BV LGI Babylon Dual Stack 2020

BV LGI Sustainable Babylon 2023

Product Specification
23-November-2023

Rev 3.1

Group Part Number: R328451/ R328453

Variant	Part Number	Color Keys?
Ziggo	R328451A99-00001 R328451B99-00001 R328451C99-00001	No

	R328453B99-00001	
Telenet	R328451A99-00003 R328451B99-00003 R328451C99-00003	No
UPC Slovakia	R328451A99-00003 R328451B99-00003 R328451C99-00003 R328453B99-00003	No
Sunrise	R328451A97-00007 R328451B97-00002 R328451C97-00002 R328453B97-00001	No
Virgin Media	R328451A99-00004 R328451B99-00004 R328451C99-00004 R328453B99-00002	Yes

Revision History

Specification		Sect.	Update Description	By
Rev	**Date**			
Rev 0.1	16-April-2020	All	Initial revision based on Apollo spec 2.8 and NextGen spec 1.24-X12.	CP
Rev 0.2	19-May-2020	8.9.7.2 8.9.7.2.1	Set auto polling period to 30s.	CP
		8.7	Added details on what is not allowed at critical voltage.	
		5.3.8.1 5.3.8.2	Separated usage profile for RF4CE and BLE.	
Rev 0.3	27-May-2020	8.11.2.1	Added command 0xF5.	CP
		2.2	Fixed revision of spec0005.	
		5.3.4	Removed antenna gain section.	
Rev 0.4	29-May-2020	6.1	Added standard revision information.	CP
Rev 0.5	5-June-2020	4.5.2	Added COG section and updated COG picture.	CP
Rev 0.6	8-June-2020	8.11.2	Added more details on response to UAPI RCU model ID command.	CP
		8.3.3.10 8.3.3.11	Changed color of LED indication for software revision blinks.	
		8.6	Added reference to pairing state table in document attached section 8.6.6 and indicated pairing should start 5s after a combo key press.	
		6.1	Added note that product should comply latest official revision.	
		8.6.6	Added section and pairing state operation table.	
Rev 0.7	15-June-2020	8.9	Updated user string and comment. Removed note below table.	CP
Rev 0.8	23-June-2020	2.1	Updated LGI Mechanical and Quality requirements document.	CP
Rev 0.9	30-June-2020	8.6	Added reference RF4CE pairing section. Added pairing information is erased upon key combo detection. Added minor details in pairing state operation table.	CP

		3.1.1 8.8.6.3 8.8.11 8.8.15.2	Removed Vrestart virtual key section and replace usage of Vrestart virtual key by flag implementation.	
		8.7	Changed Vrestart threshold to 2.7V per SW team request. Stop all operations at Vcritical in RF4CE mode. Defined Vrestart detected flag.	
		4.5.4	Added Keypad Hardness definition section	
		8.11.2	Added UAPI command to enable/disable proximity sensor. Added UAPI command to read RCU flags. Added UAPI command to configure QS app ID.	
		8.10.3 8.10.4	Specified the LED color that is implemented depending on the operating mode.	
		8.8.15.2	Updated STB power keypress sequence with STB Wake Up flag. Added Tx of STB power key upon probe response timeout.	
		8.8.15.3	Added STB wake up flag section.	
		8.8.7.1	Changed stage 2 burst to 1.28s and burst gap to 300s.	
		8.6 8.8.9 8.9.4	Initiated a second pairing attempt automatically. No switch back on previous pairing info.	
		8.9.7.1	Added manual polling upon STB and TV power key.	
		8.8.1.3	Added BLE PHY configuration.	
		8.9.11	Added dual app ID usage in case of RF4CE.	
		8.9.3	Updated case of pairing an already paired RCU.	
Rev 1.0	28-July-2020	8.11.4.2	Update audio data format to match X10 Apollo implementation.	CP
		1.3.3.1	Added table for EOS v2.	
		5.3.7.2	Modified below table note and replaced all key release by specific key release and added reference on polling sections.	
		8	Removed LED activation upon proximity event.	

		8.1	Fixed sentence .
		8.2	Added OTA image is containing BLE and RF4CE applications and stacks.
		8.3.3.13	Fixed sentence.
		8.3.4.2 8.8.9	Indicated Pairing information and operating mode are erased before starting new pairing attempt. Removed the sentence part stipulating the pairing information is over written.
		8.9.4	Replaced BLE by RF4CE. Removed the sentence part stipulating the pairing information is over written.
		8.5.2	Added UK and replaced Apollo by Babylon.
		8.6	Second pairing attempt is only in case of BLE or interleaved pairing. If RCU is paired, then referred to manual pairing sections. Added section description details. Added reference of pairing state operation table.
		8.6.1	Removed pairing success and failure sentences because they are in already described in different sections.
		8.6.4	Added data for 0xEC in RCMM Protocol details table. Added indication figure 2 is for EOS STB.
		8.8.9	Indicated RCU erase pairing info and operating mode through a factory reset and added reference to 8.6.6 excel document.
		8.9.4	Indicated RCU erase pairing info and operating mode through a factory reset and added reference to 8.6.6 excel document.
		8.6.5	BLE_IR_ASSISTED_BURST_LENGTH set to 5 frames.
		8.8.15.3	Replaced OTV key by RF STB Power key.
		8.9.7.1	Removed TV power from the list of the key triggering a Get Pending Messages.
		8.9.13.1	Added LED feedbacks.
		8.9.13.2	Changed STB probe command trigger event to STB Power key press

		8.9.13.2	Changed STB probe command trigger event to STB Power key press instead of proximity event. Added upon key release get Pending Message is sent.	
		8.11	Replace Apollo by Babylon.	
		8.11.1.4	Replaced wrong section reference by the TV power key section for BLE.	
		8.11.2.3	Replaced proximity trigger event to STB power key press.	
		8.6.4.1	Removed IR assisted signal repetition and modified the length to 5 frames.	
		8.6.4.3	Updated BLE IR assisted to match 8.6.4.1.	
		8.11.4.2	Added sox command that should be used.	
		8.11.6.2	Set the OTA duration to 90s – value needs to be updated after we could run a test.	
		9.2.1	Added Test firmware should support RF4CE and BLE.	
		8.10.2	Removed sending unpair request and replaced by link disconnection.	
		8.11.6.1	Indicated foreground mode for RF4CE and background for BLE.	
		8.7	In BLE mode, RF and IR activities are now stopped at Vcritical.	
		8.8.2	Added SW number will be 7060 for all Apollo, Babylon and dual stack projects.	
		8.1	Added FW revision details.	
		N/A	Updated document name and header as per QA request.	
		5.5	Removed reference to 981 sequence.	
		8.11.2.1	Added default value of automatic polling and polling interval.	
		8.11.2.2	Added indication notifications are supported.	
		8.9.12	Created section 8.9.12.1 and moved UAPI Audio additional features to RF4CE section.	
		8.9.3	Removed IR keys are ignored while RCU is in pairing mode.	
		8.9.7.1	Added details on “Get Pending Messages” sending upon STB power key.	
		8.9.13.2	Added Power Key Sequence flow diagram.	

