

RW BLE Scan Parameters Profile Interface Specification

Interface Specification

RW-BLE-SCPP-IS

Version 8.00

2015-07-29

Revision History

| Version | Date | Revision Description | Author |
|---------|--------------------------------|---------------------------|--------|
| 1.0 | October 30 th 2012 | Initial release | LT |
| 2.0 | December 3 rd 2012 | Client Multi-Instance API | LT |
| 7.00 | November 27 th 2014 | Updated for BLE 4.1 | FBE |
| 8.00 | July 29 th 2014 | Updated for BLE 4.2 | CM |



Table of Contents

| | | |
|---------|--|----|
| 1 | Overview..... | 4 |
| 1.1 | Document Overview | 4 |
| 1.2 | BLE Scan Parameters Profile Overview | 4 |
| 2 | SCPP Server Role API..... | 5 |
| 2.1 | Environment..... | 5 |
| 2.2 | Initialization / Database Creation..... | 5 |
| 2.3 | API Messages..... | 6 |
| 2.3.1 | Connection Messages | 6 |
| 2.3.1.1 | SCPPS_ENABLE_REQ..... | 6 |
| 2.3.1.2 | SCPPS_ENABLE_RSP..... | 6 |
| 2.3.2 | Communication Messages | 7 |
| 2.3.2.1 | SCPPS_SCAN_INTV_WD_IND | 7 |
| 2.3.2.2 | SCPPS_SCAN_REFRESH_NTF_CFG_IND..... | 7 |
| 2.3.2.3 | SCPPS_SCAN_REFRESH_SEND_REQ..... | 7 |
| 2.3.2.4 | SCPPS_SCAN_REFRESH_SEND_RSP..... | 7 |
| 3 | SCPP Client Role API..... | 9 |
| 3.1 | Environment..... | 9 |
| 3.2 | Initialization | 9 |
| 3.3 | API Messages..... | 9 |
| 3.3.1 | Connection Messages | 9 |
| 3.3.1.1 | SCPPC_ENABLE_REQ | 9 |
| 3.3.1.2 | SCPPC_ENABLE_RSP | 10 |
| 3.3.2 | Communication Messages | 11 |
| 3.3.2.1 | SCPPC_SCAN_INTV_WD_WR_REQ..... | 11 |
| 3.3.2.2 | SCPPC_SCAN_INTV_WD_WR_RSP..... | 11 |
| 3.3.2.3 | SCPPC_SCAN_REFRESH_NTF_CFG_RD_REQ..... | 11 |
| 3.3.2.4 | SCPPC_SCAN_REFRESH_NTF_CFG_RD_RSP..... | 11 |
| 3.3.2.5 | SCPPC_SCAN_REFRESH_NTF_CFG_REQ | 12 |
| 3.3.2.6 | SCPPC_SCAN_REFRESH_NTF_CFG_RSP | 12 |
| 4 | Miscellaneous | 13 |
| 5 | Abbreviations..... | 14 |
| 6 | References..... | 15 |

1 Overview

1.1 Document Overview

This document describes the non-standard interface of the RivieraWaves (RW) Bluetooth Low Energy (BLE) Scan Parameters Profile (SCPP) implementation. Along this document, the interface messages will be referred to as API messages for the profile block(s).

Their description will include their utility and reason for implementation for a better understanding of the user and the developer that may one day need to interface them from a higher application.

1.2 BLE Scan Parameters Profile Overview

The Scan Parameters Profile is used to provide devices with information to assist them in managing their connection idle timeout and advertising parameters to optimize for power consumption and/or reconnection latency.

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

The various documents edited by the Bluetooth SIG present different use cases for this profile, their GATT, GAP and security, mandatory and optional requirements. The Scan Parameters Profile specifications have been adopted by the Bluetooth SIG on December 27th 2011 ([1] and [3]). Their related Test Specifications have been released at the same time and are referenced in [2] and [4].

