# BLE RDK IR Service Specification

RDK-SP-BLE-RIR-Service-D01-200515

Document Status: Draft

May 15, 2020

**Document Status**

| | |
|---|---|
| **Document Control Number:** | RDK-SP-BLE-RIR-Service-D01-200515 |
| **Document Title:** | BLE RDK IR Service Specification |
| **Versions:** | D01 – September 13, 2019 |
| **Date:** | May 15, 2020 |
| **Status:** | Document Status: Draft |
| **Distribution:** | RDK members only |

**Document Status Codes**

**Work in Progress (W)**      An incomplete document designed to guide discussion and generate feedback that may include several alternative requirements for consideration.

**Draft (D)**                 A document in specification format considered largely complete, but lacking review. Drafts are susceptible to substantial change during the review process.

**Issued (I)**                A stable document that has undergone rigorous review and is suitable for product design and development. It will serve as a basis for testing requirements.

**Table of Contents**

**Tables**

**Figures**

# 1. Introduction

## 1.1  Overview

The RDK IR Service exposes data and associated formatting for programming infrared signals into an RDK Remote Control Device.

## 1.2  Purpose of Document

This document defines detailed requirements for the RDK IR Service. It is intended to specify download and management of IR descriptors between an RIR server and an RIR client.

## 1.3  Typographical Conventions

This specification uses different typefaces to differentiate and emphasize important information.

**Table 1 - Typographical Conventions**

| Typeface | Usage |
|---|---|
| Boldface | Used to call attention to a piece of information. For example: <br> This specification does **not** include headend diagnostic screens. |
| Boldface & Uppercase | Used to emphasize information and for readability. For example: <br> **ENTER, MUTE, INFO, VOL +/-** and other buttons on the remote control. |
| Italics | Used to emphasize that the information being presented is for informational purposes only and is not a requirement even though it may contain conformance language. For example: <br> *Note: The voice controller uses the Channel Check Request to verify that the voice target has disabled frequency agility.* |
| Uppercase | Used to define and signify a requirement. For example: <br> MUST, SHOULD, and MAY. |

## 1.4  Requirements (Conformance Notation)

Throughout this document, the words that are used to define the significance of particular requirements are capitalized.  These words are:

"MUST"           This word or the adjective "REQUIRED" means that the item is an absolute requirement of this specification document.

"MUST NOT"    This phrase means that the item is an absolute prohibition of this specification document.

 "SHOULD"      This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but

the full implications should be understood and the case carefully weighed before choosing a different course.

"SHOULD NOT" This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before choosing a different course.

"MAY"            This word or the adjective "OPTIONAL" means that this item is truly optional.   One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

## 1.5  Revision History

| Version | Date | Author | Remarks |
|---|---|---|---|
| D01 | 15 May 2020 | Comcast | Initial Version |

## 2  References

Reasonable effort is made to keep references up to date with respect to versions and release dates, however manufacturers are responsible for ensuring they have the most recent version of a reference specification (unless otherwise noted).

Where conflicts exist between requirements contained in this specification and normative references, the specification requirements govern.

### 2.1  Normative References

[BLUETOOTH] Bluetooth Core Specification version 4.0 or later

# 3  Terms and Definitions

This document uses the following terms and definitions.

**Table 2 - Terms and Definitions**

| Term | Definition |
|------|------------|
|      |            |
|      |            |

# 4 Abbreviations and Acronyms

This document uses the following abbreviations and acronyms.

**Table 3 - Abbreviations and Acronyms**

| Abbrv | Acronym |
|-------|---------|
| RIS | RDK IR Service |

**Commented [SM1]:** Update for IR

# 5   Introduction

The RDK IR Service (RIS) exposes data and associated formatting for programming infrared signals into an RDK Remote Control Device or other device with an IR transmitter.

## 5.1  Conformance

All capabilities indicated as mandatory for this Service shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated.

## 5.2  Service Dependency

This service is not dependent upon any other services.

