# L2C Interface Specification

Interface Specification

RW-BLE-L2C-IS

Version 8.00

2016-04-15

# Revision History

| Version | Date | Revision Description | Author |
|---------|------|---------------------|--------|
| 8.00 | 2016-04-15 | First version BLE 4.2 | FBELOUIN |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Table of Contents

## List of Tables and Figures

# 1    Overview

The RW-BLE Logical Link Control and Adaptation Protocol (L2CAP also named L2C in the document) defines the procedures related the ACL communication between two devices connected over a BLE link.

This document describes common format requirements for parameters accessible on the user interface level.

## 1.1 Document Overview

This document describes the non-standard interface of the RW-BLE L2CAP implementation. Along this document, the interface messages will be referred to as API messages for the profile block(s).

Their descriptions 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.

Moreover, it is recommended that the user check the html-based documentation of the RW-BLE Host, which is derived from actual RW-BLE host code and formatted via Doxygen. This material can further provide information on RW-BLE L2C implementation (e.g. data structures, states, message calling).

# 2    Default Type and Enumeration Definition

❖ **l2cc_sdu**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint16_t | cid | L2CAP Channel Identifier |
| uint16_t | credit | Number of credit used (information fill by SW) |
| uint16_t | length | SDU Data length |
| uint8_t[] | data | SDU Data |

**Table 1: L2Cap SDU Structure**

# 3    L2CAP Manager (L2CM)

The L2CAP manager does not provide any API. It's in charge of managing number of transmission buffers available at controller level.

# 4      L2CAP Controller (L2CC)

L2CAP Controller (L2CC) is a multi-instantiated task used to manage ACL communication between devices.

Not described into this document because not available at Application and/or Profile level, it's in charge of the protocol communication such as Attribute protocol, Secure Management protocol or Signaling protocol.

This module provides to Application and/or Profiles an API to create or manage LE Credit Based Connection (LECB).

Also it provides a communication Pipe between two connected devices using an established LE Credit Based Connection.

The L2CAP Controller block has handlers for these messages, defined in l2cc_task.h.

## 4.1 Operations Flags

The block uses request flag options embedded in the interface message sent to L2CAP Controller. This flag ensures correct handling of the operation request from the application.

| Value | Flag | Description |
| --- | --- | --- |
| 0x00 | L2CCC_NO_OP | No operation |
| | **LE Credit Based Connection management** | |
| 0x01 | L2CC_LECB_CONNECT | Initiate establishment of a LE credit based connection |
| 0x02 | L2CC_LECB_DISCONNECT | Disconnect a LE Credit Based connection |
| 0x03 | L2CC_LECB_CREDIT_ADD | Increment number of credit for a local channel |
| 0x04 | L2CC_LECB_SDU_SEND | Send SDU over LE credit based connection |

**Table 2: L2CC Operation Flags**

## 4.2    Generic Interface

The generic L2CAP Controller offers a set of commands that are completed with following command completed event message.

### 4.2.1    L2CC_CMP_EVT

**Parameters:**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint8_t | operation | L2CC operation code (see Table 2) |
| uint8_t | status | Status of the operation (see [4]) |
| uint16_t | cid | L2CAP Channel identifier |
| uint16_t | credit | Number of peer credit used - only relevant for LECB |

**Description:**

Complete event for L2CAP operations. All operation triggers this event when operation is finished.

## 4.3    LE Credit Based Connection (aka LE Credit Oriented Channel)

The LE credit based Connection, also called Connection Oriented Channel **(COC)** is an L2CAP feature managed by L2CAP Controller Task. It allows a LE Service to create a dedicated channel on a specific link.

Peer service client should connect to this LE Credit Based Connection before exchanging any packets.

**Registration of LE Protocol/Service Multiplexer (LE_PSM)**

On Server side, in order to accept a LE Credit Based connection from peer device on a specific LE_PSM, application/profile has to first register it. Any connection from peer device on an unregister LE_PSM is automatically rejected by L2CAP controller task without informing application. Registration of LE_PSM is managed by Generic Access Profile Manager (GAPM) task (see [7]) this should be done at device initialization.

