BLE RDK Firmware Update Service Specification

RDK-SP-BLE-RFU-Service-D01-200605

Document Status: Draft

June 5, 2020

Document Status

Document Control Number:	RDK-SP-BLE-RFU-Service-D01-200605
Document Title:	BLE RDK Firmware Update Service Specification
Versions:	D01 – June 5, 2020
Date:	June 5, 2020
Status:	Document Status: Draft
Distribution:	

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 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: This specification does **not** include headend diagnostic screens.
Boldface & Uppercase	Used to emphasize information and for readability. For example: **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: *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: MUST, SHOULD, and MAY.

1.2 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.

1.3 Revision History

Version	Date	Author	Remarks
D01	5 June 2020	Comcast	Brought into RDK format

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.

1.4 Normative References

[BLUETOOTH] Bluetooth Core Specification version 4.0 or later

2 Terms and Definitions

This document uses the following terms and definitions.

Table 2 - Terms and Definitions

Term	Definition

3 Abbreviations and Acronyms

This document uses the following abbreviations and acronyms.

Table 3 - Abbreviations and Acronyms

Abbrv	Acronym
ADPCM	Adaptive Differential Pulse-Code Modulation
RIS	RDK IR Service

4 Introduction

The RDK Firmware Update Service is used to program binary firmware images from an RDK Set Top Box into an RDK Remote Control Device.

4.1 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.

4.2 Service Dependency

This service is not dependent upon any other services.

4.3 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 [SM1]: references

4.4 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	O
Write Without Response	M
Notification	M
Read Characteristic Descriptors	M
Write Characteristic Descriptors	M

4.5 Transport Dependencies

The service shall only operate over an LE transport.

4.6 Error Codes

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

4.7 Byte Transmission Order

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

5 Service Requirements

5.1 Service Declaration

The service UUID shall be set to:

```
0000F802-BDF0-407C-AAFF-D09967F31ACD
```

5.2 Service Introduction

Error! Reference source not found. depicts the characteristics and descriptors required of the service and the primary direction of data flow from / to the RDK STB host device.

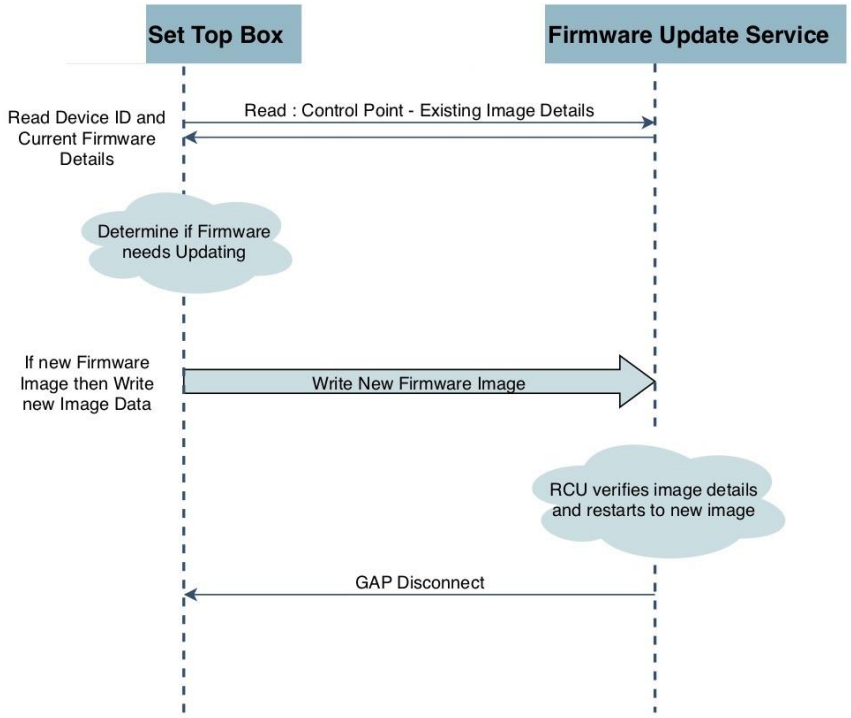


Figure 1: RDK Firmware Update Service Typical Sequence

Two characteristics are provided, one exposes information on the current firmware image loaded / being executed. The second characteristic is used for the bulk data transfer of firmware images.

The firmware image data transfer service is modelled on the TFTP protocol RFC1350 with the window size extension RFC7440. However, the packet format has been modified to be optimized for transfer over Bluetooth LE.

