revision

- protocol revision
- prect_max
- vrect_min_static
- vrect_high_static
- vrect_set
- delta_r1_value

### 2.1.3 WPTS_IDX_PRU_DYNAMIC_CHAR

This characteristic contains the dynamic information about the PRU, it contains measurement data with values that change during the charging process on the PRU. This information can only be read by the PTU, and is normally read after connection is established and periodically after that (max 250ms interval). It contains the following elements:

- Optional fields validity
- vrect
- irect
- vout
- Iout
- Temperature
- Vrect_min_dyn
- Vrect_high_dyn
- Vrect_set_dyn
- Pru alert
- Tester command

The detailed encoding of these fields in the characteristics is described in A4WP specification [1].

### 2.1.4 WPTS_IDX_PRU_CONTROL_CHAR

This characteristic is used to initiate PTU commands in the PRU. The PTU must write to this characteristic whenever it requires a state change in the PRU. This characteristic contains the following fields :

- Enables
- Permission
- Time set

The detailed encoding of these fields in the characteristics is described in A4WP specification [1].

### 2.1.5 WPTS_IDX_PRU_ALERT_CHAR

The PRU Alert characteristic enables a PTU to receive notifications/indications to inform the PTU of Over Voltage, Over Current, Over Temperature, Mode Transition etc.

It contains single octet containing a bit-field for the alerts, and an optional Bluetooth device address. The address field is only included if the mode-transition of the Alert field is non-zero.

The detailed encoding of the alert fields of this characteristic is described in A4WP specification [1].

A Client Configuration Characteristic (CCC) descriptor is associated with this characteristic, and it used to control the generation of Notifications and Indications of the value changes in this characteristic.

| Descriptor Name | Requirements | Properties |
|---|---|---|
| **PRU Alert Client Char Configuration Descr** | Mandatory as notifiy/indication to be supported. | Read/Write |

**Table 2 – Alert CCC Descriptor**

# 3 WPTP Service Discovery

The A4WP profile differs from other Bluetooth Low Energy profiles, in the manner service and characteristic discovery is performed. The GATT Service Discovery and Characteristic Discovery procedures are not explicitly required for this profile. Instead the Primary Service Handle for the WPT profile is obtained from the Advertising Data obtained by the PTU (from the Advertising data) during device discovery.

The A4WP profile specification dictates a specific order of the Characteristic definition, with each characteristic declaration, characteristic value and characteristic descriptor having a given handle offset relative to the A4WP primary service handle.

| | | |
|---|---|---|
| GATT_PRIMARY_SERVICE_UUID | Start of WPT Service | Primary Service Handle |
| GATT_CHARACTERISTIC_UUID | PRU Control Characteristic declaration | Primary Service Handle + 1 |
| WPT_CHARGING_PRU_CONTROL_UUID | PRU Control Characteristic value | Primary Service Handle + 2 |
| GATT_CHARACTERISTIC_UUID | PTU Static Parameter Characteristic declaration | Primary Service Handle + 3 |
| WPT_CHARGING_PTU_STATIC_UUID | PTU Static Parameter Characteristic value | Primary Service Handle + 4 |
| GATT_CHARACTERISTIC_UUID | PRU Alert Parameter Characteristic declaration | Primary Service Handle + 5 |
| WPT_CHARGING_PRU_ALERT_UUID | PRU Alert Parameter Characteristic value | Primary Service Handle + 6 |
| CCC _UUID | Client Characteristic Configuration UUID for PRU Alert | Primary Service Handle + 7 |
| GATT_CHARACTERISTIC_UUID | PRU Static Parameter Characteristic declaration | Primary Service Handle + 8 |
| WPT_CHARGING_PRU_STATIC_UUID | PRU Static Parameter Characteristic value | Primary Service Handle + 9 |
| GATT_CHARACTERISTIC_UUID | PRU Dynamic Parameter Characteristic declaration | Primary Service Handle + 10 |
| WPT_CHARGING_PRU_DYNAMIC_UUID | PRU Dynamic Parameter Characteristic value | Primary Service Handle + 11 |

**Table 1 - WPTP Characteristic Declarations, Values and Handles**

In this way, once the PTU receives the Primary Service handle for the WPT it can obtain all the required information about the WPT Gatt Database in the peer PRU, without the need to perform GATT Discovery. The reason for this mechanism is to shorten the time between ACL link establishment and PTU/PRU registration. A registration timer started on receipt of the first valid PRU advertisement and is stopped when the PTU writes the control characteristic to the PRU. The registration timer is 500ms and if it expires, prior to writing the control characteristic, the PTU shall terminate the connection to the PRU.

# 4 Common Fields of API

This section describes fields of the API primitives which are commonly used in the PTU and PRU. Most of these fields have equivalent fields in the characteristics, however the values and value ranges cannot be assumed to be the same. For example, the 'enables' field present in Control Characteristic is given as 3 separate fields on the API.

## 4.1 Enable_output

This value is set by the PTU in the PRU. A value of '1' allows the PRU to provide power to the load, before this it must draw less than 1.1 W. If a value is set to '0' then the PRU shall reduce its output to less than 1.1 W, within 1.1 seconds.

1 = Enable
0 = Disable

The Enable_output maps to the Bit 7 of the Enables field of the PRU Control Characteristic.

## 4.2 Enable_charge_indicator

This value is set by the PTU in the PRU. A value of '1' allows the PRU to indicate that charging may occur, otherwise set to '0'

1 = Enable
0 = Disable

The Enable_charge_indicator maps to the Bit 6 of the Enables field of the PRU Control Characteristic.

## 4.3 Adjust_power_command

This value is set by the PTU in the PRU. It instructs the PRU to adjust its power draw to less than a specific percentage of the maximum average power of the rectifier (Prect_Max). The value contained in this field maps to the Bits 4 & 5 of the *Enables* field of the PRU Control Characteristic

| Value | Description | #define |
|---|---|---|
| 0x00 | Maximum Power | WPT_PRU_CONTROL_ADJUST_POWER_MAXIMUM |
| 0x01 | 66% * $P_{RECT\_MAX}$ | WPT_PRU_CONTROL_ADJUST_POWER_0_66_MAX |
| 0x02 | 33% * $P_{RECT\_MAX}$ | WPT_PRU_CONTROL_ADJUST_POWER_0_33_MAX |
| 0x03 | 2.5W | WPT_PRU_CONTROL_ENABLE_POWER_2_5_WATT |
| Other | UNDEFINED | |

If the PRU successfully adjusts its power it will provide a response to the PTU via the Adjust Power Response bit field of the Alert field of the PRU Dynamic Characteristic (which is read by the PTU at least every 250ms).

PTUs shall not send this command to PRUs that do not support the Adjust power command (determined by reading the information field of the PRU Static Characteristic, and contained in WPTC_RD_PRU_STATIC_RSP Event ). A PTU shall not send more than one Adjust power command to a PRU in a two second period.

## 4.4 Permission

This value is set by the PTU in the PRU. This field is used to provide the PRU with information from the PTU about what is it allowed.

| Value | Description | #define |
|---|---|---|
| 0x00 | Permitted without reason | WPT_PRU_CONTROL_PERMITTED_WITHOUT_REASON |
| 0x01 | Permitted with waiting time due to limited available power | WPT_PRU_CONTROL_PERMITTED_WITH_WAIT |

| | | |
|---|---|---|
| 0x80 | Denied due to cross connection. | WPT_PRU_CONTROL_DENIED_DUE_CROSS_CONNECTION |
| 0x81 | Denied due to limited available power | WPT_PRU_CONTROL_DENIED_DUE_LIMITED_POWER |
| 0x82 | Denied due to limited PTU Number of Devices | WPT_PRU_CONTROL_DENIED_DUE_LIMITED_NUMBER_OF_DEVICES |
| 0x83 | Denied due to limited PTU Class support | WPT_PRU_CONTROL_DENIED_DUE_LIMITED_CLASS_SUPPORT |
| 0x84 | Denied due to high temperature at PTU | WPT_PRU_CONTROL_DENIED_DUE_TO_HIGH_TEMPERATURE |
| Other | RESERVED | |

## 4.5 Time_set

This value can be set by the PTU in the PRU. It is used for cross connection checks and indicates the time in which the PRU is to create a valid load variation.