		14.2	Added table for RF4CE mode.	
	13-Aug-2020	8.5.2 6.1 4.1	Added LATAM and Chile requirement to certification and IR database.	
		8.9.1.1	RCU should filter only on device type and not on STB user string anymore. Removed STB User String from spec.	
		8.6.4.1	Added IR assisted signal is repeated at each. BLE_IR_ASSISTED_BURST_INTERVAL and updated figure. Added IR assisted signal should not be repeated after CONNECT_IND has been received.	
		8.6.4.2	Added IR assisted signal repetitions. Updated figure. Added IR signals stopped upon receiving discovery response.	
		8.6.4.3	Simplified wording and updated figure.	
		8.11.2.2	Indicated battery payload status format is 4 bytes and added table describing the format.	
	17-Aug-2020	8.11.2.1	Reworded 0xF7 message description and added Configuration Payload Format Table.	
		8.9.3	Removed reference to STB user string.	
	27-Aug-2020	8.2	Updated Memory Map.	
1.1	28-Aug-2020	8.9.11 8.9.11.2	Updated RF4CE Quickset App Id latching mechanism.	CP
		8.6 8.8.9	Mentioned second pairing attempt is initiated only if pairing failed before pairing timeout.	
		8.8.9 8.9.4	Reworded the case with more than 2 keys are pressed. Removed sentence about transmission of the additional keys since it cannot be guaranteed in 100% of the cases (3 keys max supported).	
		8.8.13	Added error code if wrong app Id is received in BLE mode.	
		8.10.1	Remove Quickset app id from feature reset.	

		8.10.1	Remove Quickset app id from feature reset.	
1.2	02-Sept-2020	8.11.2.1	Added default value for ProfileChangeTimeout. Proximity sensor is now disabled by default.	CP
		8.11.6.1	Removed sentence “STB will be responsible of configuring the OTA mode”.	
		8.8.1.2	PID set to 8140 in order to maintain a single manifest file.	
		8.11.2.1	UAPI product name set to 06E7.8140 in order to maintain a single manifest file.	
			OTA manifest example.	
		8.9.11	Quickset App Id unlatched in case of factory reset or incase RCU is unpaired.	
		1.3.2 3.1.1 8.8.6.3 8.8.15.4 8.10.1 8.11.2.1 8.11.6.1	Removed Graceful Shutdown Delay feature.	
		8.9.13.2	Added GetPendingMessages command after STB Power Probe Command. Updated flow chart. Added details in the description of STB state scenarios. Added keypress scenarios during STB power key sequence.	
		8.11.6.1 8.11.6.2	Changed foreground to background OTA in RF4CE	
		8.9.11	Added QS data is invalidated if AppID is requested to switch from 0x11 to 0x13.	

	15-Sept-2020	8.11.2.1	Stipulate 0xF3, 0xF4 is valid for BLE mode only. Stipulate 0xF9 is valid for RF4CE mode only. Fixed error code sent when 0xF3 is sent in RF4CE mode. Added error code sent when 0xF4 is sent in RF4CE mode. Added error code sent when 0xF2 is sent in BLE mode.	
		5.3.7.2	Removed STB Boot time fading from RF4CE usage profile.	
	18-Sept-2020	8.11.1.3	Reworded STB power key pressed while STB power sequence is ongoing in RF4CE mode.	
		8.9.13.2	Indicated section for keypress scenarios.	
		8.9.11	RCU latches upon QS message only when it is not latched.	
		8.1	X12 corresponds to 1.2.	
	21-Sept-2020	8.8.15.3	Removed case where flag is not reset upon authentication error because white list is enabled, and the case cannot occur.	
		N/A	Added UPC Swiss item number.	
1.3	3-Nov-2020	N/A	Fixed header name.	CP
		8.7	Allow RF and IR transmissions until Vstop in BLE mode.	
		8.11.2.1	Added standard opcode to UAPI message table. Added standard opcodes table.	
		5.3.7	Updated usage profile for RF4CE and BLE.	
		3.1.1 8.8.6.3 8.8.15.2	Reintroduced STB wake up virtual key.	
		8.1	X13 FW revision corresponds to spec revision 1.3.	
		5.3.2	Set Tx power to 6dBm for both BLE and RF4CE.	
1.4	7-Dec-2020	8.11.2.1	Updated 0xF5 standard opcode.	CP
		16.1	Added FIPS test reports.	
		8.7	Fixed IR/RF Tx columns for Vcritical threshold.	
		8.11.2.1	Changed references to Vrestart into Vrestart detected for consistency	

		8.11.2.1	Changed references to VrestArt into VrestArt detected for consistency.	
		8.8.15.3	Removed STB wake up flag reset upon state transition to disconnected.	
		8.8.15.2	Removed STB wake up flag reset upon state transition to disconnected from the power sequence state machine and from text.	
		4.4	Added weight info.	
1.5	11-Jan-2021	8.7	As a matter of clarity, added voice activity details at Vcritical in battery threshold table.	CP
	19-Jan-2021	8.9.7.1	Added battery level assessment details when Get pending message is sent and uapi polling is stopped when Vcritical is reached.	
		8.11.5	Updated proximity sensor part number.	
		8.8.11 8.11.2.2 8.7	Moved battery level assessment conditions into RF4CE/BLE common section. Added battery level is not assessed when proximity sensor is disabled. Added battery level can be reported upon a read request from STB.	
1.6	02-Feb-2021	3.1.1	Removed STB graceful power down from TV power key description.	CP
	15-Feb-2021	8.9.7.2	Added 100ms Rx window is applicable to auto polling	
	16-Feb-2021	8.8.2	Updated PID to 0x8140	
1.7	02-Mar-2021	8.11.2	Added standard opcode for 0xF9	CP
1.8	08-Apr-2021	8.9.7.2	Added auto polling behavior during voice and OTA sessions	CP
1.9	01-July-2021	1.1	Reworded after LGI review	CP
		2.1	Replaced word document by pdf	
		4.1	Added UKCA compliant	
		8.1	Added X18 correspondence	
		8.2	Updated memory map	
		8.6.1	Reworded sentence “once pairing is initiated…”	
		8.8.4.1.2	Removed 1st sentence of the section	
		8.8.6.2	Replaced Apollo by Babylon	