The profile is implemented in the RW-BLE software stack as two tasks, one for each role. Each task has an API decided 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 Scan Parameters Service structure as defined in the Scan Parameter specification is exposed in the table below:

| Characteristic Name | Requirements | Properties | Security | Descriptors |
|----------------------|--------------|------------------------------------|----------|---|
| Scan Interval Window | Mandatory | Write Without Response (Mandatory) | None | None |
| Scan Refresh | Optional | Notify (Mandatory) | None | Client Characteristic Configuration <i>Requirement : If characteristic is supported.</i> <i>Permissions: Read (Mandatory) / Write (Mandatory)</i> |

The Scan Interval Window Characteristic is used to store the scan parameters of the Client. These parameters are the last known values of LE_Scan Interval (Maximum Scan Interval the Scan Client intends to use while scanning) and LE_Scan Window (Minimum Scan Windows the Scan Client intends to use while scanning in conjunction with the maximum Scan Interval written) of the Client.

The Scan Refresh characteristic is used to notify the Client that the Server requires the Scan Interval Window characteristic to be written with the latest values upon notification.



2 SCPP Server Role API

2.1 Environment

SCPPS task, is mono instantiated task, so the connection index is not present into the task number. This task is able to communicate with several peer devices in same time.

2.2 Initialization / Database Creation

During the initialization phase of the device, to use the Scan Parameters Service task, the SCPPS task has to be allocated and corresponding attribute database initialized, using GAPM API. Application has to send GAPM_PROFILE_TASK_ADD_CMD [6] with specific device required security level and following parameters.

Parameters:

| Type | Parameters | Description |
|---------|------------|---------------------|
| uint8_t | features | SCPS Configuration. |

The features parameter shall be used to specified if the Scan Refresh Characteristic is supported, possible values are SCPPS_SCAN_REFRESH_CHAR_NOT_SUP (0x00 – Scan Refresh Characteristic not added in the database) and SCPPS_SCAN_REFRESH_CHAR_SUP (0x01 – Scan Refresh Characteristic added in the database).



2.3 API Messages

2.3.1 Connection Messages

2.3.1.1 SCPPS_ENABLE_REQ

Source: TASK_APP

Destination: TASK_SCPPS

Parameters:

| Type | Parameters | Description |
|---------|------------|----------------------------|
| uint8_t | conidx | Connection Index. |
| uint8_t | ntf_cfg | Notification Configuration |

Response: SCPPS_ENABLE_RSP

Description: This API message can be used after the connection to restore bond data (peer notification configuration) of a known device.

2.3.1.2 SCPPS_ENABLE_RSP

Source: TASK_SCPPS

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---------|------------|-----------------------|
| uint8_t | conidx | Connection Index. |
| uint8_t | status | Status code (see [5]) |

Description: Inform application if restoring bond data for peer device succeed or not.

2.3.2 Communication Messages

2.3.2.1 SCPPS_SCAN_INTV_WD_IND

Source: TASK_SCPPS

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---------------------|--------------|--|
| uint8_t | conidx | Connection Index. |
| struct scan_intv_wd | scan_intv_wd | Scan Interval Window Characteristic value (see Table 4. 1) |

Description: This API message informs the application that the Scan Interval Window Characteristic value has been written by the peer device.

2.3.2.2 SCPPS_SCAN_REFRESH_NTF_CFG_IND

Source: TASK_SCPPS

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---------|------------|---|
| uint8_t | conidx | Connection Index. |
| uint8_t | ntf_cfg | Scan Refresh Notification Configuration |

Description: This API message is sent to the application to inform it that the peer device has enabled or disabled sending of notifications for the Scan Refresh Characteristic. This information can be stored as bond data in a non-volatile memory.