| Value | Description | #define |
|---|---|---|
| 0x00 | Do not perform time set | WPT_PRU_CONTROL_SET_TIME_DO_NOT_SET |
| 0x01 | 10ms | WPT_PRU_CONTROL_SET_TIME_10_MS |
| 0x02 | 20ms. | WPT_PRU_CONTROL_SET_TIME_20_MS |
| 0x03 | 30ms | WPT_PRU_CONTROL_SET_TIME_30_MS |
| 0x04 | 40ms | WPT_PRU_CONTROL_SET_TIME_40_MS |
| 0x05 | 50ms | WPT_PRU_CONTROL_SET_TIME_50_MS |
| 0x06 | 60ms | WPT_PRU_CONTROL_SET_TIME_60_MS |
| 0x07 | 70ms | WPT_PRU_CONTROL_SET_TIME_70_MS |
| 0x08 | 80ms | WPT_PRU_CONTROL_SET_TIME_80_MS |
| Other | RESERVED | |

The Time Set value shall be set to zero if the PRU does not support Time Set (determined from the information field of the PRU Static Characteristic, and contained in WPTC_RD_PRU_STATIC_RSP Event). For the PRU that supports Time Set, the PTU shall use a non-zero value at least once before using a Time Set value of zero.

## 4.6 ptu_power

This field indicates the power value of the PTU. The mapping of values in this field to actual power in Watts is given in Annex 1. The maximum value is 139 which is equivalent to a power of 61 watts.

## 4.7 ptu_max_source_impedance

This field is only relevant if its presence has been indicated in the “optional_fields_validity” field in the same API message. The value of this field indicates the maximum output impedance of the PA/filter in the PTU. The mapping between the values contained in this field and the PTUs maximum source impedance are given in Annex 1. The maximum value is 18 which corresponds to a source impedance of 375 ohms.

## 4.8 ptu_max_load_resistance

This field is only relevant if its presence has been indicated in the "optional_fields_validity" " field in the same API message. The value of this field indicates the maximum PTU load resistance as seen at the input to the PTU resonator. The mapping between the values contained in this field and the PTUs maximum load resistance is given in Annex 1. The maximum value is 10 corresponding to a load resistance of 55 ohms.

## 4.9 ptu_class

This field defines the class to which the PTU is assigned

| Value | Class | #define |
|---|---|---|
| 0x00 | Class 1 | WPTP_PTU_CLASS_1 |
| 0x01 | Class 2 | WPTP_PTU_CLASS_2 |
| 0x02 | Class 3 | WPTP_PTU_CLASS_3 |
| 0x03 | Class 4 | WPTP_PTU_CLASS_4 |
| 0x04 | Class 5 | WPTP_PTU_CLASS_5 |
| Other | RESERVED | |

## 4.10 Hardware_rev

The current PTU or PRU hardware revision number. The numbering system for this field is vendor proprietary.

## 4.11 Firmware_rev

The current PTU or PRU firmware revision number. The numbering system for this field is vendor proprietary.

## 4.12 Protocol_rev

This field maps to the highest A4WP specification revision to which the PTU or PRU is certified. The following numbers are used :

| Value | A4WP Revision |
|---|---|
| 0 | BSS v1.2.1 |
| 1 | BSS v1.3 |

## 4.13 pru_category

The PRU is assigned a category number which indicates the maximum power it can draw from the PTU. *( note : Some confusion here in the A4WP spec where Category 6&7 are also sometimes mentioned – but not defined as valid values in the GATT characteristic).*

| Value | Category | Power | #define |
|---|---|---|---|
| 0x00 | UNDEFINED | N/A | WPTS_CATEGORY_UNDEFINED |
| 0x01 | Category 1 | TBD | WPTS_CATEGORY_1 |
| 0x02 | Category 2 | 3.5w | WPTS_CATEGORY_2 |
| 0x03 | Category 3 | 6.5w | WPTS_CATEGORY_3 |
| 0x04 | Category 4 | 13w | WPTS_CATEGORY_4 |
| 0x05 | Category 5 | 25w | WPTS_CATEGORY_5 |

| | | | |
|---|---|---|---|
| Other | UNDEFINED | | |

## 4.14 pru_information

This field indicates the optional capabilities supported in the PRU. It is interpreted as a bit-field as shown below. The PTU receiving this information should not invoke a command on a feature which not supported by the peer PRU.

| Bit | Description | Values |
|---|---|---|
| 7 | NFC Receiver Supported | '1' NFC supported, Otherwise bit is set to '0' |
| 6 | Separate BTLE radio | '1' the BTLE radio is a separate entity in the PRU. |
| 5 | Power Control Algorithm | '1' if the preference is for Max System. Otherwise, this bit is set to '0' for VRECT_MIN_ERROR. |
| 4 | Adjust power capability | set to '1' if the PRU supports the Adjust power capability. |
| 3 | Charge Complete Connected | set to '1' if the PRU supports Charge Complete Connected Mode |
| 2 | PTU Test Mode | set to '1' if the PRU is capable of operating as a PRU simulator and is requesting that the PTU enter PTU Test Mode. |
| 1,0 | UNDEFINED | |

## 4.15 prect_max

This is an 8 bit value, indicating the maximum rectifier power of the PRU. Values are given in increments of 100mw (e.g a value of 48 indicates a max rectifier power of 4.8w )

## 4.16 vrect_min_static

A 16bit value indicating minimum initial voltage of the rectifier, given in mV.

## 4.17 vrect_high_static

A 16bit value indicating the highest initial voltage of the rectifier, given in mV.

## 4.18 Vrect_set

The voltage currently set for the rectifier, given in mV. This values has to be between the min/max values of vrect_min_static and vrect_high_static.

## 4.19 ptu_num_devices_supported

Total number of devices supported by the PTU. Max value is 8.

## 4.20 Delta_r1_value

This is a 16bit value which indicate the change in the measured resistance of a PTU resonator when a PRU, with an open-circuit PRU resonator, is placed in the center of the charge area of the PTU resonator. The value is given in increments of 0.01 ohms. This is an optional component and is only valid if its presence is indicated by "delta_r1_value_present" = 1.

## 4.21 Delta_r1_value_present

This octet indicates if the value contained in the "delta_r1_value" field is valid. If the delta_r1_value field is valid if this field contains a value = '1'

## 4.22 Vrect

This is a 16bit value for the voltage at the PRU rectifier output. Value is given in mV. . This corresponds to the 'Vrect' field of the PRU Dynamic Characteristic

## 4.23 Irect

This is a 16bit value for the current at the PRU rectifier output. Value is given in mA. . This corresponds to the 'Irect' field of the PRU Dynamic Characteristic

## 4.24 Vout

This is a 16bit value for Voltage at Charge Battery Port of the PRU. Value is given in mV. This corresponds to the optional 'Vout' field of the PRU Dynamic Characteristic. The validity of its value in determined by the 'optional_value_field' of the API message.

## 4.25 Iout

This is a 16bit value for current at Charge Battery Port of the PRU. Value is given in mA. This corresponds to the optional 'Iout' field of the PRU Dynamic Characteristic. The validity of its value in determined by the 'optional_value_field' of the API message.

## 4.26 Temperature

This is an 8bit value indicating the temperature for the PRU. The minimum temperature is -40C, which is indicated by a value of zero. The temperature is given in one degree increments, thus the maximum temperature value is +215c. This corresponds to the optional 'temperature' field of the PRU Dynamic Characteristic. The validity of its value in determined by the 'optional_value_field' of the API message.

## 4.27 Vrect_min_dyn

A 16bit value indicating minimum dynamic voltage of the rectifier, given in mV. This corresponds to the optional 'Vrect_min_dyn' field of the PRU Dynamic Characteristic. The validity of its value in determined by the 'optional_value_field' of the API message.

## 4.28 Vrect_high_dyn

The current dynamic maximum rectifier voltage desired , given in mV. This corresponds to the optional 'Vrect_high_dyn' field of the PRU Dynamic Characteristic. The validity of its value in determined by the 'optional_value_field' of the API message.

## 4.29 Vrect_set_dyn

A 16bit value indicating the desired voltage for the rectifier, given in mV. This values has to be between the min/max values of Vrect_min_dyn and Vrect_high_dyn. This corresponds to the optional 'Vrect_set_dyn' field of the PRU Dynamic Characteristic. The validity of its value in determined by the 'optional_value_field' of the API message.