		8.8.9	Fixed typo. Added details if key combo is pressed more than 5s longer than pairing session.	
		8.8.13	Fixed grammar and indicates STB should set the key data in NVM	
		8.8.14	Removed “streaming is independent from MIC key state” which is not the case on this project. Added MIC key should be pressed to start the streaming.	
		8.9.1	Fixed typo	
		8.9.3	Removed “as supported by node”	
		8.9.7.2	Stipulated OTA data request	
		8.9.8	Added references to 8.9.9 an 8.9.10	
		8.9.11	Added sections numbers related to App Id 0x11	
		8.9.12.1	Added audio capture starts at 1st audio state message	
		8.9.13.3	Added TV power toggle signal is sent	
		8.9.10.1	Fixed grammar.	
		8.11.1.3	Updated with OTV OFF and OTV ON	
		8.11.5	Added details on activation command	
		8.11.6.2	Indicated POR activates the new image	
		12	Updated version 1.1 of the manifest and detailed update_flag Updated major and minor to Babylon	
2.0	20-July-2021	1.3.2	Added STB should not repeat MIC and STB power key.	CP
		8.8.4.1.2	Deleted the first sentence to avoid confusion	
		8.9.12.3	Created section to specify X19 flexible RF4CE audio	
2.1	23-July-2021	1.3.3.2	Added Humax EOS STB as Host device	CP
2.2	1-Sept-2021	8.1	Added X19 in the FW rev table	CP
		8.4.2	Added SFF and LFF for reference	
2.3	16-Nov-2021	8.11.6.3	Added section to cover OTA with Selene ECOs	CP

		8.9.3	Rephrased to mention factory reset is issued at the beginning of pairing procedure	
		8.1	Added X21	
		8.3.4.2	Indicated key combo triggers a factory reset	
		8.7	Added UAPI polling battery level assessment case	
2.4	22-Nov-2021	8.4.2	Added model Id 3 for identifying Added Hw info definition Added Booster/Charge Pump configuration	CP
		8.11.6.4	Added section detailing how firmware guarantee the proper external flash driver selection	
	31-Jan-2022	8.1	Updated Table with X24	
2.5	31-Jan-2022	8.9.13	Added case where RF4CE ACK is not received when STB is in cold standby	CP
		8.1	Updated table with X25	
2.6	24-Mar-2022	8.4.2	Added release version where HW version format has changed. HW version resides at address 0x407FE02 and not HW Info. Fixed WLCP into WLCSP. Added how to retrieve HW configuration for units reporting HW version 0x00	CP
		16.2	Added hardware configurations description	
2.7	07-Mar-2023	Title	Updated title and part numbers	CP
		10	Added sections related to Babylon green light	
		1.1	Added reference on Babylon green light section	
2.8	25-Apr-2023	8.10.2.1	Updated all W parameters in R/W Updated handling of 0xF7 to/from RAM from X26 release STB Probe Value = 0x00 is standby only (not Off because STB does not respond to probe when off)	CP
		8.7.7.2	Added a key press during stage 2 should restart stage 1 Removed “triggered by STB”	

		10	Replaced Babylon green light wording by “sustainable Babylon” Added CMFs section	
		1.1	Replaced Babylon green light wording by “sustainable Babylon”	
		8.1	Updated table with X26	
		8.3.3.1	Added number of white blinks after activating an OTA	
		8.2	Updated memory map with no text	
		8.5.2	Added Ireland, Slovakia and Poland	
		8.10.1	Indicated the flowchart is an example	
		8.3.3.1	Added 2 blinks after an OTA	
2.9	03-Oct-2023	2.1 7.2	Updated packaging transportation for SUPF	CP
		10.3	Updated Sustainable CMFs	
		12.1.1	Replaced T Mobile by UPC	
		12.1.2 12.1.4	Updated CMF	
		12.1.3	Replaced Telenet by Sunrise	
		8.5.4 8.5.5 8.5.6 8.5.7 8.8.3 8.8.4 8.3.4	Updates related to new pairing implementation which should switch back to the previously paired information Removed 2 pairing attempts in case of BLE	
		8.5.10	Updated Dual Stack Pairing State operation Matrix document as per pairing feature ECO	
3.0	06-Oct-2023	8.3.4.2 8.5.4 8.5.5	Removed <<LIVE TV + 1>>, <<LIVE TV + 2>>, <<LIVE TV + 3>>, <<LIVE TV + 4>>, <<LIVE TV + 5>>. Removed wording indicating RCU should delete pairing key when	CP

		8.5.6 8.5.7 8.5.10 8.7.9 8.8.4	pairing failed. Removed IR assisted feature disabled scenario.	
3.1	23-Nov-2023	8.3.4.2	Removed pairing success and indication IR is enabled from the table Added reference to section 8.5.4 for <<LIVE TV +0>>	CP
		8.5.4	Fixed missing word Removed duplicated information in case of pairing failure.	
		8.7.9	Removed end of section note Added case where pairing failed	
		8.5.10	Extracted pairing state operation table from excel file	
		13	Updated manifest file info to format 1.2	

Approvals

Organization	*Name*	*Title*	*Date*
Universal Electronics b.v.	Bram van de Laar	Manager Application Engineering	
Universal Electronics b.v.	Christophe Porcel	Senior Field Application Engineer	
Liberty Global b.v.	Aaldrik Haaijer	STB Architect	

Table of Contents

1 Introduction

1.1 Purpose & Description

This specification covers the dual stack implementation merging in a single implementation the features of the Apollo RCU which is a BLE technology remote control and of EOS/Nextgen which is a RF4CE remote control.
The target product of the implementation can be either Apollo product or Babylon product that can support the dual stack firmware.
This specification will focus on Babylon product implementation. For further details on Apollo product please refer to Apollo BLE product Specification.
Note: in Liberty Global terminology, there are two UEI Babylon RCU's: the Small Form Factor (SFF) and the Large Form Factor (LFF). In UEI terminology, "Apollo product" is the same as Babylon SFF and "Babylon product" is the same as Babylon LFF.

Sustainable Babylon product specifics are covered in section 10.