2.3.2.3 SCPPS_SCAN_REFRESH_SEND_REQ

Source: TASK_APP

Destination: TASK_SCPPS

Required state: CONNECTED

Parameters:

| Type | Parameters | Description |
|---------|------------|-------------------|
| uint8_t | conidx | Connection Index. |

Response: SCPPS_SCAN_REFRESH_SEND_RSP

Description: This API message is used to notify the Client that the Server writes the latest intended scan parameters to the Scan Interval Window Characteristic.

2.3.2.4 SCPPS_SCAN_REFRESH_SEND_RSP

Source: TASK_SCPPS

Destination: TASK_APP



Parameters:

| Type | Parameters | Description |
|---------|------------|-----------------------|
| uint8_t | conidx | Connection Index. |
| uint8_t | status | Status code (see [5]) |

Description: This API message is used to inform the application if the Scan Refresh Characteristic value has been notified or not.



3 SCPP Client Role API

3.1 Environment

Within the SCPPC task, three states are defined: FREE, IDLE, BUSY.

Important Note: The TASK_SCPPC task is multi-instantiated, one instance is created for each connection for which the profile will be enabled and each of these instances will have a different task ID. To communicate with the peer device, the corresponding connection index has to be used to calculate the SCPPC task instance.

The term TASK_SCPPC_IDX will be used in the rest of the document to refer to any instance of the Scan Parameters profile Client Role Task. The term TASK_SCPPC will refer to the first instance of this task.

3.2 Initialization

During the initialization phase of the device, to use the Scan Parameters Client task, the SCPPC task has to be allocated using GAPM API. Application has to send GAPM_PROFILE_TASK_ADD_CMD [6].

3.3 API Messages

3.3.1 Connection Messages

3.3.1.1 SCPPC_ENABLE_REQ

Source: TASK_APP

Destination: TASK_SCPPC_IDX

Required state: IDLE

Parameters:

| Type | Parameters | Description |
|---------------------|------------|--|
| uint8_t | con_type | Connection type |
| struct scps_content | scps | For bonded devices, information about the SCPS that has been found during the last discovery process (see Table 4. 2). |

Response: SCPPC_ENABLE_RSP

Description: This API message is used for enabling the Client role of the SCPP. This Application message contains BLE connection handle, the connection type and the previously saved discovered SCPS details on peer.

The connection type may be PRF_CON_DISCOVERY (0x00) for discovery/initial configuration or PRF_CON_NORMAL (0x01) for a normal connection with a bonded device. Application shall save those information to reuse them for other connections. During normal connection, previously discovered device information can be reused.

If it is a discovery /configuration type of connection, it is useless to fill the scps parameter are useless. Otherwise they will contain pertinent data which will be kept in the Client environment while enabled.

For a normal connection, the response to this request is sent right away after saving the SCPS content in the environment and registering SCPPC in GATT to receive the notifications for the known attribute handles in SCPS that would be notified (Scan Refresh Characteristic).

For a discovery connection, discovery of the peer SCPS is started and the response will be sent at the end of the discovery with the discovered attribute details.



Following the provided connection type and the SCPS information, several behaviors may happen. They are presented in the table below:

| Connection Type | Scan Refresh Characteristic Found | Comments |
|---|-----------------------------------|--|
| <i>Discovery (Connection establishment to non-bonded Scan Server)</i> | YES | Write the latest scan parameters (scan_intv_wd parameter) to the Scan Interval Windows Characteristic. |
| | NO | |
| <i>Normal (Connection establishment to bonded Scan Server)</i> | YES | Enable notifications of the Scan Refresh characteristic. |
| | NO | Write the latest scan parameters (scan_intv_wd parameter) to the Scan Interval Windows Characteristic. |

The Task for this profile role will go from IDLE state to CONNECTED state for a normal connection, and to DISCOVERING state for a discovery/configuration type of connection.

3.3.1.2 SCPPC_ENABLE_RSP

Source: **TASK_SCPPC_IDX**

Destination: TASK_APP

Required state: IDLE

Parameters:

| Type | Parameters | Description |
|---------------------|------------|---|
| uint8_t | status | Discovery status code (see [5]) |
| struct scps_content | scps | Information about the SCPS that has been found during the discovery process (see Table 4. 2). |

Description: This API message is used by the Client role task to either send the discovery results of SCPS on the peer device and confirm enabling of the Client role, or to simply confirm enabling of Client role if it is a normal connection and the attribute details are already known.