LE PSM is a <u>unique</u> parameter which corresponds to:

| Range | Type | Server Usage | Client Usage |
|---|---|---|---|
| 0x0001 - 0x007F | Fixed, SIG assigned | LE_PSM is fixed for all implementations | LE_PSM may be assumed for fixed service. Protocol used is indicated by the LE_PSM as defined in the Bluetooth SIG assigned numbers page. |
| 0x0080 - 0x00FF | Dynamic | LE_PSM may be fixed for a given implementation or may be assigned at the time the service is registered in GATT | LE_PSM shall be obtained from the service in GATT upon every reconnection. LE_PSM for one direction will typically be different from the other direction. |
| 0x0100 - 0xFFFF | Reserved | Not applicable | Not applicable |

**Connection to a specific LE Credit Based channel**

Figure 4-1 shows a normal LE Credit based Connection establishment.

First, before any connection, the server side has to register a LE Protocol/Service Multiplexer (LE_PSM).

Then over a BLE Link, client side can initiate a LE Credit Based connection using L2CC_LECB_CONNECT_CMD message.

On server side, if LECB connection request pass all internal sanity check, a L2CC_LECB_CONNECT_REQ_IND is issued to task that manage the requested LE_PSM. A L2CC_LECB_CONNECT_CFM message has then to be sent in response to accept or not the connection.

Finally both client and server receive a L2CC_LECB_CONNECT_IND message when LECB connection is established.
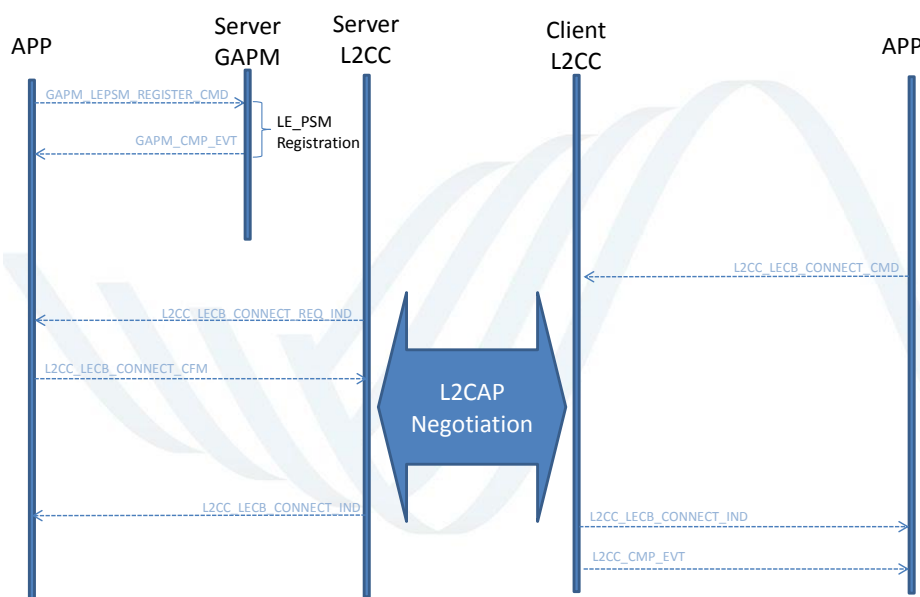


**Figure 4-1: LE Credit Based Connection establishment**

At LE Credit Based connection establishment, each device provides:

- Maximum Transmission Unit (MTU) size which is the Max SDU size that can be received

- Maximum Packet Size (MPS) which is the maximum LE-Frame size (L2CAP PDU size) that device can receive

- Local device channel identifier which shall be between specified range: [0x40, 0x7F]

- Number of local credit. On current implementation, this number of credit shall be at least **N** in order to receive at least a complete SDU.

  o **N = floor( ((MTU + 2) + (MPS -1)) / MPS ) + 1**

**Data transfer and flow control**

Figure 4-2 shows Service Data Unit (SDU) transfer and flow control.

- **SDU Transmission:**

To send a SDU, application has to send L2CC_LECB_SDU_SEND_CMD message with destination channel identifier and data. A L2CC_CMP_EVT is issued by L2CAP controller task when the SDU is ready to be sent over BLE link.

The L2Cap controller task is in charge of segmenting the SDU in several LE-Frames according to peer device Max Packet Size (MPS). For each LE-Frame, one peer credit is used. When number of peer credit reach zero, no more LE-Frame are sent until peer device increment its local number of credit. When peer device increment its number of credit, automatically the SDU send is resumed. When all segment have been pushed to controller, L2CAP controller task inform application that SDU transfer is over using L2CC_CMP_EVT and provide number of credit used.