1.2 Product Feature Summary

- UEI UE878NMEH, 512kB Flash, 64K RAM
- White and Red LED with light guide in top right corner.
- BLE 4.2
    - HID over GATT (HOGP)
    - Battery Service
    - Voice: 16kHz, 16 bits, ADPCM
    - Background OTA
    - Quickset support

    - Data Length Extension Support
    - UAPI and BLE standard battery services
- RF4CE
    - ZRC 2.0 and ZRC 1.1
    - Background OTA
    - Voice: 16kHz, 16 bits, ADPCM
    - Quickset support
    - UAPI battery service
- 34+4 clickable keys with metal domes
    - Including side key for profile selection
- External flash: 512Kb for dual stack implementation
- Proximity sensor with the option to enable/disable it from the STB
- IR enabling Quickset support and IR assisted pairing
- Quickset features
    - One touch view
    - Auto Discovery
    - Brand/Model Search

1.3 Host Device (STB)

1.3.1 Host Throughput Requirements

1.3.1.1 Voice

*For Linux (16kHz): (1 packet / 2.5ms) * Connection Interval (11.25ms) = 4.5 packets per Connection Interval, sustained average throughput for 16kHz ADPCM. Double it (9) packets per Connection Interval) for the peak throughput requirement.*

16kHz, 16 bits Voice over UAPI

A typical Linux-based host must support sustained average throughput of 4.5 packets per Connection Interval. Host should support a peak throughput of at least 9 packets per Connection Interval to allow enough capacity for retransmissions as required due to interference.

1.3.1.2 BLE – HID over USB

The HID packet Length over USB (HCI) is normally 64 bytes. For the DLE to be handled properly the driver needs to take care of the difference in packets size (ATT_MTU vs HID packet length). High speed endpoints are allowed a maximum packets data size 1024 bytes over USB (2.0+).

1.3.2 Other notes for STB Developers

This section will be used as a quick reference for Liberty Global STB developers.
In order to guarantee the RCU behave properly and features are full functional the STB should fit the following requirement list:

- For BLE connection/reconnection or OTV:
  - STB should be able to wake up from cold state by indirect advertising with WAKEUP data sequence.
  - STB should be able to connect from indirect advertising with WAKEUP data sequence even if in active state. This is required by OTV state machine.
  - STB should not connect from direct advertising if in cold standby state.
  - STB should gracefully initiate a RCU link disconnect before it goes to cold standby state. If STB does not disconnect and if STB power key is pressed, then the RCU will not be able to send the WAKEUP data sequence supposed to wake up the SoC of the STB before the link supervision timeout expired.
  - When RCU is not connected STB should maximize its BLE Scan activity (maximize scan window length and minimize scan interval) in order to reduce the connection latency.
  - STB should ignore long STB power key (preferred) and should not auto repeat the keycode.

- For Audio feature:
    - STB should accept the connection parameters suggested by the RCU at connection.
    - STB should ignore long MIC keypress and should not auto repeat the keycode.
    - STB should be able to handle an audio stop message from RCU in each of its audio streaming state.
    - STB should implement an audio streaming timeout reflecting the maximum allowed duration of an audio session.

- Other:
    - STB should prevent initiating an OTA activity if a QS session is ongoing
    - For BLE DLE please refer to section 8.7.12.
    - STB should store Long Term Key (LTK) in a file and a location that can be read from a terminal emulator.

1.3.3 Host Information

1.3.3.1 BLE STB

Item	Host Parameter	STB Model: Arris VIP5002W "Apollo"
1	STB Manufacturer	ARRIS
2	Main CPU (Chipset)	BCM72554
3	ARM/MIPS	ARM
4	32/64bits Support	32bits
5	BLE Chipset Manufacturer	Broadcom
6	BLE Chipset Part Number	BCM43570
7	BLE RF Stack Details – version, documentation on stack interface	Fluoride 9 (Arris specific)

8	OS (& version)	Linux-based: BCM Nexus
9	Middleware (if applicable)	LGI "One Middleware" RDK
10	Tool Chain Name, Version, Provider, MPU, Type (Little or Big Endian)	Little endian
12	ASR Engine	Nuance

Item	Host Parameter	STB Model: "EOS v2"
1	STB Manufacturer	Humax
2	Main CPU (Chipset)	BCM72180
3	ARM/MIPS	ARM
4	32/64bits Support	32bits
5	BLE Chipset Manufacturer	Broadcom
6	BLE Chipset Part Number	BCM4375 Integrated in SOC
7	BLE RF Stack Details – version, documentation on stack interface	Fluoride 9 (Arris specific)
8	OS (& version)	RDK-V (Linux-based)
9	Middleware (if applicable)	LGI "One Middleware" RDK
10	Tool Chain Name, Version, Provider, MPU, Type (Little or Big Endian)	arm-rdk-linux-gnueabi 5.3.0 – Little endian
12	ASR Engine	Nuance

1.3.3.2 RF4CE STB

Item	Host Parameter	STB Model: Arris DCX960 "EOS"
1	STB Manufacturer	ARRIS
2	Main CPU (Chipset)	BCM7252S

3	ARM/MIPS	ARM
4	32/64bits Support	32bits
5	RF4CE Chipset Manufacturer	GreenPeak
6	RF4CE Chipset Part Number	GP 501
7	RF4CE profile	ZRC 2.0
8	OS (& version)	N/A
9	Middleware (if applicable)	LGI “One Middleware” RDK
10	Tool Chain Name, Version, Provider, MPU, Type (Little or Big Endian)	Little endian
12	ASR Engine	Nuance

Item	Host Parameter	STB Model: Humax 1008C/R “EOS”
1	STB Manufacturer	HUMAX
2	Main CPU (Chipset)	BCM7252S
3	ARM/MIPS	ARM
4	32/64bits Support	32bits
5	RF4CE Chipset Manufacturer	GreenPeak
6	RF4CE Chipset Part Number	GP 502
7	RF4CE profile	ZRC 2.0
8	OS (& version)	N/A
9	Middleware (if applicable)	LGI “One Middleware” RDK
10	Tool Chain Name, Version, Provider, MPU, Type (Little or Big Endian)	Little endian
12	ASR Engine	Nuance

Item	Host Parameter	STB Model: Samsung “Selene”

1	STB Manufacturer	Samsung
2	Main CPU (Chipset)	Intel Groveland CLCE4277EDS0
3	ARM/MIPS	X86
4	32/64bits Support	32 bits
5	RF4CE Chipset Manufacturer	Radio Pulse
6	RF4CE Chipset Part Number	Radio Pulse
7	RF4CE profile	ZRC v1
8	OS (& version)	RDK-V (Linux)
9	Middleware (if applicable)	LGI “One Middleware”
10	Tool Chain Name, Version, Provider, MPU, Type (Little or Big Endian)	gcc
12	ASR Engine	Nuance

2 Reference Documentation

2.1 Customer Documentation

- LGI Mechanical and Quality Requirements

LGI Mechanical and Quality Requirements

Refer to Liberty Jira for Project Specific Requirements: https://jira.lgi.io

2.2 UEI Standard Documentation

Spec 0001, Rev 3, General Cosmetic Specification
Spec 0003, Rev 26, Mechanical Standard for Design and Testing of Hand-held Remote Control (HRC)
Spec 0004, Rev 9, Silicon Rubber Keypad, Poly-dome, and Metal Dome Specification
Spec 0005, Rev 4, Environmental Specification – Drop Test
Spec 0010, Rev 3, Marking
Spec 0020, Rev C, ESD
Spec 0080, Rev 1, GSM Immunity Test

2.3 Other Specifications

Specifications for BLE:

- Bluetooth – Core Specification v5.1

Specifications for RF4CE:

- ZigBee RF4CE: ZRC Profile Specification Version 2.0: Document 13-0442-23, Sept 4th, 2014.
- ZigBee RF4CE: Generic Device Profile Version 2.0: Document 13-0396r17ZB, Feb 24th, 2014.
- ZigBee ZRC Profile Action Banks, v1.0.0: Document 13-0614r07ZB, July 10th, 2014.