## 4.30 pru_alert

This is an 8bit bit-field which is used to transfer Alert conditions from the PRU to the PTU.

| Bit | Name | #define | Description |
|---|---|---|---|
| 7 | PRU Over-voltage | WPTP_ALERT_OVER_VOLTAGE | set to '1', indicates that VRECT at the PRU has exceeded the OVP limit |
| 6 | PRU Over-current | WPTP_ALERT_OVER_CURRENT | Set to '1', indicates that IRECT at the PRU has exceeded the PRU's current limit. |
| 5 | PRU Over-temperature | WPTP_ALERT_OVER_TEMP | This bit, when set, indicates that measured temperature at the PRU |

| | | | |
|---|---|---|---|
| | | | has exceeded the PRU's temperature. |
| 4 | PRU Self Protection | WPTP_ALERT_PRU_SELF_PROTECTION | This bit, when set, indicates that the PRU is protecting itself by reducing power to its load. |
| 3 | PRU Charge Complete | WPTP_ALERT_CHARGE_COMPLETE | This bit, when set, indicates that the PRU does not require charging. |
| 2 | PRU Wired Charger Detect | WPTP_ALERT_WIRED_CHARGER_DETECT | This bit, when set, indicates that the PRU is powered by external wired power |
| 1 | PRU Charge Port | WPTP_ALERT_PRU_CHARGE_PORT | set to '1' to indicate that the PRU charge port output is activated |
| 0 | Adjust Power Response | WPTP_ALERT_ADJUST_POWER_RESPONSE | The Adjust Power Response bit is used to indicate whether or not the PRU has limited its power draw in response to the Adjust power command. If the PRU does limit its power draw according to the PTU Adjust power command, it shall set the 'Adjust Power Response' bit to '1' |

This maps to values included in both the PRU Dynamic Characteristic and the PRU Alert Characteristic. For Alert indications sent to the Application the bottom two bits of this field are not used.

## 4.31 Mode_Transition

This field is used to inform the PTU of a Mode Transition in the PRU. A Mode Transition set to a non-zero value indicates to the PTU the duration of the pending Mode Transition procedure

00 -- No Mode Transition
01 – 2s mode transition time limit
02 – 3s mode transition time limit
03 – 6s mode transition time limit

This field maps to the Mode_Transition bits (Bit 0-1) of the Alert Characteristic value.

## 4.32 Tester Command

This field is optional and shall only be used when the PTU has previously been placed in test mode during device registration:

| Command Number | Description |
|---|---|
| 0x00 | Normal operation power control |
| 0x01 | Increase I*TX* by one step |
| 0x02 | Decrease I*TX* by one step |
| 0x03 | ITX_COIL current shall not be adjusted |
| Other Values | Reserved |

# 5 Structures on API

This section describes the structures which are used on the API. These structures correspond to the format of the characteristic values used in the WTP profile. They are used as members of unions on the API.

## 5.1 PTU Static

This structure is used to transfer information about the PTU Static Information across the API. This is used when the PTU writes the PTU Static Parameters to the PRU. The 'optional_fields_validity' is used to determine if the 'ptu_max_source_impedance' and 'ptu_max_load_resistance' values are valid in this API message and should be included in the corresponding characteristic write.

In C format - *struct wptc_ptu_static* or *struct wpts_ptu_static*

| Type | Parameters | Description |
|---|---|---|
| uint8_t | *optional_fields_validity* | Defines which fields are valid:<br>**Bit 7 = 1**: Max Impedance – is valid<br>**Bit 6 = 1**: Max Resistance – is valid |
| uint8_t | *ptu_power* | Power of PTU (see conversion table in Annex 1) |
| uint8_t | *ptu_max_source_impedance* | Maximum source impedance of the PTU (see conversion table in Annex 1) |
| uint8_t | *ptu_max_load_resistance* | Maximum load resistance of the PTU (see conversion table in Annex 1) |
| uint8_t | *ptu_class* | PTU class, see section 4.9 for allowed values |
| uint8_t | *hardware_rev* | Revision of the PTU HW |
| uint8_t | *firmware_rev* | Revision of the PTU SW |
| uint8_t | *protocol_rev* | A4WP Supported Revision., see section 4.12 |
| uint8_t | *ptu_num_devices_supported* | Maximum Number of Devices |

.

## 5.2 PRU Static

In C format - *struct wptc_pru_static* or *struct wpts_pru_static or struct wpts_db_cfg*

| Type | Parameters | Description |
|---|---|---|
| uint8_t | *protocol_rev* | A4WP Supported Revision, see section 4.12 |
| uint8_t | *pru_category* | Category of PRU, see section 4.13 for valid values. |
| uint8_t | *pru_information* | Capabilities of PRU (bit field). see section 4.14 for description of bit field |
| uint8_t | *hardware_rev* | Revision of the PRU HW |
| uint8_t | *firmware_rev* | Revision of the PRU SW |
| uint8_t | *prect_max* | PRECT_MAX of the PRU. The value is in increments of 100mW |
| uint16_t | *vrect_min_static* | VRECT_MIN (static, first estimate). The value is in mV. |
| uint16_t | *vrect_high_static* | VRECT_HIGH (static, first estimate). The value is in mV. |
| uint16_t | *vrect_set* | VRECT_SET. The value is in mV. |
| uint8_t | *delta_r1_value_present* | Indicates if the "delta_r1_value" field contains a valid value. |
| uint16_t | *delta_r1_value* | Delta R1 caused by PRU. The PRU may report its Delta R1, if included, in increments of 0.01 ohms. |

As the 'delta_r1_value' is optional in the characteristic, the 'delta_r1_value_present' field indicates if it contains a valid value.

## 5.3 PRU Dynamic

This structure is used to transfer PRU Dynamic Parameters across the API.

In C format - *struct wptc_pru_dynamic* or *struct wpts_pru_dynamic*

| Type | Parameters | Description |
|---|---|---|
| uint8_t | optional_fields_validity | Defines which fields of the API event contain valid values. If an optional field is not indicated as valid in the parameter, its value should be ignored. |
| uint16_t | vrect | DC voltage at the output of the rectifier. mV |
| uint16_t | irect | DC current at the output of the rectifier. mA |
| *uint16_t* | *vout* | *Voltage at charge/battery port. mV (optional field)* |
| *uint16_t* | *Iout* | *Current at charge/battery port. mA (optional field)* |
| *uint8_t* | *temperature* | *Temperature of PRU. Deg C from -40C (optional field)* |

| | | |
|---|---|---|
| *uint16_t* | *vrect_min_dyn* | *The current dynamic minimum rectifier voltage desired. mV (optional field)* |
| *uint16_t* | *vrect_set_dyn* | *Desired VRECT (dynamic value). mV (optional field)* |
| *uint16_t* | *vrect_high_dyn* | *The current dynamic maximum rectifier voltage desired. mV (optional field)* |
| uint8_t | pru_alert | This is a bit-field which is used to transfer Alert conditions from the PRU to the PTU. See section 4.30 for a complete description of this field and range of values allowed. |
| uint8_t | tester_command | PTU Test Mode Command |

The 'optional_fields_validity' is a bit-field indicates which of the fields in the API message are valid. If the bit is set the corresponding field is valid.

| Bit No. | Field Validity | Values |
|---|---|---|
| 7 | *Vout* | ('1' Present, '0' Absent ) |
| 6 | *Iout* | ('1' Present, '0' Absent ) |
| 5 | *Temperature* | ('1' Present, '0' Absent ) |
| 4 | *Vrect_min_dyn* | ('1' Present, '0' Absent ) |
| 3 | *Vrect_set_dyn* | ('1' Present, '0' Absent ) |
| 2 | *Vrect_high_dyn* | ('1' Present, '0' Absent ) |
| 1 | *NOT USED* | N/A |
| 0 | *NOT USED* | N/A |

## 5.4 PRU Control

This structure is used to control the operation of the PRU.