**Note:** It's not allowed to request transmission of a new SDU until the L2CC_CMP_EVT is received for previous SDU. In the other hand, it's possible to request SDU transmission even if there is no more credit available for peer device.

- **SDU Reception:**

When a SDU is fully received at L2CAP controller level (L2CC is in charge of packet reassembly) the L2CC_LECB_SDU_RECV_IND message is issued to application with received data and number of credit used.

Application is in charge, when received SDU is processed, to re-increment number of credit using L2CC_LECB_ADD_CMD.

Application is also informed with L2CC_LECB_ADD_IND message when peer device adds credit on its channel. This message is here only for information but can be ignored by application.

**Note:** L2Cap Controller task has an automatic credit adding mechanism. When a SDU is received with multiple LE-Frames, if LE-Frame size is less than connection Max Packet Size (MPS), task checks if number of credit used is greater than number of credit expected. If too many credits are used, L2CAP provides more number of credits to peer device.

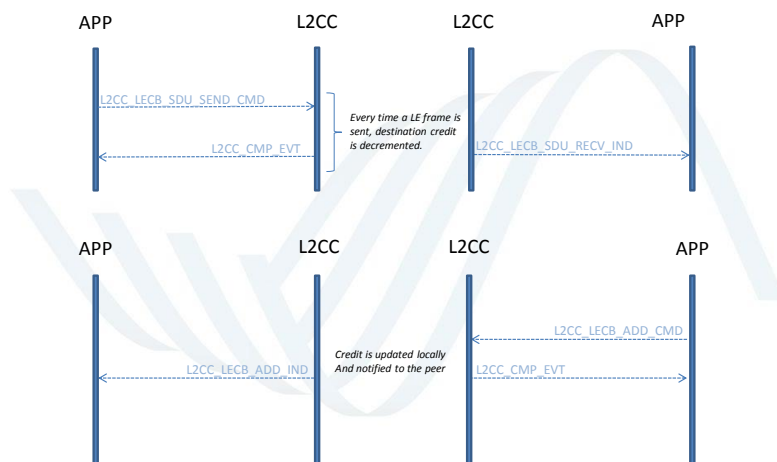**Figure 4-2: SDU transmission and reception over LECB Link and flow control**

## Disconnection

Figure 4-3 shows that the disconnection can be initiated by both device of a LE Credit Based Connection using L2CC_LECB_DISCONNECT_CMD message. When the communication is over, a L2CC_LECB_DISCONNECT_IND is issued by L2CAP controller task.
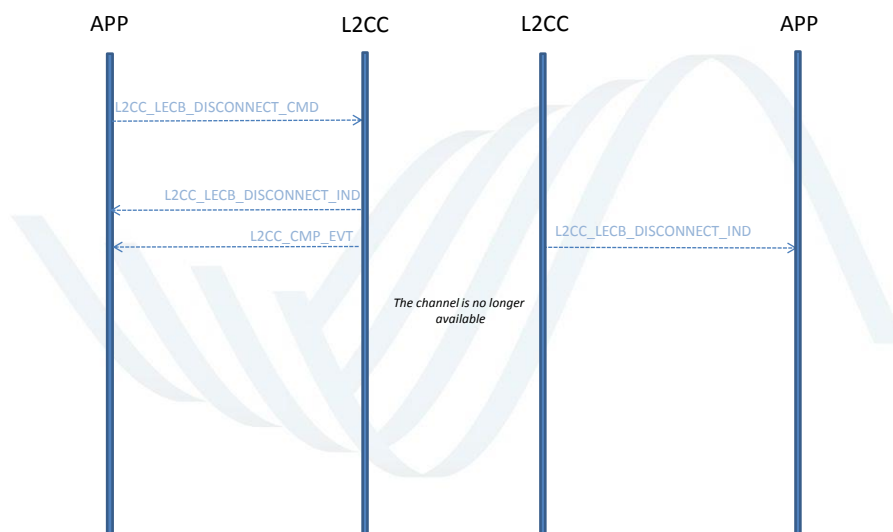