5.3 Characteristic Overview

The RDK Firmware Update service exposes one instance of the characteristics listed in the following table.

Characteristic Name	Requirement	Mandatory Properties	Security Permissions
OTA Control Point	M	Read	TBD
OTA Packet	M	Write With Response, Notify	TBD

Table 5: RDK Firmware Update Characteristics

Notes:

- Security Permissions of “None” means that this service does not impose any requirements.
- Profiles utilizing this Service may impose security requirements beyond those defined in Table 2.1 for all characteristics defined in Table 5.
- Properties not listed as mandatory (M) or optional (O) are excluded.

5.4 OTA Control Point Characteristic

The Control Point characteristic is used to expose a hardware identifier and the current firmware image details.

Only a single instance of this characteristic shall exist as part of the RDK Firmware Update Service.

The characteristic UUID shall be set to:

0000EC01-BDF0-407C-AAFF-D09967F31ACD

5.4.1 OTA Control Point Characteristic Value

A read of the OTA Control Point characteristic shall return details for the current firmware image that is executing.

Table 2.3 shows the data format of the characteristic value, all values shall be in little endian order.

Name	Requirement	Format

Device Model Identification	Mandatory	uint32
Firmware Version	Mandatory	uint32
Firmware CRC32	Mandatory	uint32

Table 6: OTA Control Point Characteristic Value

The Device Model Identification value comprises of the fields detailed in Table 7.

Bits	Description
31 - 24	Manufacturer Identifier
23 - 16	Major Version Number
15 - 8	Minor Version Number
7 - 0	Micro Version Number

Table 7: Device Model Version Fields

The Firmware Version comprises of the fields detailed in table 2.5. On a Read operation this shall return the version of firmware currently executing.

Bits	Description
31 - 16	Major Version Number
15 - 8	Minor Version Number
7 - 0	Micro Version Number

Table 8: Firmware Version Fields

The Firmware CRC32 field contains the CRC-32 of the currently executing firmware, in little endian octet order.

5.4.1.1 CRC-32 Algorithm

The standard CRC-32 algorithm shall be used for checking the entire firmware image, it's polynomial is defined as $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$.

5.5 OTA Packet Characteristic

The OTA Packet characteristic is used for the bulk data transfer of the firmware image. Only a single instance of this characteristic shall exist as part of the RDK Firmware Update Service.

The characteristic UUID shall be set to:

0000EC02-BDF0-407C-AAFF-D09967F31ACD

5.5.1 OTA Packet Characteristic Descriptors

5.5.1.1 Client Characteristic Configuration Descriptor

A Client Characteristic Configuration descriptor shall be included in the OTA Packet characteristic.

5.5.1.2 OTA Packet Window Size Descriptor

An optional OTA Packet Window Size descriptor may be included in the OTA Packet characteristic, if present it defines the size of the transfer window used for all transfers, this is the value **N** described in section ##.

If present the descriptor shall contain a single read only octet value, the value shall be constant for a given bonded client.

Name	Requirement	Format	Minimum Value	Maximum Value
OTA Packet Window Size	Mandatory	uint8	1	255

The descriptor UUID shall be set to:

0000EC03-BDF0-407C-AAFF-D09967F31ACD

5.5.2 Overview of the OTA Packet Protocol

The transport protocol is based on the TFTP protocol (RFC1350), however packet formats have changed to pack in more data for the (effective) 20 byte MTU limit for GATT Write operations and GATT Notifications. In addition, only the TFTP Write Request (WRQ) mode is supported, it is not possible to read a firmware image back from a RDK RCU device.

Commented [SM2]: Can base this on the negotiated MTU

A firmware image upload starts with a request to write (WRQ) packet, the packet contains the length of the image along with the image version and CRC. The WRQ packet is transferred as a GATT *Write Without Response* operation. If the RDK RCU is ready to receive an image it shall send back an acknowledgment packet (ACK) as a GATT *Notification*. An error packet (ERROR) may be returned also as a GATT *Notification* if the RDK RCU cannot service the request.

If an image upload was already in progress when a WRQ packet was received, the current upload shall be cancelled and the RDK RCU shall return to the initial state.