IR Specification:

- ZipIR specification 0.7 01-Oct-2018

RTC Specification:

- UEI RTC Whitepaper 2a – 18-Dec-2019

TSI Specification:

- TSI Whitepaper 1b 21-Nov-2018
- TSI API documentation: test_support_interface_0.3_Doc0.3_LGI_NDA.pdf

Customer Reference Sheet:

- CRSs

Relevant Specifications Hosted Online:

- UAPI Protocol Specification
- UAPI Over BLE
- UAPI Over ZRC

3 Key Description

3.1.1 Key Table

Below is the key table that has been defined as RF code set ID: C6247 for BLE and C5367 for RF4CE.

Key #	Key Labels	Description	BLE HID 0x0C	BLE HID 0x01	BLE HID 0x07	STB (RF4CE) [HDMI CEC action bank]	IR
1.	TV Power	TV Power Toggle	-	-			QS
2.	STB Power	STB Power (One Touch View On and One Touch View Off)	0x0030	-	-	0x6B	QS (OTV ON and OFF)
3.		Home	0x0223	-	-	0x09	-

Key #	Key Labels	Description	BLE HID 0x0C	BLE HID 0x01	BLE HID 0x07	STB (RF4CE) [HDMI CEC action bank]	IR
4.		EPG	0x008D	-	-	0x53	-
5.		Live TV	0x0089	-	-	0x10	-
6.	Up Arrow (D-Pad)	Up	0x0042	-	-	0x01	QS
7.	Left Arrow (D-Pad)	Left	0x0044	-	-	0x03	QS
8.	OK (D-Pad)	Select	0x0041	-	-	0x2B	QS
9.	Right Arrow (D-Pad)	Right	0x0045	-	-	0x04	QS
10.	Down Arrow (D-Pad)	Down	0x0043	-	-	0x02	QS
11.		Back	0x0224	-	-	0x32	-
12.		Context key	-	0x0084	-	0x0C	-
13.		Voice/Search	0x0221	-	-	0x2D	-
14.	Vol+	Volume Up	-	-	-		QS
15.	Vol-	Volume Down	-	-	-		QS
16.	CH+	Channel Up	0x009C	-	-	0x30	-

17.	CH-	Channel Down	0x009D	-	-	0x31	-
18.	Record	Record	0x00B2	-	-	0x47	-
19.	Rewind	Rewind	0x00B4	-	-	0x48	-
20.	Play/Pause	Play/Pause	0x00CD	-	-	0x44	-
21.	Fast Forward	Fast Forward	0x00B3	-	-	0x49	-
22.	1	1	-	-	0x001E	0x21	-
23.	2	2	-	-	0x001F	0x22	-
24.	3	3	-	-	0x0020	0x23	-
25.	4	4	-	-	0x0021	0x24	-
26.	5	5	-	-	0x0022	0x25	-
27.	6	6	-	-	0x0023	0x26	-
28.	7	7	-	-	0x0024	0x27	-
29.	8	8	-	-	0x0025	0x28	-
30.	9	9	-	-	0x0026	0x29	-
31.	0	0	-	-	0x0027	0x20	-
32.	Mute	Mute	-	-	-		QS
33.		TV Input	-	-	-		QS
34.	Side key	Profile	0x018D	-	-	0x2E	-

Key #	Key Labels	Description	BLE HID 0x0C	BLE HID 0x01	BLE HID 0x07	STB (RF4CE) [HDMI CEC action bank]	IR
35.	Red	Fastext Red			0x003A	0x72	-
36.	Green	Fastext Green	-	-	0x003B	0x73	-
37.	Yellow	Fastext Yellow	-	-	0x003C	0x74	-
38.	Blue	Fastext Blue	-	-	0x003D	0x71	-
39.	Virtual Key	Power Probe	0x003F	-	-	0x3F	-
40.	Virtual Key	STB WakeUp[1]		0x0083		[2]	

Table 1: Functional Key Table

[1] Refer to section 8.7.15.2 for further details on "STB WakeUp" virtual key usage.

[2] These virtual keys are not supported when RCU operates as an RF4CE device.

Quickset Enabled Keys

Key	External TV	External AMP
TV Power	x	-
STB Power (OTV)	x	x
Up	x	-
Down	x	-

Down	x	-
Left	x	-
Right	x	-
Select/OK	x	-
Vol Up	x	x
Vol Down	x	x
Mute	x	x
Av Input	x	-

Table 2: Quickset Enabled Key Table

4 Mechanical Requirements

4.1 Mechanical Information

Mechanical Information	
Information Description	Standard (Yes), Custom, No
1) Pro/E Shell Design Responsibility.	1) Liberty Global / D&E
2) IR Lens.	2) No
3) IR Lens Color e.g. dark blue; purple; dark red; dark grey; black.	3) N/A
4) Keypad Definition (Force versus Travel).	4) Standard Force is in the range 1.5N-4.5N
5) Cluster Key pad key cap vertical pull-off force	5) Minimum 1.5 kgf (vertical)
6) Keypad Printing/Coating (check with ME whether this is for painting or coating).	6) NO, but HOME, TV POWER, STB POWER and VOICE keys are UV printing oil.
7) Hard Cap Keys. a. Type (Hard Epoxy, Soft Epoxy, ABS Cap). b. Hard cap key Coating.	7) Yes, all a. ABS b. NO, but HOME, TV POWER, STB POWER and VOICE keys are are UV printing oil.
8) Backlight Diffuser Type and material i.e. Diffuser Plate – PC, LGF.	8) No Backlight

9) Metal Domes / Poly domes. a. Number and Location. b. If yes, then specify dome diameter size.	9) Metal Domes a. All b. 5mm
10) Programming/Debugging Opening On Plastics.	10) Yes
11) Battery Door Type (Specify – latch, sliding, push button, etc.).	11) Latch, sliding.
12) Painted Plastics. a. Top Case, Bottom Case, Door, Whole unit.	12)
13) Rubberized Coating. a. Top Case, Bottom Case, Door, Whole Unit.	13) N/A
14) TPE.	14) N/A
15) Special Finishes: NCVM, IML/IMD, metal parts.	15) N/A
16) Patent Numbers (Printed / Embossed).	16) N/A
17) Country of Origin Marking (Required for all products being shipped into USA).	17) Yes (For Puerto Rico) Laser engraving in Battery Area.
18) Tooling Inserts.	18) No
19) Compliance Marking Requirements (WEEE, CE, FCC, UL, etc.). a. Location (e.g. Battery area, bottom case).	19) CE, FCC, UKCA (including Chile) a. Laser engraving in Battery Area.
20) Plastic Inlay / Bezel.	20) Logo Plate, refurbishable.
21) Counter Weight (Certain customers may require this for balancing the remote design).	21) A provision for 20 grams of metal weight. COG shall be along the upper edge of the play/pause key. (Including Batteries.)