**Figure 4-3: Disconnection of a LE Credit Based Connection.**

### 4.3.1 L2CC_LECB_CONNECT_CMD

**Parameters:**

| Type | Parameters | Description |
|---|---|---|
| uint8_t | operation | GAPC requested operation (see Table 2):<br>- **L2CC_LECB_CONNECT:** LE credit based connection establishment |
| uint16_t | le_psm | LE Protocol/Service Multiplexer |
| uint16_t | local_cid | Local Channel identifier (range [0x40, 0x7F])<br>**0**: automatically allocate a free channel |
| uint16_t | local_credit | Credit allocated for the LE Credit Based Connection<br>Shall be at least: **floor(((MTU + 2) + (MPS - 1)) / MPS) + 1**<br>To be sure that 1 SDU can be fully received without requesting credits to application |
| uint16_t | local_mtu | Maximum SDU size - Shall not exceed device MTU that local device can receive |
| uint16_t | local_mps | Maximum Packet size - Shall not exceed device MPS that local device can receive |

**Response:**

L2CC_LECB_CONNECT_IND: Triggered when connection is created

L2CC_CMP_EVT: When operation is completed.

**Description:**

Initiate establishment of a LE Credit based channel for specific LE Protocol/Service Multiplexer.

If the Local channel identifier is set to zero, host stack will find an available channel in the authorized range. Value of allocated channel is provided in L2CC_LECB_CONNECT_IND message.

### 4.3.2 L2CC_LECB_CONNECT_REQ_IND

**Parameters:**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint16_t | le_psm | LE Protocol/Service Multiplexer |
| uint16_t | peer_cid | Peer channel identifier used for the LE Credit Based Connection |
| uint16_t | peer_mtu | Maximum SDU size that peer device can receive |
| uint16_t | peer_mps | Maximum Packet size that peer device can receive |

**Response:**

L2CC_LECB_CONNECT_CFM: to confirm that connection can be established or not.

**Description:**

Event triggered when peer device requests a LE Credit connection to be established.

### 4.3.3    L2CC_LECB_CONNECT_CFM

**Parameters:**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint16_t | le_psm | LE Protocol/Service Multiplexer |
| uint16_t | peer_cid | Peer channel identifier |
| uint8_t | accept | **1** to accept the incoming connection<br>**0** to reject it |
| uint16_t | local_cid | Local Channel identifier (range [0x40, 0x7F])<br>**0**: automatically allocate a free channel |
| uint16_t | local_credit | Credit allocated for the LE Credit Based Connection<br>Shall be at least: **floor(((MTU + 2) + (MPS - 1)) / MPS) + 1**<br>To be sure that 1 SDU can be fully received without requesting credits to application |
| uint16_t | local_mtu | Maximum SDU size - Shall not exceed device MTU that local device can receive |
| uint16_t | local_mps | Maximum Packet size - Shall not exceed device MPS that local device can receive |

**Response:**

> L2CC_LECB_CONNECT_IND: When connection creation is over or if an error occurs

**Description:**

> Response to L2CC_LECB_CONNECT_REQ_IND for accepting or rejecting an incoming LE Credit based connection for specific LE Protocol/Service Multiplexer. Status reflects the result of the operation and should be filled according to the following table.

### 4.3.4 L2CC_LECB_CONNECT_IND

**Parameters:**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint8_t | status | Status of the connection establishment (see [4]) GAP_ERR_NO_ERROR means connection established |
| uint16_t | le_psm | LE Protocol/Service Multiplexer |
| uint16_t | local_cid | Local Channel identifier |
| uint16_t | peer_cid | Peer Channel identifier |
| uint16_t | peer_credit | Peer number of credits for the LE Credit Based Connection |
| uint16_t | peer_mtu | Peer Maximum SDU size that can be received |
| uint16_t | peer_mps | Peer Maximum Packet size that can be received |

**Description:**

Event triggered when LE Credit connection is established.

### 4.3.5    L2CC_LECB_DISCONNECT_CMD

**Parameters:**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint8_t | operation | L2CC requested operation  (see Table 2):<br>-    **L2CC_LECB_DISCONNECT**: LE credit based disconnection request |
| uint16_t | peer_cid | Peer channel identifier |

**Response:**

L2CC_LECB_DISCONNECT_IND: Triggered when connection is terminated

L2CC_CMP_EVT: When operation completed.

**Description:**

Disconnect an LE Credit Based connection

### 4.3.6     L2CC_LECB_DISCONNECT_IND

**Parameters:**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint16_t | le_psm | LE Protocol/Service Multiplexer |
| uint16_t | local_cid | Local Channel identifier |
| uint16_t | peer_cid | Peer Channel identifier |
| uint8_t | reason | Reason for disconnection (see [4]) |

**Description:**

Event triggered when a LE Credit connection is terminated.