In C format - *struct wptc_pru_control* or *struct wpts_pru_control*

| Type | Parameters | Description |
|---|---|---|
| uint8_t | *enable_pru_output* | 0 – PRU Turn Off, 1 – PRU Turn On |
| uint8_t | *enable_charge_indication* | 0 – No Charge Indication, 1 – Charge Indication |
| uint8_t | *adjust_power_command* | Instructs the PRU to adjust its power draw to less than a specific percentage of the maximum average power of the rectifier (Prect_Max) See Section 4.3 for full description of valid range and #define for this field |
| uint8_t | *permission* | This field is used to provide the PRU with information from the PTU about what is it allowed. See section 4.4 for description of this field and values allowed. |
| uint8_t | *time_set* | Time in which PRU is to create a valid load. For range of values see section 4.5 |

## 5.5 Mode Transition

This structure is used to transfer information about alert conditions with mode transition. The 'alert' field contains a bit field which indicates alert conditions in the PRU

In C format - *struct wptc_pru_mode_transition*

| Type | Parameters | Description / Values |
|---|---|---|
| uin8_t | *alert* | Alert Condition in the PRU. Only bits 7-2 of the alert field are used in this structure. |
| uint_8 | *mode_transition* | 0x00 --- No Mode Transition<br>0x01 --- 2 sec Mode Transition Time Limit<br>0x02 --- 3 sec Mode Transition Time Limit<br>0x03 --- 6 sec Mode Transition Time Limit<br>0x04-0xFF -- UNDEFINED |
| uint_8 | *addr[6]* | A Bluetooth device address which is to be used by the PTU/PTU when they attempt to reconnect. |

# 6 PTU API

## 6.1 PTU API Messages

This section overviews the PTU API messages which are provided in "wptc_task.h". Additional details on the fields of each message can be found in section 5 of this document.

### 6.1.1 WPTC_ENABLE_CMD

Source: TASK_APP

Destination: TASK_WPTC

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | Operation Code, filled by the profile |
| uint16_t | start_hdl | The primary service handle for the WPT service. This is normally obtained from the Advertising Data during device discovery. If the value given is '0x00' then it is assumed no Service Handle is available and service and characteristic discovery will be performed. |
| uint8 | registration | This is a flag to indicate if device registration should be initiated by the PTU. When set the Profile will automatically read the PRU_Static characteristic from the peer PRU. |

Response: WPTC_CMP_EVT

WPTC_VALUE_IND

Description: This API message is used for enabling the Collector role (PTU) of the WPTC. This message contains the Service Handle for the WPT profile which is obtained during device discovery. The registration flag will determine if the registration flow should begin automatically. If 'registration = 1', the PRUs Static Characteristic is retrieved as part of this command, and will be delivered over API using the WPTC_VALUE_IND msg.

### 6.1.2 WPTC_READ_CMD

Source: TASK_APP

Destination: TASK_WPTC

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | Set by profile – should not be used by Application |
| uint8_t | read_code | Read Code<br>0x01 - WPTC_RD_PRU_CONTROL<br>0x02 - WPTC_RD_PTU_STATIC<br>0x03 - WPTC_RD_PRU_STATIC<br>0x04 - WPTC_RD_PRU_DYNAMIC<br>0x05 - WPTC_RD_PRU_ALERT |

Response: WPTC_CMP_EVT

WPTC_VALUE_IND

Description: This message is used to request the read characteristic information from the server (PRU). The 'read_code' is used to determine which characteristic value is to be read from the PRU. The response is returned on a WPTC_VALUE_IND message.

### 6.1.3 WPTC_WRITE_CMD

Source: TASK_APP

Destination: TASK_WPTC

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | Operation Code, filled by the profile |
| uint8_t | write_code | Identifies what is to be written<br>0x06 - WPTC_WR_PRU_CONTROL<br>0x07 - WPTC_WR_PTU_STATIC<br>0x08 - WPTC_WR_ALERT_CCC |
| **union** | **value** | |
| struct wptc_pru_control | pru_ctrl | The PRU Control Information |
| struct wptc_ptu_static | ptu_static | The PTU Static Characteristic Information |
| uint16_t | alert_ccc | CCC descriptor of the Alert Characteristic |

Response: WPTC_CMP_EVT

Description :- This message is used by the PTU to write information to the PRU. The element to be written is identified by the 'write_code'. When the write is completed a WPTC_CMP_EVT will be sent to the application.

### 6.1.4 WPTC_VALUE_IND

Source: TASK_WPTC

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | attribute_code | The attribute whose value is reported<br>0x01 - WPTC_PRU_STATIC_ATT_ID<br>0x02 - WPTC_PTU_STATIC_ATT_ID<br>0x03 - WPTC_PRU_CONTROL_ATT_ID<br>0x04 - WPTC_PRU_DYNAMIC_ATT_ID<br>0x05 - WPTC_PRU_ALERT_ATT_ID<br>0x06 - WPTC_PRU_MODE_TRANSITION_ID<br>0x07 - WPTC_PRU_ALERT_CCC_ATT_ID |
| **Union** | **value** | |
| struct wptc_pru_control | pru_ctrl | The PRU Control Information |
| struct wptc_ptu_static | ptu_static | The PTU Static Characteristic Information |
| struct wptc_pru_static | pru_static | The PRU Static Characteristic Information |
| struct wptc_pru_dynamic | pru_dynamic | The PRU Dynamic Characteristic Information |
| struct wptc_pru_alert | pru_alert | A (simple) Alert from the PRU |
| struct wptc_pru_mode_transition | pru_mode_transition | A mode Transition indication from the Peer PRU |
| uint16_t | alert_ccc | CCC descriptor of the Alert Characteristic |

This message is sent to the application to indicate an attribute value. This can occur as a result of a 'WPTC_READ_CMD' or can be an unsolicited indication/notification from the PRU (ie.. ALERT indication). The 'attribute_id' identifies the contents of the message.

### 6.1.5 WPTC_CMP_EVT

Source: TASK_WPTS

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | Operation | Operation Code |
| uint8_t | Status | Status |

Description: The API message is used by the WPTC task to inform the APP that procedure is complete in the client profile and returns the status field indicating the outcome of the procedure.

The valid operation codes are the following :

| Operation Code | VALUE |
|---|---|
| WPTC_RD_PRU_CONTROL | 0x01 |
| WPTC_RD_PTU_STATIC | 0x02 |
| WPTC_RD_PRU_STATIC | 0x03 |
| WPTC_RD_PRU_DYNAMIC | 0x04 |
| WPTC_RD_PRU_ALERT | 0x05 |
| WPTC_WR_PRU_CONTROL | 0x06 |
| WPTC_WR_PTU_STATIC | 0x07 |
| WPTC_WR_ALERT_CCC | 0x08 |
| WPTC_ENABLE | 0x09 |

# 7 PRU API

## 7.1 Environment

This Implementation is single instance task.

This role should be activated in the application that a Wireless Power Transfer Service is required; the provided API is capable of sending Alarm notifications, indications, and exchanging control and configuration information with the Client. The profile implementation forward requests to Read and Write Characteristic information from the Collector to the Application and returns the Application responses back to Client. Please, refer to "wpts_task.h" for implementation of this API.

Within the WPTS task several states are defined as shown in Figure 2. The 3 state are**: FREE**, **IDLE, OP_READ** and **OP_NOTIFY,**

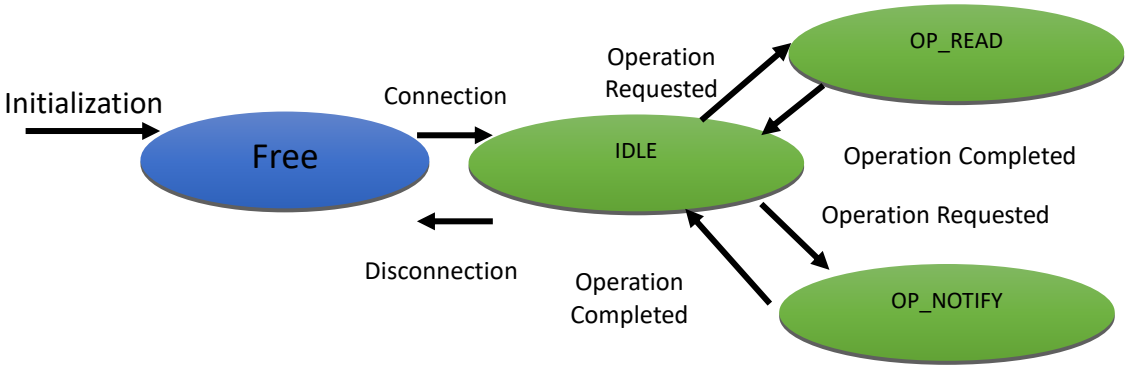


***Figure 2 :- States of the WPT Server Implementation***

## 7.2 PRU API Messages

### 7.2.1 Initialisation

During the initialization phase of the Wireless Power Transfer System Server, the memory for this task must be allocated using the message GAPM_PROFILE_TASK_ADD_CMD provided by the GAPM interface. Obtaining the start handle is important for the profile as it is used as a parameter in the advertising payload. This handle is then used by the client to access the characteristic without performing full service discovery. The start handle can be obtained in one of the 2 ways.