22) Type of fit between the top and bottom case.	22) Snap Fit.
23) Double Sided Adhesive	23) In digit area for metal plate.

4.1.1 CMF

This project includes two major skews: with and without teletext keys (on lowest row of remote). See renderings below for both default skews (without customer logo).
Different lower keypad with frame sub-assemblies and different logo plates will be used to build both skews.
For additional renderings with different customer derivative versions please refer to section 12.1.


Pure White
Pad printing
Matte
Logoplate printing - Pad printing
2x AAA Batteries
All keys metaldomes
1x Red LED
1x White LED
Nominal weight 100gr (including 2 x AAA battery)
NOTE: Colors on rendering are for reference only. Please refer to Pantone color book for true colors.
Laser parameters 3mm dot:
55W, 800mm/s, 20 kHz
BV LGI Babylon URC 2020
165 x 42,4 x 22 mm
1
A
30 April 2020


PANTONE
Topcase - No Paint
NCS S 9000-N
ABS
IR Translucent ABS
Bottomcase - No Paint
batterydoor - No Paint
Hard key caps - All keys
Double shot key
Chrome plated ring / NCS S 9000N core
Side-key cap - Includes internal rubber barrier
Lightguide
Milky White
Opaque white
Logoplate - Printed logo
Keypad printing - Pad printing
Pure White
Pad printing
Matte
Green PMS 354U
Yellow PMS 7404U
Blue PMS 2995U
Topcase printing - Pad printing
2x AAA Batteries
All keys metaldomes
1x Red LED
1x White LED
Nominal weight 100gr (including 2 x AAA battery)
NOTE: Colors on rendering are for reference only. Please refer to Pantone color book for true colors.
Microphone
OK
Pairing: press 5 sec
Laser parameters 3mm dot:
55W, 800mm/s, 20 kHz

Note that the engraved 3mm dot will be printed in the battery compartment reflecting the RCU is based on BLE technology.

4.2 Housing Material and Finish Details

Item #	Part	Material	Colour	Texture / Finish
1	Top case	ABS	NCS S 9000-N	Matte/ YS 20043-B
2	Bottom case	IR Transparent ABS	NCS S 9000-N	Satin/MT-11520 Gloss/SPI A-2 (side band)
3	Battery Door	IR Transparent ABS	NCS S 9000-N	Satin/MT-11520
4	Logo plate	ABS	NCS S 9000-N	Satin/ YS 20043-B
5	Light Guide	PC+RTP pigment	Milky White	MT-11007 outside SPI A-2 inside
6	Key Caps	ABS	NCS S 9000-N	Matte/ YS 20043-B
7	Home, STB POWER, TV POWER and Voice Key Caps	ABS	NCS S 9000-N	Gloss/SPI A-2
8	Side Key Cap	ABS	NCS S 9000-N	Matte/ YS 20043-B
			Follow BV LGI Nextgen (EOS) colours	

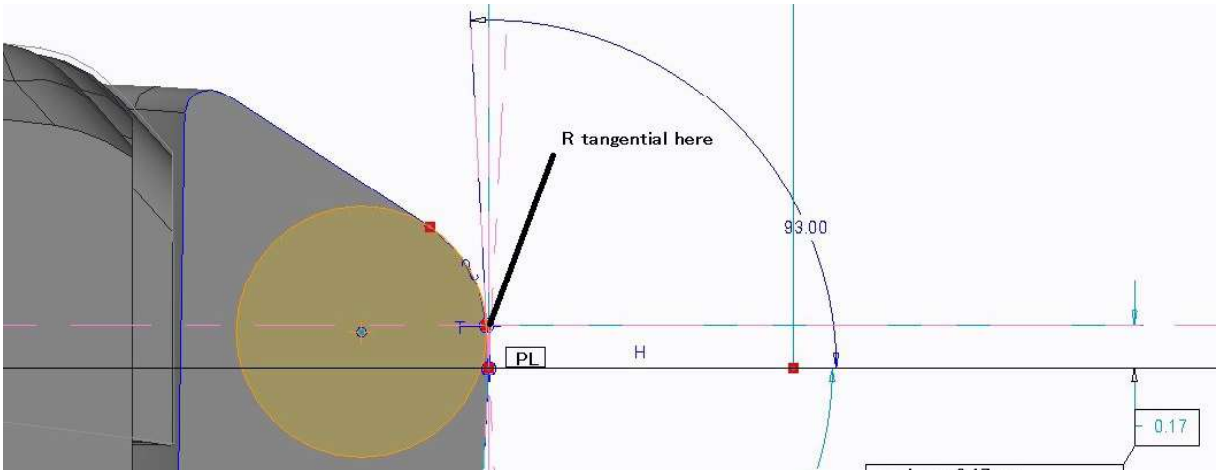
4.3 Dimensions and weight

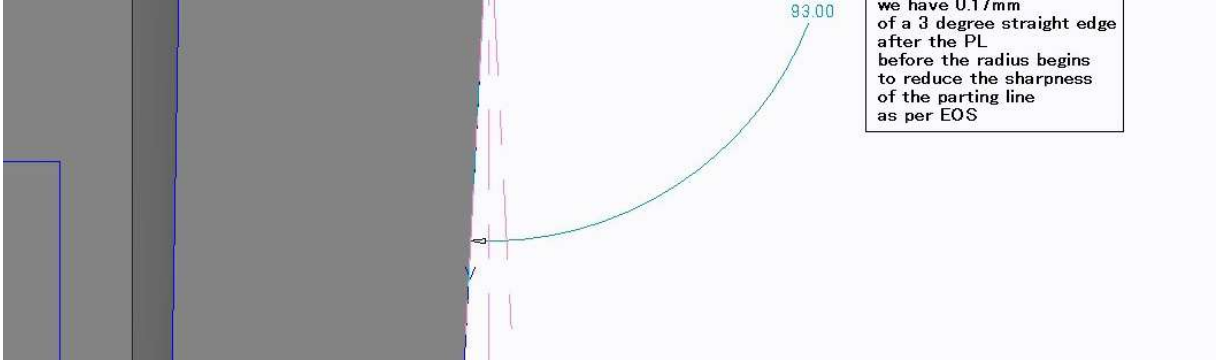
Length (mm)	**165.5**
Width (mm)	42.4
Thickness (mm)	22
Net Weight RCU	82g
Weight RCU incl. batteries	105g