## 5.3  Bluetooth Specification Release Compatibility

This specification is compatible with any Bluetooth core specification][BLUETOOTH] that includes the Generic Attribute Profile (GATT) specification and the Bluetooth Low Energy Controller specification.

**Commented [SM2]:** references

## 5.4  GATT Sub-Procedure Requirements

Requirements in this section represent a minimum set of requirements for an RDK Remote Control Device (GATT Server). Other GATT sub-procedures may be used if supported by both Client and Server.

Table 4 below summarises additional GATT sub-procedure requirements beyond those required by all GATT Servers.

**Table 4 - GATT Sub-Procedure Requirement**

| GATT Sub-Procedure | Requirement |
|---|---|
| Read Characteristic Value | M |
| Write Characteristic Value | M |
| Write Without Response | O |
| Queued Write Characteristic Value | C.1 |
| Read Characteristic Descriptors | M |
| Write Characteristic Descriptors | M |
| C.1 Queued writes are required if infrared data is larger than 20 octets. | |

## 5.5  Transport Dependencies

The service shall only operate over an LE transport.

## 5.6 Error Codes

This service does not define any application error codes that are used in Attribute Protocol.

## 5.7 Byte Transmission Order

All characteristics used with this service shall be transmitted with the least significant octet first (i.e., little endian).

# 6   Service Requirements

## 6.1   Service Declaration

The service UUID shall be set to:

```
0000F801-BDF0-407C-AAFF-D09967F31ACD
```

## 6.2   Service Roles

A remote control or similar low power device enabled with an IR transmitter should function as an RIS Server.

A settop box or other host capable of supplying IR data should function as an RIS Client.

## 6.3   Service Introduction

Figure 1 depicts the characteristics and descriptors required of the service and the primary direction of data flow from / to the RDK STB host device.

RIS Client        RIS Server
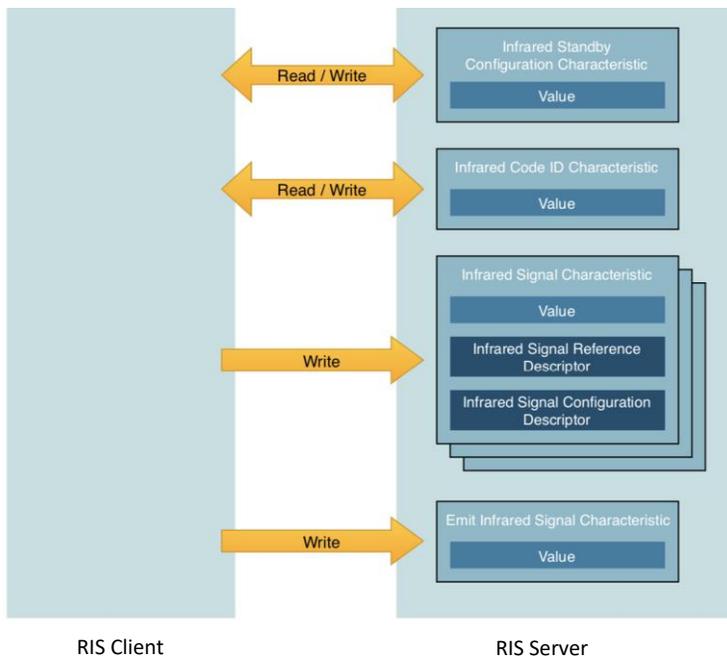
**Figure 1 - RDK IR Service**

## 6.4  Service Sequence Examples

TBD

## 6.5  Characteristic Overview

The RDK IR Service is composed of the following characteristics used to provide access to the infrared signal capabilities and data.

Unless otherwise specified, only one instance of each characteristic is permitted within an RDK IR Service.

**Table 5 - Voice Service Characteristics**

| Characteristic Name | Requirement | Mandatory Properties | Optional Properties | Security Permissions |
|---|---|---|---|---|
| IR Standby Configuration | M | Read, Write, Write Without Response | | None |
| IR Code ID | M | Read, Write, Write Without Response | | None |

Commented [SM3]: Should this be optional?

| IR Signal | C.2 | Write, Write Without Response | Read, Queued Write | None |
|-----------|-----|-------------------------------|--------------------|------|
| Emit IR Signal | M | Write, Write Without Response | Read | None |

C.2: Mandatory to support at least one Infrared Signal.