1/ Setting the start handle when initializing the service by calling GAPM_PROFILE_ADD_CMD with a the start handle set to a known value.

2/ Getting the start handle when initializing the service with "dynamic allocation" parameters. Application task should handle the GAPM message GAPM_PROFILE_ADDED_IND. In the handler the first parameter should be checked against the TASK_ID_WPTS and then obtaining the start_hdl that can be used.

During server initialization the PRU Static information is transferred to the profile and loaded into the ATT-DB. As this is static information it avoids this information being request from the Application during registration.

| Type | Parameters | Description |
|---|---|---|
| struct wpts_db_cfg | param | The PRU Static information which is to be written to the local ATT-DB |

Description: This API message shall be used to add one instance of the Wireless Power Transfer System Service in the database.

### 7.2.2 WPTS _ENABLE_CMD

Source: TASK_APP

Destination: TASK_WPTS

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint16_t | Conidx | Identifier for the current connection. Obtained from the GAP during connection establishment |
| uint16_t | ntf_ind_cfg | Characteristic Configuration Descriptor bit field value for a bonded device:<br>**Bit 0** : Notifications Enabled<br>**Bit 1**: Indication enabled |

Response: WPTS_CMP_EVT

Description: This API message shall be used after the connection with a peer device has been established. This starts internal state machine of the task. The connection identifier obtained during connection establishment has to passed as a parameter in 'conidx'.

If the device is bonded, the previous setting of the Alert CCC is passed into the profile using the 'ntf_ind_cfg' field.

### 7.2.1 WPTS_VALUE_IND

Source: TASK_WPTS

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | attrib_id | The attribute whose value is reported<br>0x02 - WPTC_PTU_STATIC_ATT_ID<br>0x03 - WPTC_PRU_CONTROL_ATT_ID<br>0x07 - WPTC_PRU_ALERT_CCC_ATT_ID |
| **Union** | **value** | |
| struct wpts_pru_control | pru_ctrl | The PRU Control Information |
| struct wpts_ptu_static | ptu_static | The PTU Static Characteristic Information |
| uint16_t | alert_ccc | CCC descriptor of the Alert Characteristic |

This message is sent to the application to indicate an attribute needs to be updated – due to write request from peer PTU. This can occur as a result when the peer device wants to update the PRU Control Characteristic Value or modify the CCC setting for the Alert Characteristic.

### 7.2.2 WPTS_VALUE _REQ_IND

Source: TASK_WPTS

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | Attrib_id | The attribute whose value is requested<br>*0x01 - WPTC_PRU_STATIC_ATT_ID*<br>0x02 – WPTC_PTU_STATIC_ATT_ID<br>0x03 - WPTC_PRU_CONTROL_ATT_ID<br>0x04 - WPTC_PRU_DYNAMIC_ATT_ID<br>0x05 - WPTC_PRU_ALERT_ATT_ID |

Response: WPTS_VALUE _CFM

Description: This message is used to Request the read of the PRU Information from the application. Information on four difference characteristic values cen be requested.

*NOTE :- As the PRU Static information is stored below the API in the ATT-DB, although shown here are a attribute which can be read, in reality the peers request to read the PRU static information will not result in an WPTS_VALUE_REQ_IND.*

### 7.2.3 WPTS_VALUE _CFM

Source: TASK_APP

Destination: TASK_WPTS

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | attrib_id | The item whose value was requested<br>*0x01 - WPTC_PRU_STATIC_ATT_ID*<br>0x02 – WPTC_PTU_STATIC_ATT_ID<br>0x03 - WPTC_PRU_CONTROL_ATT_ID<br>0x04 - WPTC_PRU_DYNAMIC_ATT_ID<br>0x05 - WPTC_PRU_ALERT_ATT_ID |
| uint8_t | status | Status of the operation:<br>• PRF_ERR_OK (0x00)<br>Other error code value (except 0x00) |
| **Union** | **value** | |
| struct wpts_pru_control | pru_ctrl | The PRU Control Information |
| struct wpts_ptu_static | ptu_static | The PTU Static Characteristic Information |
| *struct wpts_pru_static* | *pru_static* | *The PRU Static Characteristic Information* |
| struct wpts_pru_dynamic | pru_dynamic | The PRU Dynamic Characteristic Information |
| struct wpts_pru_alert | pru_alert | Alert Information on the PRU |

Description: The message supplies the value for the Read Request of the PRU Characteristic Value. This Message is the applications response to a WPTS_VALUE_REQ_IND message.

### 7.2.4 WPTS_PRU_ALERT_NOTIFY_CMD

Source: TASK_APP

Destination: TASK_WPTS

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | pru_alert | This is a bit-field which is used to transfer Alert conditions from the PRU to the PTU. See section 4.30 for a complete description of this field and range of values allowed (the lower 2 bits of the pru_alert are not used and this not to be interpreted in this message) |

Response: WPTS_CMP_EVT

Description: This message is used to notify the client of an alert condition. It excludes the Mode Transition which should be sent using the WPTS_PRU_MODE_TRANSITION_INDICATE_CMD.

### 7.2.5 WPTS_PRU_MODE_TRANSITION_INDICATE_CMD

Source: TASK_APP

Destination: TASK_WPTS

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | pru_alert | This is a bit-field which is used to transfer Alert conditions from the PRU to the PTU. See section 4.30 for a complete description of this field and range of values allowed. (the lower 2 bits of the pru_alert are not used and this not to be interpreted in this message) |
| uint_8 | mode_transition | 0x00 --- No Mode Transition<br>0x01 --- 2 sec Mode Transition Time Limit<br>0x02 --- 3 sec Mode Transition Time Limit<br>0x03 --- 6 sec Mode Transition Time Limit<br>0x04-0xFF -- UNDEFINED |
| uint_8 | addr[6] | A Bluetooth device address which is to be used by the PTU/PTU when they attempt to reconnect. |

Description: This message is used to notify the client of a Mode Transition on the server.

### 7.2.6 WPTS_CMP_EVT

Source: TASK_WPTS

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---|---|---|
| uint8_t | Operation | 0x00 – Reserved<br>0x01 – WPTS_NTF_ALERT<br>0x02 – WPTS_IND_MODE_TRANSITION<br>0x03 - WPTS_ENABLE<br>0x04-0xFF - Reserved |
| uint8_t | Status | Status |

Description: The API message is used by the WPTS task to inform the APP that a previously requested command is complete. For example this normally is used to inform the Application that the Mode_Transition_Indictate or Alert_Notify procedures is complete in the server and returns the status field indicating the outcome of the procedure.

# 8 Message Sequence Charts (MSCs)

This section uses message sequence charts to describe the operation of the profile. Message sequence charts are presented for the following scenarios.

## 8.1 Device Discovery and Connection

The operation of initial device discovery between and RW PTU and PRU is shown below in Figure 3.

The PRU becomes limited discoverable by initiating Undirected Advertising with the Advertising data containing the WPT UUID (to identify the service), the WPT Primary Service Handle and the PRU RSSI. To enable fast connections the Advertising interval should be less than 20ms.

The PTU configures LE scanning for optimal discovery and power performance, and initiates LE scanning. The interval chosen for scanning is dependent on state of the PTU and dependent on the number of links already existing. Short scan intervals or continuous scanning provide the fastest discovery time.

For each advertising report received the PTU checks if the WPT UUID is contained within the Advertising Data and also ensures that the PTU RSSI field meets requirements. If the PTU discover a device which meets the requirements for a connection, the PTU initiates a LE Connection. For unbonded devices the GAP General Connection Establishment procedure is used, however for bonded device the choice of connection establishment type is dependent on PTU requirements as described in A4WP specification [1].