### 4.3.7    L2CC_LECB_ADD_CMD

**Parameters:**

| Type | Parameters | Description |
|------|------------|-------------|
| uint8_t | operation | L2CC requested operation  (see Table 2):<br>-    **L2CC_LECB_CREDIT_ADD**: Add credit to local channel |
| uint16_t | local_cid | Local Channel identifier |
| uint16_t | credit | Number of credits added locally for channel identifier |

**Response:**

   L2CC_CMP_EVT: When operation completed.

**Description:**

   Inform peer device that new credits are available for LE Credit Connection

### 4.3.8 L2CC_LECB_ADD_IND

**Parameters:**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint16_t | peer_cid | Peer Channel identifier |
| uint16_t | peer_added_credit | Relative number of credit added by peer on its channel identifier |

**Description:**

Event triggered when peer device increases number of credit. This event can be ignored by application, just used for information.

### 4.3.9    L2CC_LECB_SDU_SEND_CMD

**Parameters:**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint8_t | operation | L2CC requested operation  (see Table 2):<br>-     **L2CC_LECB_SDU_SEND:** Send a SDU |
| uint16_t | offset | Internal value, no need to initiate it. |
| struct l2cc_sdu | sdu | SDU information (see Table 1) |

**Response:**

L2CC_CMP_EVT: When device controller is ready to send full SDU packet.

**Description:**

Send a Service Data Unit (SDU) to peer device.

Application has to wait L2CC_CMP_EVT before sending a new SDU on same channel identifier.

Application can send simultaneously several SDU on different channel identifier without waiting for L2CC_CMP_EVT message.

There is no need to have credit on peer channel identifier to request sending SDU. L2CAP controller task will send it as soon as new credits are added by peer device.

Number of credit used for packet transmission is present in L2CC_CMP_EVT message.

### 4.3.10     L2CC_LECB_SDU_RECV_IND

**Parameters:**

| Type | Parameters | Description |
|------|-----------|-------------|
| uint8_t | status | Status of the packet reception(see [4])<br>GAP_ERR_NO_ERROR means packet correctly received |
| uint16_t | offset | Internal value, can be ignored |
| struct l2cc_sdu | sdu | SDU information (see Table 1) |

**Description:**

Event triggered when a complete Service Data Unit (SDU) has been received from peer device.

This message contains receive SDU data and number of local credit used for the reception.

When the SDU is processed, application should request to add used credit to ensure reception of new SDU from peer device using L2CC_LECB_ADD_CMD.

# References

| [1] | Title | Specification of the Bluetooth System | | |
|---|---|---|---|---|
| | Reference | Bluetooth Specification | | |
| | Version | 4.2 | Date | 2014-12-02 |
| | Source | Bluetooth SIG | | |

| [2] | Title | RW-BLE-SW-HOST-FS | | |
|---|---|---|---|---|
| | Reference | RW-BLE Host Functional Specification | | |
| | Version | 8.00 | Date | 2015-06-29 |
| | Source | RivieraWaves SAS | | |

| [3] | Title | RW-BLE-SW-IS | | |
|---|---|---|---|---|
| | Reference | Interface Specification of RW-BLE Link Layer | | |
| | Version | 7.0 | Date | 2014-10-13 |
| | Source | RivieraWaves SAS | | |

| [4] | Title | RW-BLE-HOST-ERR-CODE-IS | | |
|---|---|---|---|---|
| | Reference | RW BLE Host Error Code Interface Specification | | |
| | Version | 8.04 | Date | 2015-10-26 |
| | Source | RivieraWaves SAS | | |

| [5] | Title | org.bluetooth.characteristic.gap.appearance | | |
|---|---|---|---|---|
| | Reference | Bluetooth appearance field description | | |
| | Version | N/A | Date | N/A |
| | Source | http://developer.bluetooth.org/gatt/characteristics/Pages/CharacteristicViewer.aspx?u=org.bluetooth.characteristic.gap.appearance.xml | | |

| [6] | Title | AD Type | | |
|---|---|---|---|---|
| | Reference | EIR Data Type and Advertising Data Type (AD Type) Values | | |
| | Version | N/A | Date | N/A |
| | Source | https://www.bluetooth.org/en-us/specification/assigned-numbers-overview/generic-access-profile | | |

| [7] | Title | RW-BLE-GAP-IS | | |
|---|---|---|---|---|
| | Reference | Generic Access Profile Interface Specification | | |
| | Version | 8.05 | Date | 2016-04-15 |
| | Source | RivieraWaves SAS | | |