Notes:

- Security Permissions of "None" means that this service does not impose any requirements.

- Profiles utilising this Service may impose security requirements beyond those defined in Table 2.1 for all characteristics defined in Table 2.1.

- Properties not listed as mandatory (M) or optional (O) are excluded.

## 6.6  IR Standby Configuration Characteristic

The *IR Standby Configuration* characteristic controls the behaviour of an RIR Client when the physical standby button is pressed and the RIR Client is not connected but is bonded to an RIR Server. The purpose is to control the events (IR and BLE) used to bring an RIR Server out of standby.

The UUID of the IR Standby Configuration characteristic is:

$$0000EB01-BDF0-407C-AAFF-D09967F31ACD$$

A single instance of this characteristic shall exist as part of the RIR Service.

### 6.6.1   IR Standby Configuration Characteristic Behavior

The *IR Standby Configuration* characteristic value can be read using the GATT Read Characteristic Value and is written using the GATT Write sub-procedure.

### 6.6.2  IR Standby Configuration Characteristic Value

The characteristic contains a single octet, table 2.2 shows the allowed values.

**Table 6 - IR Standby Configuration Possible Values**

| Value | Description |
|-------|-------------|
| 0x00 | IR Fallback mode when not connected |
| 0x01 | BLE directed advertising mode when not connected |
| 0x02-0xFF | Reserved for future use |

This characteristic value shall be persistent across connections for bonded devices. The *IR Standby Configuration* characteristic is unique for each RIR Client. An RIR Client may read and write this descriptor to determine and set the configuration for that client.

The default value for the *IR Standby Configuration* characteristic is `0x00`. Upon connection of non-bonded clients, this characteristic value is set to the default value.

If the RIR Server stores client information for multiple clients then it is the last connected client's value that shall be used to determine the standby key mode.

## 6.7  IR Code ID Characteristic

The IR Code ID characteristic is used to expose the current *IR code set id* of the RIR IR Service with which it is associated, or to set the desired *code set id* of the Service.

> **Commented [SM4]:** Unclear what this actually means

The UUID of the IR Code ID characteristic is：

<div align="center">

`0000EB02-BDF0-407C-AAFF-D09967F31ACD`

</div>

Only a single instance of this characteristic shall exist as part of the RIR Service.

### 6.7.1  IR Code ID Characteristic Behaviour

Both IR code set id values are arbitrary signed 32-bit values that are written and read by the RIR Client. Their values shall not be interpreted or used by the RIR Server; the values shall be treated as opaque data.

The IR Code ID Characteristic value shall be maintained across power cycles and device bonding / un-bonding (i.e. the value should be stored in non-volatile memory and not linked to a bound client). The default shall be set to -1 by the service.

### 6.7.2  IR Code ID Characteristic Value

The characteristic shall contain two 32-bit signed integer fields, one for the *code set id* of a TV and one for an AV amplifier.

Table 7 shows the data format of the characteristic value.

<div align="center">

**Table 7 - IR Code ID Characteristic Values**

</div>

| Name | Requirement | Format |
|---|---|---|
| TV IR Code ID | Mandatory | sint32 |
| AV Amplifier IR Code ID | Mandatory | sint32 |

*Table 2.3: Infrared Code ID characteristic value*

## 6.8   IR Signal Characteristic

The IR Signal characteristic is used to set the infrared signal data for a single physical button or function on an RIR Server.