Following connection establishment, both the PTU and PRU enable the profile. In the PTU the Primary Service Handle of the WPT obtained from the Advertising Data is passed to allow the WPTC to determine the Handle values for the Characteristics and Descriptors.

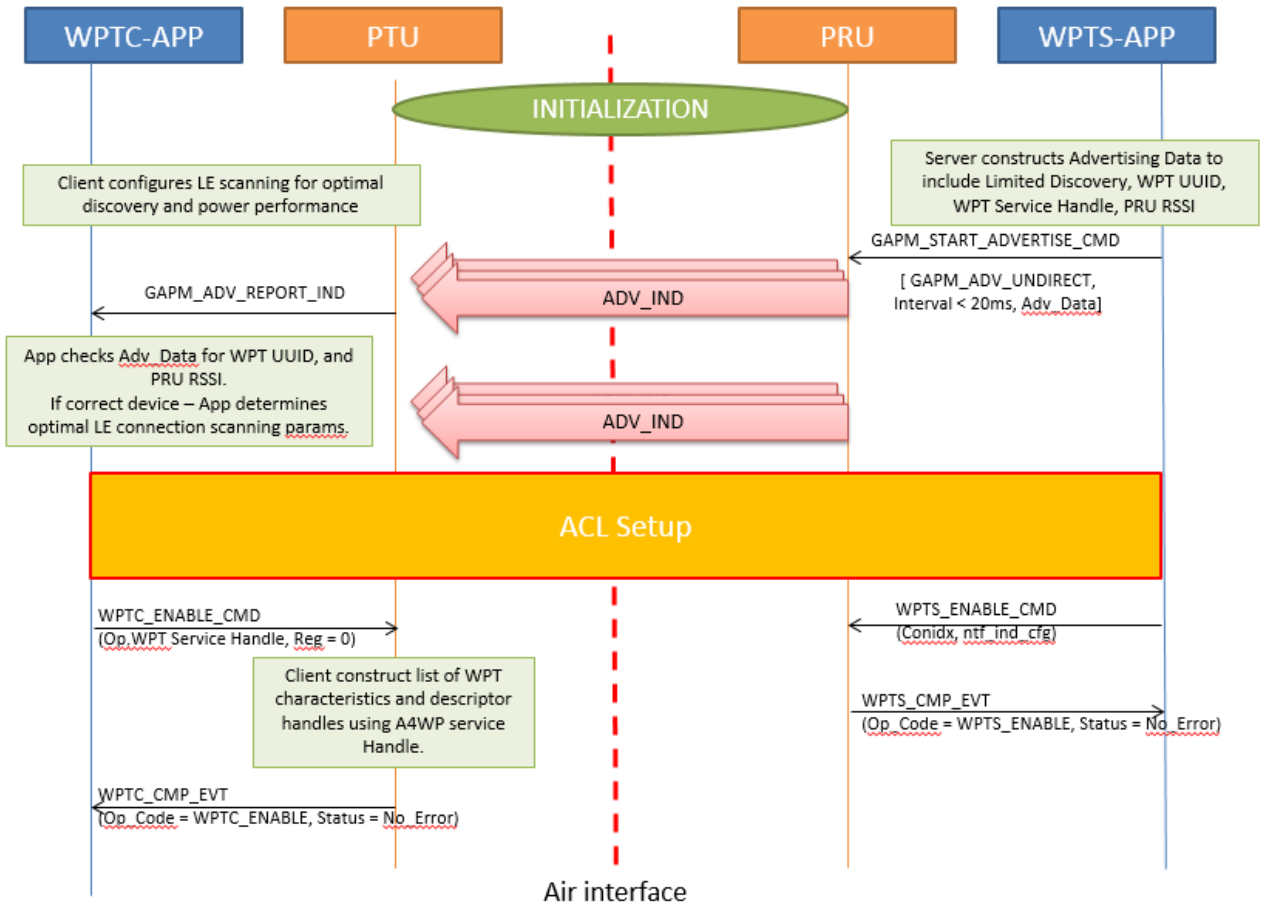


***Figure 3 :- Device Discovery and Connection***

The WPT places specific conditions for the acceptance of an Advertising. The PTU shall issue an attempt to establish a connection to the advertising device only if the following conditions are satisfied:

- The RSSI of the advertisement is greater than ADV_PWR_MIN (implementation dependent but -60dBm recommended) as measured at the receive antenna.
- The PTU observes an impedance shift close to the time of the advertisement.

If neither of these condition are satisfied the PTU should ignore advertisements from that device. However, if one of these conditions is satisfied then once the 11$^{th}$ Advertisement is received or more than 1.7 seconds has elapsed the PTU shall issue a connection request.

In all case the PTU shall issue a connection request within 50ms of receiving an acceptable Advertisement.

## 8.2 Device Registration

Figure 4 and 5 below shows the sequence of events for Device Registration. The PRU/PTU Static information is first exchanged using ATT-READ and ATT-WRITE-WITHOUT-RESPONSE. This is shown in Figure 4, where the PRU_Static Information is automatically retrieved as part of the WPTC_Enable_Cmd, which has Registration = 1. The Application then proceeds to write the PTU Static information into the PRU using WPTC_Write_Cmd.

As can be seen in Figure 5, Following the exchange of the Static information, The PRU Dynamic Information is then send form the PRU to the PTU using ATT-READ (the PRU Dynamic information will be retrieved from the PRU at least every 250ms). Finally the PTU writes modifies the state of the PRU by writing to the PRU control, using ATT-WRITE-WITHOUT-RESPONSE.

After writing to the PRU Control, the PTU can configure Alert notifications/indications as shown in Figure 6.

Following registration, the PTU should receive updates on the Dynamic Characteristic at least every 250ms. This is done by the PTU reading the Dynamic information at least every 250ms.