3.3.2 Communication Messages

3.3.2.1 SCPPC_SCAN_INTV_WD_WR_REQ

Source: TASK_APP

Destination: TASK_SCPPC_IDX

Required state: CONNECTED

Parameters:

| Type | Parameters | Description |
|---------------------|--------------|----------------------------------|
| struct scan_intv_wd | scan_intv_wd | Scan Parameters (see Table 4. 2) |

Response: SCPPC_SCAN_INTV_WD_WR_RSP

Description: This API message shall be used to inform the Scan Server that the Scan Client has changed its intended scanning behavior. It will write the Scan Interval Window Characteristic value in the Scan Server database.

The provided scan parameters are saved within the role task environnement so that when the Scan Server requires the latest scan parameters (sends a notification for the Scan Refresh Characteristic), these values are automatically written in its database.

3.3.2.2 SCPPC_SCAN_INTV_WD_WR_RSP

Source: TASK_SCPPC_IDX

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---------|------------|--------------------------------|
| uint8_t | status | Write request status (see [5]) |

Description: This API message is sent to the application when a write response has been received from the peer device after sending of a write request.

3.3.2.3 SCPPC_SCAN_REFRESH_NTF_CFG_RD_REQ

Source: TASK_APP

Destination: TASK_SCPPC_IDX

Parameters: None

Response: SCPPC_SCAN_REFRESH_NTF_CFG_RD_RSP

Description: This API message shall be used to read the value of the Scan Refresh Characteristic Client Characteristic Configuration Descriptor.

3.3.2.4 SCPPC_SCAN_REFRESH_NTF_CFG_RD_RSP

Source: TASK_SCPPC_IDX

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|----------|------------|--|
| uint8_t | status | Status code (see [5]) |
| uint16_t | ntf_cfg | Configuration value : PRF_CLI_STOP_NTFIND or PRF_CLI_START_NTF. |

Description: This API message is sent to the application to inform it about the read Client Characteristic Configuration Descriptor value for the Scan Refresh Characteristic.

3.3.2.5 SCPPC_SCAN_REFRESH_NTF_CFG_REQ

Source: TASK_APP

Destination: TASK_SCPPC_IDX

Required state: CONNECTED

Parameters:

| Type | Parameters | Description |
|----------|------------|--|
| uint16_t | ntf_cfg | Configuration value to write: PRF_CLI_STOP_NTFIND or PRF_CLI_START_NTF. |

Response: SCPPC_SCAN_REFRESH_NTF_CFG_RSP

Description: This API message shall be used to either enable or disable notifications for the Scan Refresh Characteristic.

3.3.2.6 SCPPC_SCAN_REFRESH_NTF_CFG_RSP

Source: TASK_SCPPC_IDX

Destination: TASK_APP

Parameters:

| Type | Parameters | Description |
|---------|------------|--------------------------------|
| uint8_t | status | Write request status (see [5]) |

Description: This API message is sent to the application when a write response has been received from the peer device after sending of a write request.

4 Miscellaneous

| Type | Parameters | Description |
|----------|----------------|---------------|
| uint16_t | le_scan_intv | Scan Interval |
| uint16_t | le_scan_window | Scan Window |