The UUID of the Infrared Signal characteristic is：

```
0000EB03-BDF0-407C-AAFF-D09967F31ACD
```

Multiple instances of this characteristic shall exist as part of the RIR Service. There shall be single instance of this characteristic for each physical button or function that supports infrared programming.
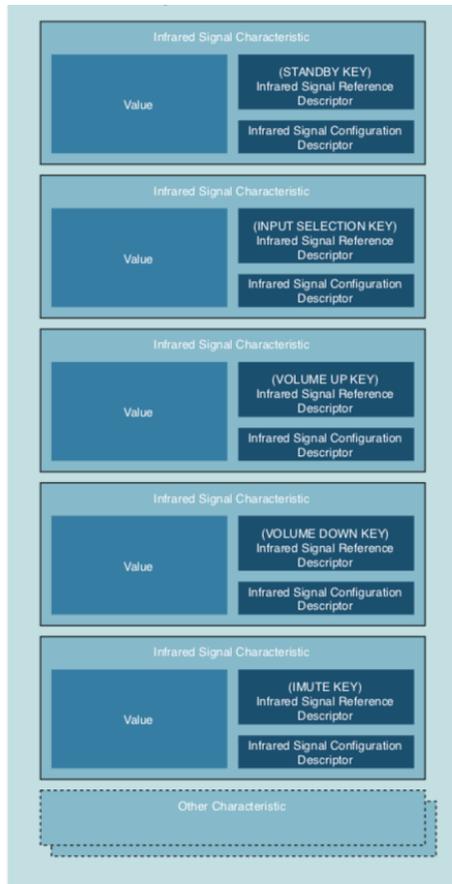
### 6.8.1 IR Signal Characteristic Behavior

The IR Signal characteristic is used to store infrared format information such that the RIR Server can emit the given infrared signal when a physical button is pressed or an event happens.

Each characteristic shall have a single instance of the *IR Signal Reference* descriptor and a single instance of the *IR Signal Configuration* descriptor. The *IR Signal Reference* descriptor shall be used to identify which physical button the characteristic represents.

### 6.8.2  IR Signal Characteristic Value

The Infrared Signal characteristic value shall be variable length octet array, it's format is outside of the scope of this document. It is expected that the RIR Client will receive an opaque blob of data from a vendor supplied infrared database. The complete blob of data will be written to this characteristic value.

On any Infrared Signal characteristic value write the enabled state of the characteristic shall be set to `FALSE` and a read of the *Infrared Signal Configuration Descriptor* shall indicate the signal is disabled.

The RIR Server will enable the signal with a write to the *Infrared Signal Configuration Descriptor* once it has completed the write to the Infrared Signal characteristic value. Figure 2.2 depicts the expected sequence for programming an infrared signal.
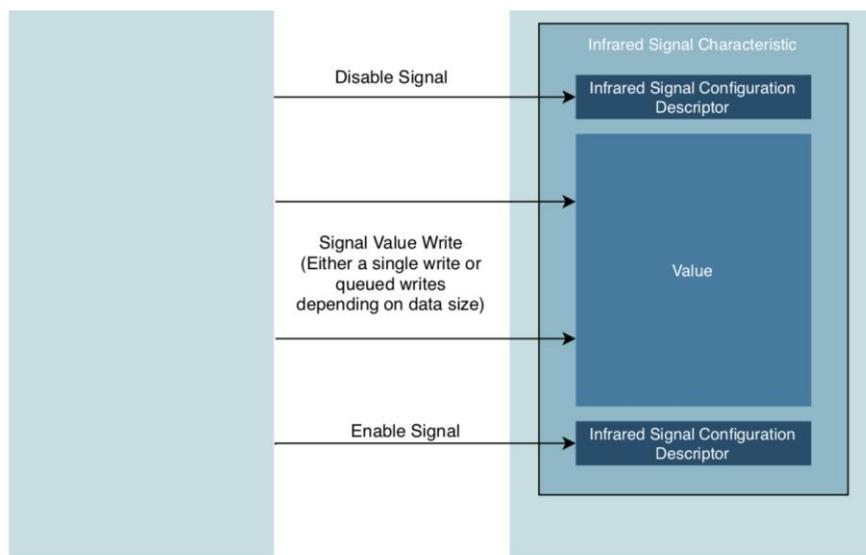


#### 6.8.2.1    Large IR Signal Data Characteristic Values

If writes of larger than ATT_MTU - 3 (20 octets) are required, then the device shall support *Queued Writes* as described in *Bluetooth Core v4.0, Volume 3, Part F, Section 3.4.6.*

If queued writes are used then the device shall support at least one buffer of a size large enough to store a single infrared signal blob. The RIR Server will guarantee that only one infrared signal characteristic is written at a time. A device may reject a queued write request if a queued write on another characteristic is in progress.