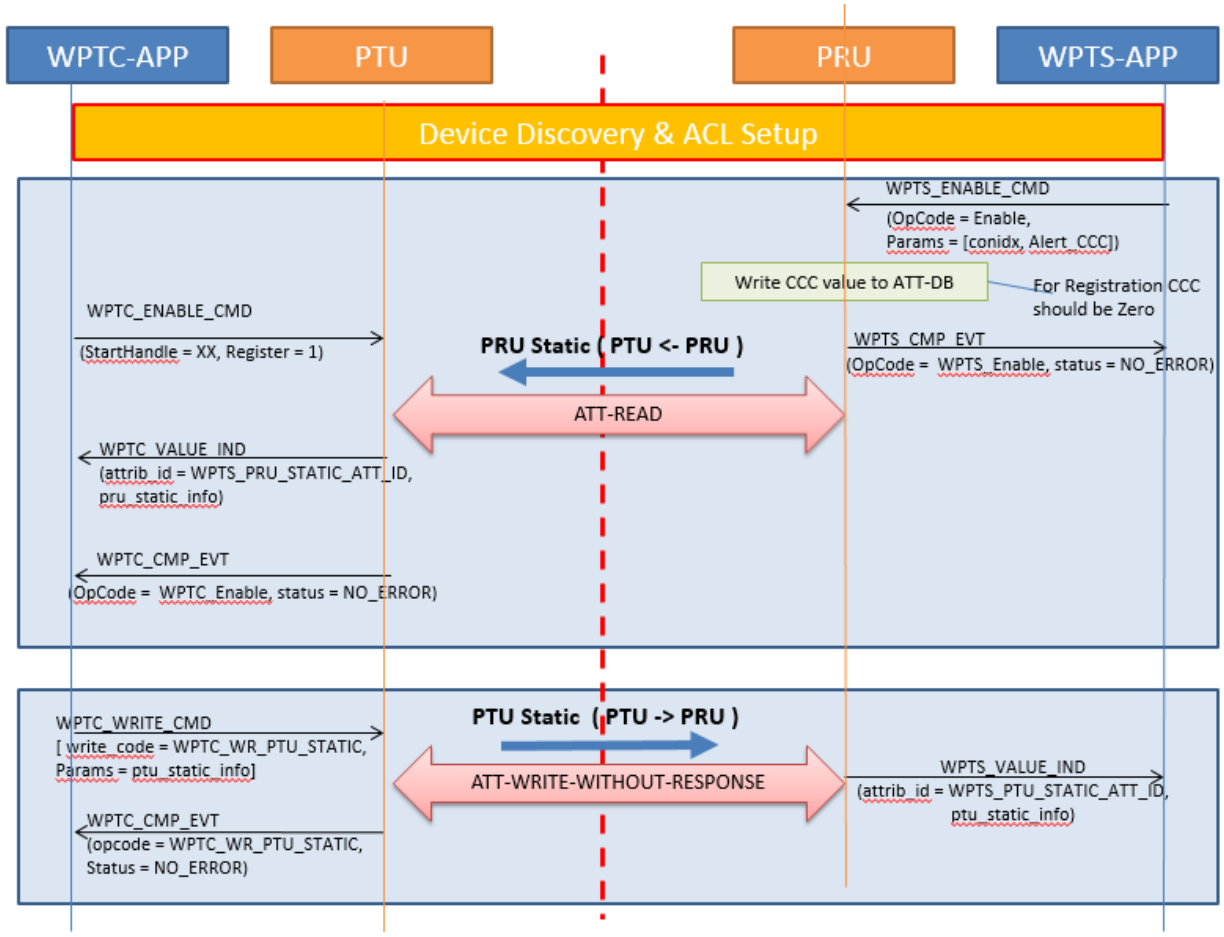


***Figure 4 :- Device Registration – Part 1***

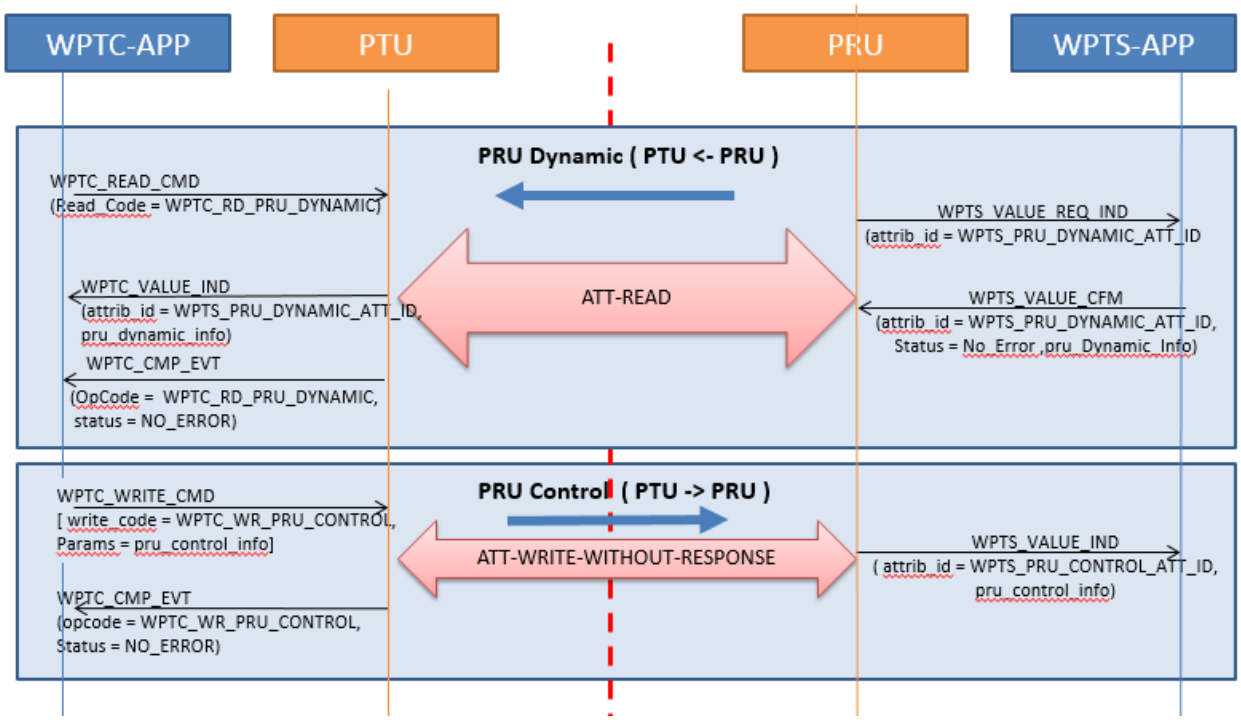


***Figure 5 :- Device Registration – Part 2***

## 8.3 Configuration of the Alert CCC

Following Registration the PTU configures the Alert CCC in the PRU to enable Notification and Indications. This allows the Alert_Notifications and Mode_Transition_Indications to be send from the PRU to the PTU.

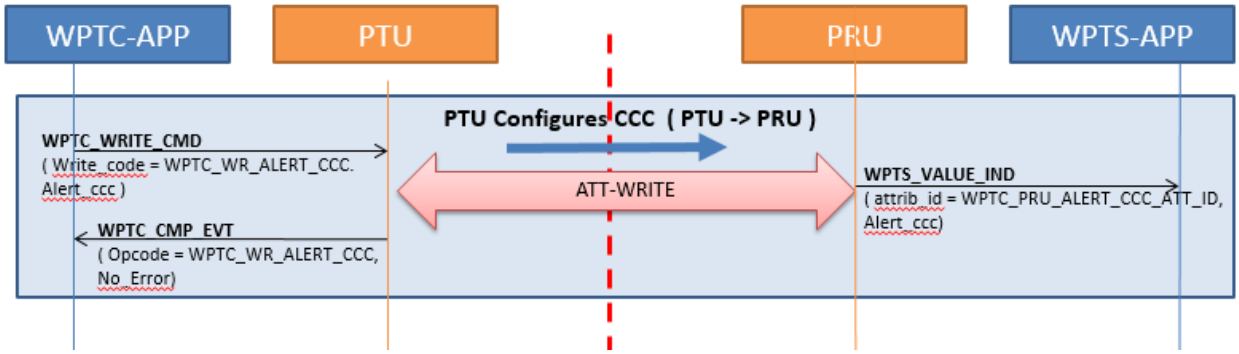


**Figure 6 :- PTU Configuring the CCC for the Alert Characteristic in the PRU**

## 8.4 Alert and Mode Transition

Figure 7 below shows the flow of messages at the API and GATT for both Alerting and Mode Transition. Alerting is used to inform the PTU of error conditions in the PRU, for example excess Voltage or Current, but does not include any Mode Transition information.

The Mode Transition informs the PTU of a specific error and intended actions to be taken by PRU to handle this error condition. This often involves the release of the BLE link and a reboot of the PRU. The Mode Transition message carries information on the length of time before the PRU is available for re-connection and the Bluetooth Device Address it will be using during advertising. Due to the sensitivity of the timing for re-connection the Mode Transition uses ATT-INDICATE which is acknowledged by the PTU. The WPTS_CMP_EVT is only sent when the ATT-INDICATE has been acknowledged, thus allowing the PRU to release link and perform re-boot on the receipt of WPTS_CMP_EVT.