Once the WRQ packet has been acknowledged the RDK STB will send the first **N** number of data packets (DATA), each DATA packet contains a block number and 18 bytes of firmware image data. The block number shall start at 1 and increment for each new block of data, block numbers wrap back to 0 after reaching 16383 (214). DATA packets are sent with a GATT *Write Without Response* operation. When the RDK RCU device has received **N** number of DATA packets it shall send an ACK packet with the last block number that was received.

The firmware image upload operation shall be completed when a DATA packet is received with less than 18 octets of data; if a firmware image is an exact multiple of 18 octets then the last packet sent shall contain no data (i.e. just the two octet OpCode and block number header).

Upon receiving the final DATA packet the RDK RCU shall verify the image received by comparing the CRC-32 value from the initial WRQ packet, with the value it calculated over the data it received, if the values match it should immediately activate the image. If the CRC-32 values do not match or the device detects another error that prevents an image activation, then the final DATA packet shall not be acknowledged and instead an ERROR packet shall be sent.

The process by how a new firmware image is activated is not explicitly mandated by this specification, but it is expected that at some point the device will be rebooted and the connection will be lost and then restored.

Figure 2 below illustrates the sequence of an image transfer.

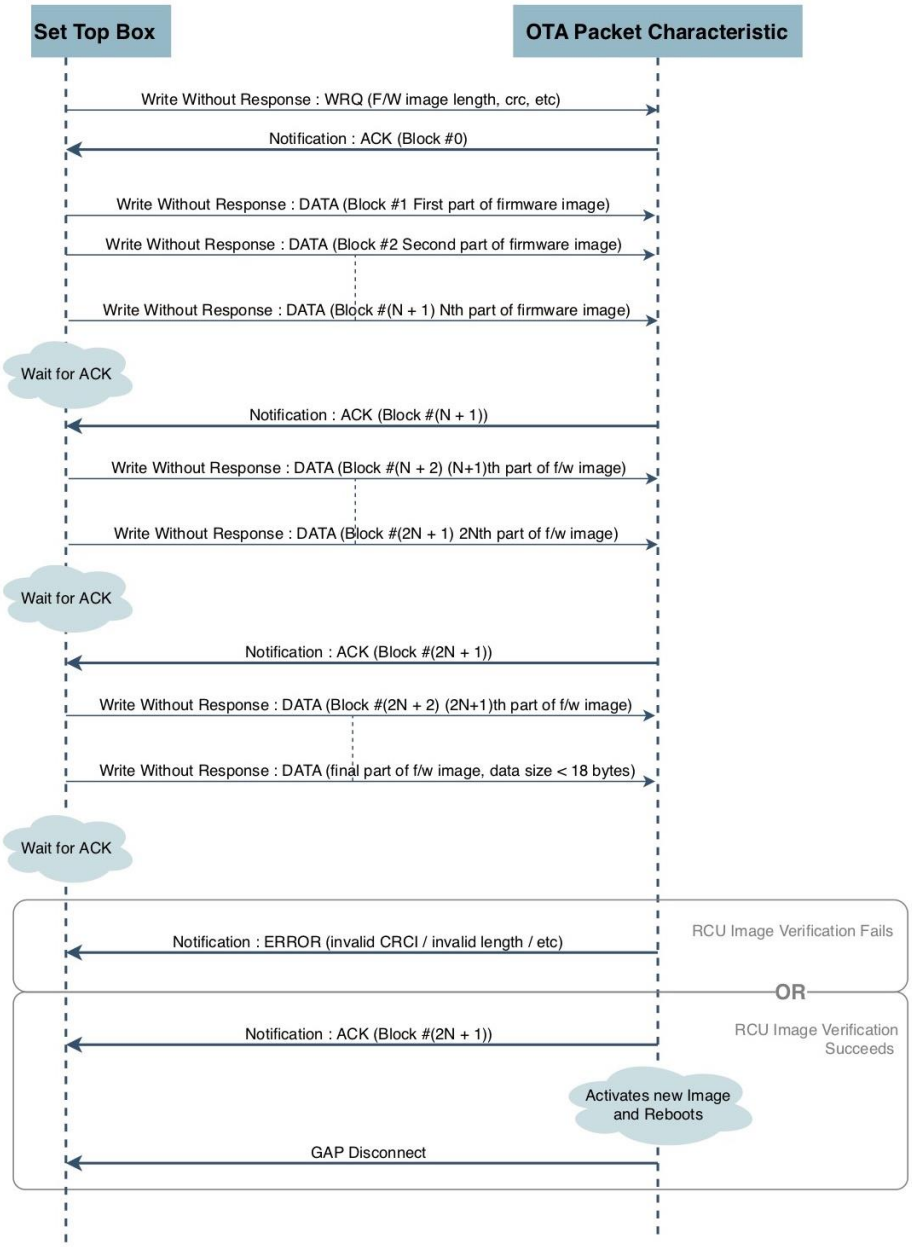
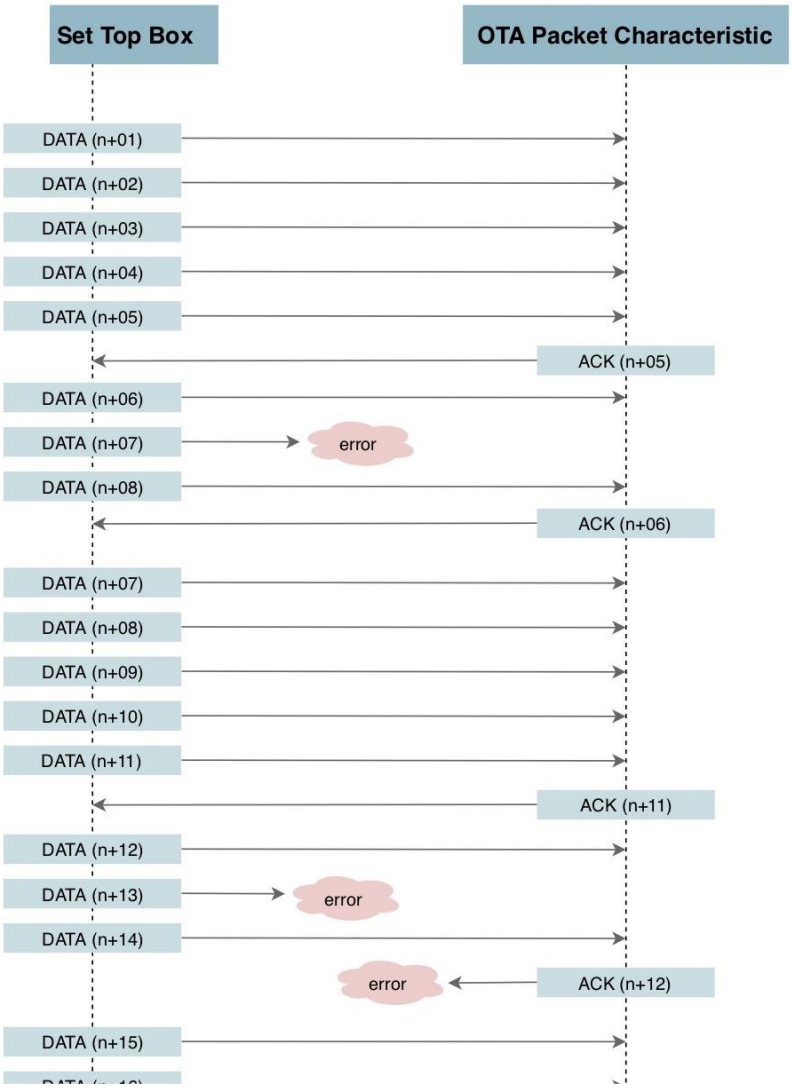