### 6.8.3  IR Signal Characteristic Descriptors

#### 6.8.3.1    IR Signal Reference Descriptors

The Infrared Signal Reference characteristic descriptor is used to provide the physical button identifier for the Infrared Signal characteristic value.

The UUID of the Infrared Signal Reference descriptor is:

> 0000EB04-BDF0-407C-AAFF-D09967F31ACD

The descriptor shall contain a single read only octet attribute. Table 2.2 shows the definition of the Infrared Signal Reference characteristic descriptor Attribute Value field.

| Value | Description | Notes |
|---|---|---|
| 0xFF | Reserved | NA |
| 0x0C | Standby Key | |
| 0x29 | Input Key | |
| 0x10 | Volume Up Key | |
| 0x11 | Volume Down Key | |
| 0x0D | Mute Key | |
| 0x5C | Select Key | |
| 0x58 | Up Key | |
| 0x59 | Down Key | |
| 0x5A | Left Key | |
| 0x5B | Right Key | |
| All Other Values | Reserved | |

#### 6.8.3.2    IR Signal Configuration Descriptor

The IR Signal Configuration characteristic descriptor is used to expose the signal enable / disable setting of the infrared signal.

The UUID of the Infrared Signal Configuration descriptor is:

> 0000EB05-BDF0-407C-AAFF-D09967F31ACD

The descriptor shall contain a 1 octet readable and writeable attribute. Table 2.3 shows the definition of the Infrared Signal Configuration characteristic descriptor Attribute Value field.

| Value | Description |
|-------|-------------|
| 0x00 | IR Signal Disabled |
| 0x01 | IR Signal Enabled |

The RIR Server may choose to not enable the given signal if it detects that the current Infrared Signal Characteristic value is invalid. In such cases a read of the attribute shall indicate the signal is still disabled.

## 6.9  Emit IR Signal Characteristic

The Emit Infrared Signal characteristic is used to request that the service emit an infrared signal.

The UUID of the Emit Infrared Signal characteristic is：

```
0000EB06-BDF0-407C-AAFF-D09967F31ACD
```

A single instance of this characteristic shall exist as part of the RIR Service.

### 6.9.1  Emit IR Signal Characteristic Behavior

Either the GATT Write or GATT Write Without Response sub-procedure is used to write to the Emit Infrared Signal characteristic.

A write shall trigger an infrared signal to be emitted for a period of 200 milliseconds. If an infrared signal is programmed and enabled via the Infrared Control Point Characteristic then that shall be used for the signal. If an infrared signal is not programmed or disabled then the default infrared signal shall be emitted.

The signal shall be immediately stopped if another GATT write to this characteristic is received.

A GATT Read is optional, if implemented it shall return the current signal being emitted, this should only apply to signals being emitted that were triggered by a write to this characteristic (i.e. an infrared signal emitted in response to a physical button press shall not be indicated in the value read).

### 6.9.2  Emit IR Signal Characteristic Value

The Emit Infrared Signal characteristic value is a single octet containing an enumeration of values as shown in Table 2.7.

| Value | Description |
|---|---|
| 0xFF | No Signal |
| 0x0C | Standby/Power Toggle Key |
| 0x29 | Input Key |
| 0x10 | Volume Up Key |
| 0x11 | Volume Down Key |
| 0x0D | Mute Key |
| 0x5C | Select Key |
| 0x58 | Up Key |
| 0x59 | Down Key |
| 0x5A | Left Key |
| 0x5B | Right Key |
| 0xE0 | Power On |
| 0xE1 | Power Off |
| All Other Values | Reserved |

A GATT write of `0xFF` can be used to cancel a previous request to emit an infrared signal.