Table 4. 1 – Scan Interval Window Characteristic value structure (struct scan_intv_wd)

| Type | Parameters | Description |
|--------------------------|-----------------------|---|
| struct prf_svc | svc | Start handle and End handle of the SCPS (see Table 4. 3). |
| struct prf_char_inf | chars[SCPPC_CHAR_MAX] | Informations about the found Characteristics (see Table 4. 4). <i>SCPPC_CHAR_MAX = 2</i> <i>chars[SCPPC_CHAR_SCAN_INTV_WD] = Scan Interval Window Characteristic</i> <i>chars[SCPPC_CHAR_SCAN_REFRESH] = Scan Refresh Characteristic</i> |
| struct prf_char_desc_inf | descs[SCPPC_DESC_MAX] | Informations about the found Descriptors (see Table 4. 5). <i>SCPPC_DESC_MAX = 1</i> <i>descs[SCPPC_DESC_SCAN_REFRESH_CFG] = Client Characteristic Configuration Descriptor</i> |

Table 4. 2 – Scan Parameters Service Content structure (struct scps_content)

| Type | Parameters | Description |
|----------|------------|------------------------------|
| uint16_t | shdl | Start handle of the Service. |
| uint16_t | ehdl | End handle of the Service. |

Table 4. 3 – Service description structure (struct prf_svc)

| Type | Parameters | Description |
|----------|---------------|--|
| uint16_t | char_hdl | Characteristic declaration attribute handle. |
| uint16_t | val_hdl | Characteristic value attribute handle. |
| uint8_t | prop | Properties |
| uint8_t | char_ehdl_off | Number of attribute within the Characteristic. |

Table 4. 4 – Characteristic description structure (struct prf_char_inf)

| Type | Parameters | Description |
|----------|------------|-----------------------------|
| uint16_t | desc_hdl | Descriptor attribute handle |

Table 4. 5 – Descriptor description structure (struct prf_char_desc_inf)



5 Abbreviations

| Abbreviation | Original Terminology |
|--------------|-------------------------------------|
| API | Application Programming Interface |
| BLE | Bluetooth Low Energy |
| GAP | Generic Access Profile |
| GATT | Generic Attribute Profile |
| SCPP | Scan Parameters Profile |
| SCPPS | Scan Parameters Profile Server Role |
| SCPPC | Scan Parameters Profile Client Role |
| SCPS | Scan Parameters Service |
| RW | RivieraWaves |

6 References

| | | | | |
|------------|------------------|---------------------------------------|-------------|------------|
| [1] | Title | SCAN PARAMETERS PROFILE SPECIFICATION | | |
| | Reference | ScPP_SPEC_V10 | | |
| | Version | V10r00 | Date | 2011-12-27 |
| | Source | Bluetooth SIG | | |

| | | | | |
|------------|------------------|--|-------------|------------|
| [2] | Title | SCAN PARAMETERS PROFILE TEST SPECIFICATION | | |
| | Reference | ScPP.TS.1.0.0 | | |
| | Version | 1.0.0 | Date | 2011-12-27 |
| | Source | Bluetooth SIG | | |

| | | | | |
|------------|------------------|---------------------------------------|-------------|------------|
| [3] | Title | SCAN PARAMETERS SERVICE SPECIFICATION | | |
| | Reference | ScPS_SPEC_V10 | | |
| | Version | V10r00 | Date | 2011-12-27 |
| | Source | Bluetooth SIG | | |

| | | | | |
|------------|------------------|--|-------------|------------|
| [4] | Title | SCAN PARAMETERS SERVICE TEST SPECIFICATION | | |
| | Reference | ScPS.TS.1.0.0 | | |
| | Version | 1.0.0 | Date | 2011-12-27 |
| | Source | Bluetooth SIG | | |

| | | | | |
|------------|------------------|--|-------------|------------|
| [5] | Title | RW BLE Host Error Code Interface Specification | | |
| | Reference | RW-BLE-HOST-ERR-CODE-IS | | |
| | Version | 7.00 | Date | 2014-06-30 |
| | Source | RivieraWaves SAS | | |

| | | | | |
|------------|------------------|-----------------------------|-------------|------------|
| [6] | Title | GAP Interface Specification | | |
| | Reference | RW-BLE-GAP-IS | | |
| | Version | 7.00 | Date | 2014-06-30 |
| | Source | RivieraWaves SAS | | |