Figure 2: Image Transfer Sequence

The number of DATA packets sent before waiting for an ACK packet (**N** in the above diagram) defaults to a value 5, however it may be changed by the addition of the optional OTA Packet Window Size descriptor. DATA packets are transferred via GATT *Write Without Response* operations.

5.5.3 Traffic Flow and Error Handling

RFCs 1350 and 7440 shall be the master source for information on how to handle errors, unexpected packets, time-outs, etc. In particular the diagram on page 5 of RFC7440 is pertinent, a copy of it has been reproduced in figure 2.2 below.


Set Top Box
OTA Packet Characteristic
DATA (n+01)
DATA (n+02)
DATA (n+03)
DATA (n+04)
DATA (n+05)
ACK (n+05)
DATA (n+06)
DATA (n+07)
error
DATA (n+08)
ACK (n+06)
DATA (n+07)
DATA (n+08)
DATA (n+09)
DATA (n+10)
DATA (n+11)
ACK (n+11)
DATA (n+12)
DATA (n+13)
error
DATA (n+14)
error
ACK (n+12)
DATA (n+15)

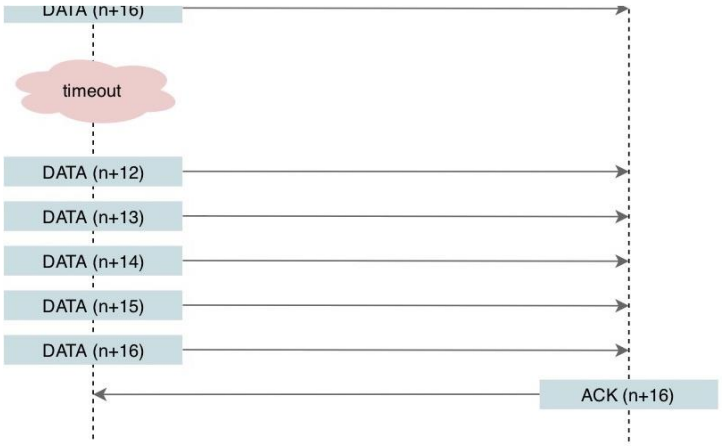


Figure 3: Firmware Update Sequence Diagram

5.5.4 Slave Latency Recommendation

To increase transfer speed it is recommended that the slave latency is temporarily reduced after the RDK RCU device sends an ACK packet, such that the subsequent DATA packet sent by the RDK STB device is processed quickly.

The slave latency shall be reduced for a maximum of 1 seconds after an ACK is sent.

5.5.5 OTA Packet Types

Table 9 shows the four packet types supported and their transport details.

Type	OpCode	Direction	GATT Operation
Write Request (WRO)	0x0	STB->RCU	Write Without Response
Data (DATA)	0x1	STB->RCU	Write Without Response
Acknowledgement (ACK)	0x2	STB<-RCU	Notification
Error (ERROR)	0x3	STB<-RCU	Notification

Table 9: OTA Packet Types

5.5.5.1 OTA Write Request Packet Format

Table 10 shows the format of the Write Request (WRQ) packet.

Size	Field	Value
2 bits	OpCode	0x0
14 bits	Reserved	0x000
4 octets	Firmware Image Length (in octets)	-
4 octets	Firmware Image Version	-
4 octets	Firmware Image CRC32	-

Table 10: OTA WRQ Packet Format

The 2-bit opcode is stored in bits 7 and 6 of the first octet.

All data fields are least significant octet first (i.e. little endian).

5.5.5.2 OTA Data Packet Format

Table 11 shows the format of the Data (DATA) packet.

Size	Field	Value
2 bits	OpCode	0x1
14 bits	Block Number	-
18 octets	Data	-

Table 11: OTA Data Packet Format

The 2-bit opcode is stored in bits 7 and 6 of the first octet.

The most significant 6 bits of the block number are stored in bits 0 to 5 of the first octet, the least significant 8 bits of the block number are stored in the second octet.

All packets shall have 18 octets of firmware image data, except the last DATA packet that terminates the operation.

5.5.5.3 OTA Acknowledgement Packet Format

Table 12 shows the format of the Acknowledgement (ACK) packet.

Size	Field	Value
2 bits	OpCode	0x2
14 bits	Block Number	-

Table 12: OTA Acknowledgement Packet Format

The 2-bit opcode is stored in bits 7 and 6 of the first octet.

The most significant 6 bits of the block number are stored in bits 0 to 5 of the first octet, the least significant 8 bits of the block number are stored in the second octet.

5.5.5.4 OTA Error Packet Format

Table 13 shows the format of the Error (ERROR) packet.

Size	Field	Value
2 bits	OpCode	0x3
6 bits	Reserved	0x00
1 octet	Error Code	-

Table 13: OTA Error Packet Format

The 2-bit opcode is stored in bits 7 and 6 of the first octet.

Table 2.11 lists the possible error codes that RDK RCU device may response with.

Value	Name	Description
0x01	Invalid CRC	The CRC-32 received in the WRQ packet doesn't match the image data received
0x02	Invalid Size	The size specified for the image in the WRQ packet is not supported
0x03	Size Mismatch	The size of the image data received doesn't match the value in the WRQ packet
0x04	Low Battery	The battery is too low to start a firmware upgrade
0x05	Invalid OpCode	Received opcode other than 0x00 (WRQ) and 0x01 (Data)
0x06	Internal Error	An unrecoverable error occurred, for example a flash write failed
0x07	Invalid Hash	The security hash check failed.