Table 3: Dimensions & Weight Table

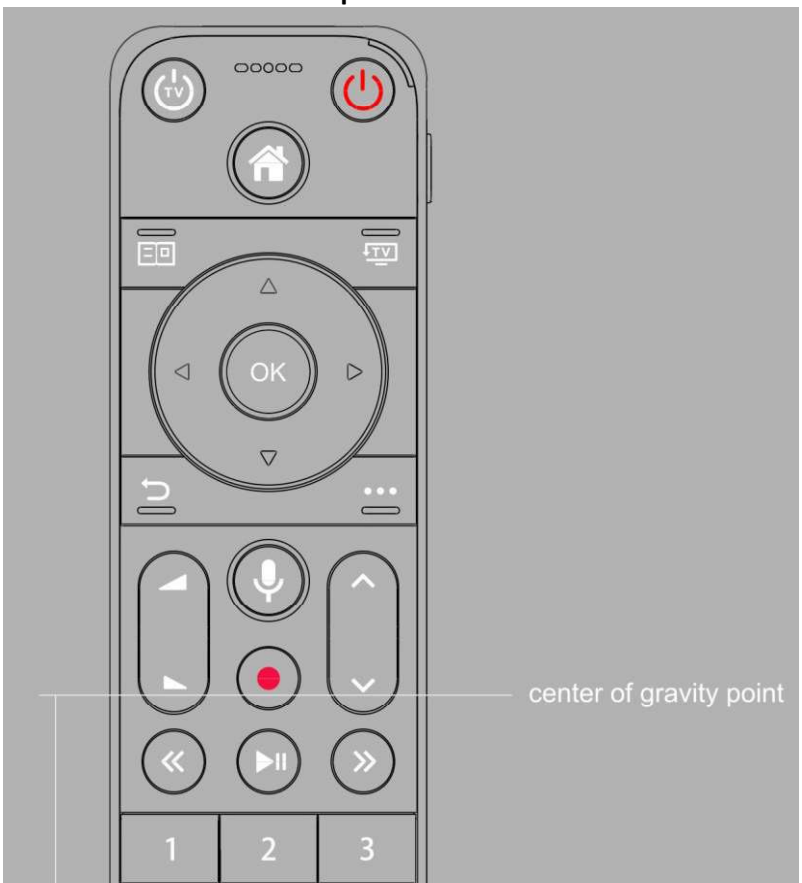
4.4 Drawings or other ME requirements

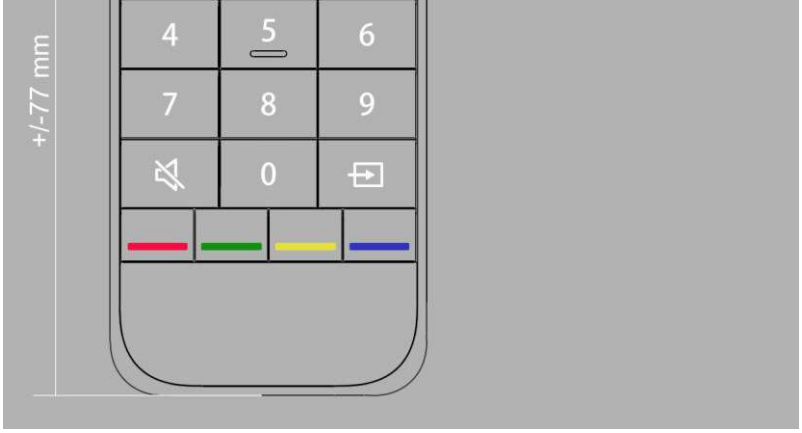
4.4.1 Casing requirements





4.4.2 Center of Gravity


TV
OK
center of gravity point
1
2
3



4.4.3 Battery lid

The battery lid should be removable more than 50 times without affecting the lid function.

4.4.4 Keypad Hardness Definition

Upper keypad: 65°- 70° Shore A; mic boot area is 70°
Side key rubber: 65°- 70° Shore A
Lower keypad rubber (co-molded with PC frame and ABS key caps, with- and without-teletext keys versions): 60±5° Shore A

5 Electrical Requirements

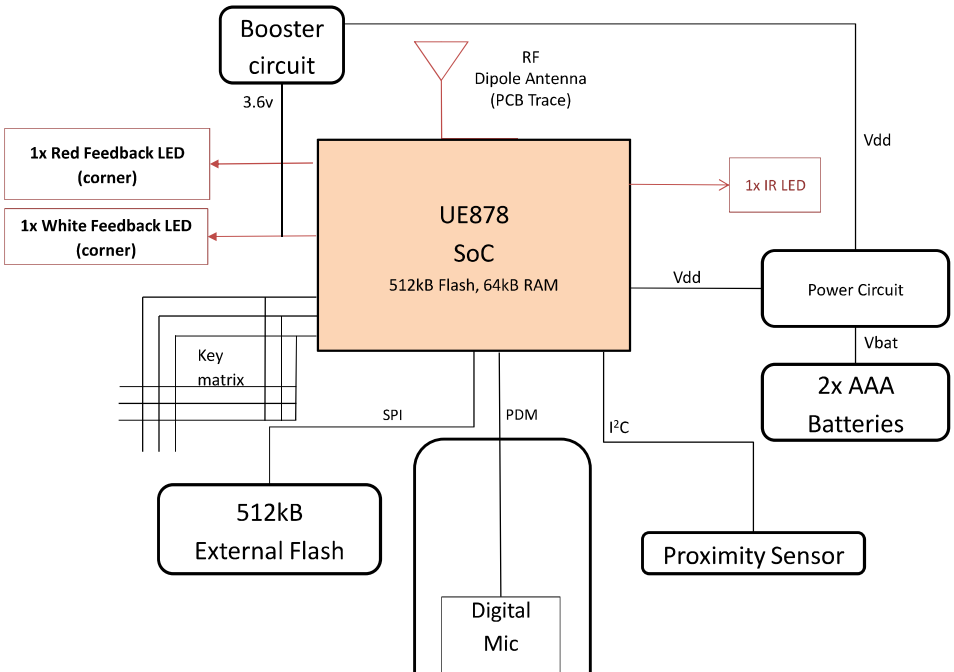
5.1 Hardware Information

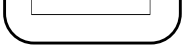
Hardware Information	
Information Description	Standard (Yes) , Custom, No
1) 3, 4.5, or 6 Volt Operation (AAAA, AAA, AA, Coin cell). a. Battery Type (Alkaline or Heavy Duty). b. Included in BOM. c. Batteries installed or in Polybag. d. Battery Pull Tab.	1) 3 Volt operation 2xAAA, 1050mAh. a. Alkaline. b. Yes. c. Installed. d. Yes.
2) Number of physical keys	2) 34+4
3) JTAG/Programming Interface on PCB. a. Type (e.g. pins or pads. If pads are chosen, then a fixture maybe required for programming the boards.)	3) Yes. a. Pads.
4) Crystal or Ceramic Resonator	4) Crystal
5) IR LED a. Quantity (1 or 2) b. Type (W, N, WW) “W=Wide N= Narrow”(3mm, 5mm)	5) Yes. 940nm a. 1. b. Wide, 3mm.
6) Low Battery Detection a. Type (External or Internal)	6) Yes. a. Internal.