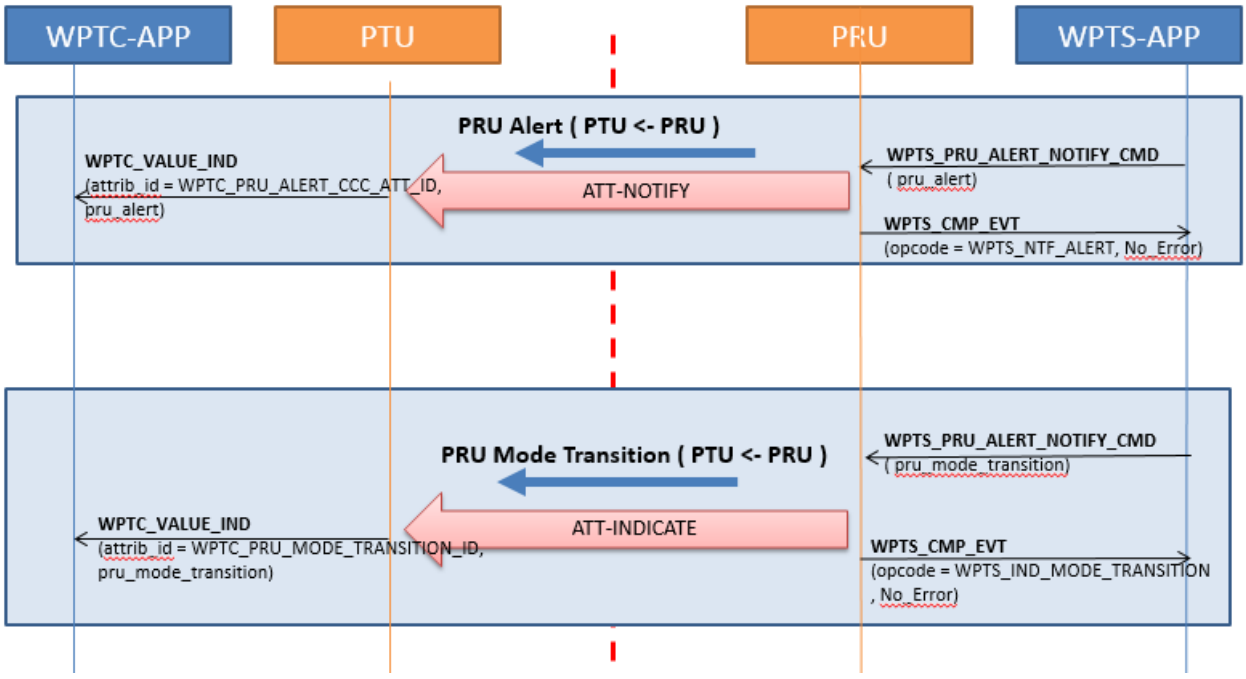


**Figure 7 :- Alerts and Mode Transitions**

# Abbreviations

| Abbreviation | Original Terminology |
|---|---|
| API | Application Programming Interface |
| BLE | Bluetooth Low Energy |
| GAP | Generic Access Profile |
| GATT | Generic Attribute Profile |
| WPTP | Wireless Power Transfer Profile |
| WPTS | Wireless Power Transfer Server Role |
| WPTC | Wireless Power Transfer Client Role |
| MSC | Message Sequence Chart |
| RW | RivieraWaves SAS |
| SM | Security Manager |
| PRU | Power Receive Unit |
| PTU | Power Transfer Unit |
| RFU | Reserved for Future Use |

# 9 References


| | | | | |
|---|---|---|---|---|
| **[1]** | **Title** | A4WP Wireless Power Transfer System Baseline System Specification(BSS) | | |
| | **Reference** | A4WP-S-0001 v1.3 | | |
| | **Version** | 1.3 | **Date** | 2014-05-11 |
| | **Source** | Alliance for Wireless Power | | |

| | | | | |
|---|---|---|---|---|
| **[2]** | **Title** | Wireless Power Transfer System Test Specification | | |
| | **Reference** | WPTP_TS_0.0.1 | | |
| | **Version** | 0.1 | **Date** | 2016-11-11 |
| | **Source** | RiveraWave | | |

| | | | | |
|---|---|---|---|---|
| **[3]** | **Title** | | | |
| | **Reference** | | | |
| | **Version** | | **Date** | |
| | **Source** | | | |

| | | | | |
|---|---|---|---|---|
| **[4]** | **Title** | | | |
| | **Reference** | | | |
| | **Version** | | **Date** | |
| | **Source** | | | |

# 10 Annex 1

This section provide some of the mappings used in fields of the API. These mappings are identical to the mappings provided in the A4WP for the corresponding fields.

The fields covered here are :

- PTU_power (decimal value of field to power value in Watts)
- PTU Maximum Source Impedance (decimal value of field to Impedance in Ohms)
- PTU Max Load Resistance (decimal value of field to Resistance in Ohms)

## 10.1 PTU Power

| Value | Pwr | Value | Pwr | Value | Pwr | Value | Pwr | Value | Pwr |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 32 | 4.4 | 64 | 13.6 | 96 | 28 | 128 | 50 |
| 1 | 0.1 | 33 | 4.6 | 65 | 14 | 97 | 28.5 | 129 | 51 |
| 2 | 0.2 | 34 | 4.8 | 66 | 14.4 | 98 | 29 | 130 | 52 |
| 3 | 0.3 | 35 | 5 | 67 | 14.8 | 99 | 29.5 | 131 | 53 |
| 4 | 0.4 | 36 | 5.2 | 68 | 15.2 | 100 | 30 | 132 | 54 |
| 5 | 0.5 | 37 | 5.4 | 69 | 15.6 | 101 | 30.6 | 133 | 55 |
| 6 | 0.6 | 38 | 5.6 | 70 | 16 | 102 | 31.2 | 134 | 56 |
| 7 | 0.7 | 39 | 5.8 | 71 | 16.4 | 103 | 31.8 | 135 | 57 |
| 8 | 0.8 | 40 | 6 | 72 | 16.8 | 104 | 32.4 | 136 | 58 |
| 9 | 0.9 | 41 | 6.3 | 73 | 17.2 | 105 | 33 | 137 | 59 |
| 10 | 1 | 42 | 6.6 | 74 | 17.6 | 106 | 33.6 | 138 | 60 |
| 11 | 1.1 | 43 | 6.9 | 75 | 18 | 107 | 34.2 | 139 | 61 |
| 12 | 1.2 | 44 | 7.2 | 76 | 18.4 | 108 | 34.8 | 140-255 | RFU |
| 13 | 1.3 | 45 | 7.5 | 77 | 18.8 | 109 | 35.4 | | |
| 14 | 1.4 | 46 | 7.8 | 78 | 19.2 | 110 | 36 | | |
| 15 | 1.5 | 47 | 8.1 | 79 | 19.6 | 111 | 36.6 | | |
| 16 | 1.6 | 48 | 8.4 | 80 | 20 | 112 | 37.2 | | |
| 17 | 1.7 | 49 | 8.7 | 81 | 20.5 | 113 | 37.8 | | |
| 18 | 1.8 | 50 | 9 | 82 | 21 | 114 | 38.4 | | |
| 19 | 1.9 | 51 | 9.3 | 83 | 21.5 | 115 | 39 | | |
| 20 | 2 | 52 | 9.6 | 84 | 22 | 116 | 39.6 | | |
| 21 | 2.2 | 53 | 9.9 | 85 | 22.5 | 117 | 40.2 | | |
| 22 | 2.4 | 54 | 10.2 | 86 | 23 | 118 | 40.8 | | |
| 23 | 2.6 | 55 | 10.5 | 87 | 23.5 | 119 | 41.4 | | |
| 24 | 2.8 | 56 | 10.8 | 88 | 24 | 120 | 42 | | |
| 25 | 3 | 57 | 11.1 | 89 | 24.5 | 121 | 43 | | |
| 26 | 3.2 | 58 | 11.4 | 90 | 25 | 122 | 44 | | |
| 27 | 3.4 | 59 | 11.7 | 91 | 25.5 | 123 | 45 | | |
| 28 | 3.6 | 60 | 12 | 92 | 26 | 124 | 46 | | |
| 29 | 3.8 | 61 | 12.4 | 93 | 26.5 | 125 | 47 | | |
| 30 | 4 | 62 | 12.8 | 94 | 27 | 126 | 48 | | |
| 31 | 4.2 | 63 | 13.2 | 95 | 27.5 | 127 | 49 | | |

## 10.2 PTU Maximum Source Impedance

| Value (Decimal) | PTU Maximum Source Impedance (ohms) |
|---|---|
| 0 | 50 |
| 1 | 60 |
| 2 | 70 |
| 3 | 80 |
| 4 | 90 |
| 5 | 100 |
| 6 | 110 |
| 7 | 120 |
| 8 | 130 |
| 9 | 140 |
| 10 | 150 |
| 11 | 175 |
| 12 | 200 |
| 13 | 225 |
| 14 | 250 |
| 15 | 275 |
| 16 | 300 |
| 17 | 350 |
| 18 | 375 |
| 19 - 31 | RFU |

## 10.3 PTU Maximum Load Resistance

| Value (Decimal) | PTU Max Load Resistance (ohms) |
|---|---|
| 1 | 10 |
| 2 | 15 |
| 3 | 20 |
| 4 | 25 |
| 5 | 30 |
| 6 | 35 |
| 7 | 40 |
| 8 | 45 |
| 9 | 50 |
| 10 | 55 |
| 11 - 31 | RFU |