7) PCB Details a. Type (Single Sided, Double Sided Silver thru hole, Double Sided Plated thru hole, Double Sided Carbon thru hole) b. Material (XPC, FR1, FR4) c. Fabrication process (Wet film or Dry film) d. Flammability Requirement (If mandated by customer)	7) a. Double Sided, Carbon Through Hole. b. FR1. c. Wet film. d. No.
8) RF Operation a. RF Type (433MHz; 868MHz; RF4CE; Z-Wave; Bluetooth; etc.) b. RF PCB (Integrated on Main board or Module) c. RF PCB Material (if separate PCB) d. Antenna Type (as recommended by EE team)	8) a. BLE and RF4CE. b. Integrated. c. Integrated. d. Dipole, PCB trace.
9) Visible LED for user feedback a. Type (SMT/Thru-hole) b. Color	9) Yes, 2x side-fire shining into the same light-guide. a. SMT b. White and Red.
10) Backlight	10) No.
11) Voice a. Microphone type (Analog, Digital) b. Mounting method (Top, bottom)	11) Yes. a. Digital b. Bottom.
12) Remote Finder	12) N/A
13) Sensors a. Type	13) Yes a. Proximity sensor

14) External Flash	14) Yes, 512kB.
15) Tactile Switch	15) Yes, for side key

5.2 Block Diagram





5.3 Performance Requirements

5.3.1 RF Range

50 meters minimum, open field. Test against CSR BLE USB dongle.

5.3.2 Transmit Power

Conducted transmit power with transmit power setting to maximum, measured at antenna feed-point:

Parameter	Condition	Min	Typ	Max	Unit
TX Power	BLE	-	6	-	dBm
	RF4CE	-	6	-	dBm

Table 4: Transmit Power Table

5.3.3 Receive Power

Conducted receive sensitivity measured at antenna matching input, shielded room.

Parameter	Condition	Min	Typ	Max	Unit
RX Sensitivity	BLE	-	-92	-	dBm
	RF4CE	-	-92	-	dBm

Table 5: Receive Power Table

5.3.4 IR Range and Angle Requirements

Transmission range depends upon the receive angle and the sensitivity of the receiver.

Requirements:

1. IR Angle and Range tests are based on an NEC carrier-based code.
2. Fresh batteries shall be used for verification.
3. Single Wide IR LED.

Head on Range: 15 meters (in the horizontal plane).
Remote Angle coverage: 22.5° Up/Down (either direction) and 22.5° Left/Right (either direction) from the Center/Head on target position) @ 6 meter.

5.3.5 IR Carrier Frequency

The RCU will support IR protocols with frequencies up to 455 kHz.

5.3.5.1 IR Carrier Frequency Accuracy

The IR Carrier Frequency Tolerance is IR code dependent. Unless otherwise specified, the following tolerances shall be observed:

1. IR code T0810: *f*Carrier = 40kHz, the Tolerance shall be ±3%
2. IR code C5557(NEC-based): *f*Carrier = 38kHz, the Tolerance shall be ±3%
3. IR code C1877: *f*Carrier = 56.9kHz, the Tolerance shall be ±3%
4. 56.9kHz< *f*Carrier ≤455khz, the Tolerance shall be ±5%

5.3.5.2 IR transparent bottom case

Power loss due to the IR transparent bottom case will be less than or equal to 1.0 dB.

5.3.6 Voice

5.3.6.1 Requirements for Nuance Voice Recognition Target

This section describes requirements for the PCM audio data measured after software equalization filters (high-pass, low-pass, and equalization).

Item	Requirement
Sample Rate	16 kHz
Sample Size	16 bits
Channels	1 channel
Frequency Response (after software EQ is applied)	+/- 4dB from 250 Hz – 7.3 kHz
Dynamic Range	≥48 dB
Sensitivity	95 dB SPL @ 1 kHz will produce at least 2 bits of headroom
DC Offset	DC offset shall be avoided
Background Noise	< 30 dBA SPL at the microphone
Linearity	Amplitude should linearly track input SPL change
THD	< 1% @ 94 dB SPL measured at 1kHz at the microphone, < 5% @ 90dB SPL measured at 1kHz entire system.

Table 6: Audio Requirement Table

5.3.7 Battery life

Battery life shall be 12 months minimum using 2x Alkaline AAA batteries and based on the following usage profiles.

5.3.7.1 BLE Usage Profile

STB in Active (On) State	4 hours per day
STB in Standby State	Case 1: 20 hours per day / Case 2: 0 hour
STB Powered Off	Case 1: 0 hours / Case 2: 20 hours per day
IR	30 key presses per day (0.5 s duration)
RF	240 key presses per day (0.5 s duration)
Pickup sensor activations	150 per day
Short LED blinks White (key presses)	270 per day
Voice	7 sessions of 6s each (42s) per day
Battery Life Status requests	12 events per day
Connection settings	See 8.7.7.18.5.9
Advertising settings	See 8.7.7.1
STB Power Up fading	6 sec (white LED fading in-out): 2 per day

Table 7: BLE Daily Usage Table

Power profile for BLE STB:

- STB is TV-powered. On: 4 hours, Standby: 0 hours, Off: 20 hours per day.
- STB is powered by external PSU. On: 4 hours, Standby: 20 hours, Off: 0 hours per day.

5.3.7.2 RF4CE Usage Profile

STB in Active (On) State	4 hours per day
STB in Standby State	Case 1: 20 hours per day / Case 2: 0 hour

STB Powered Off	Case 1: 0 hour / Case 2: 20 hours per day
IR	30 key presses per day (0.5 s duration)
RF	240 key presses per day (0.5 s duration)
Pickup sensor activations	150 per day
Short LED blinks White (key presses)	270 per day
Voice	7 sessions of 6s each (42s) per day
Battery Life Status requests	12 events per day
Auto Polling Period	40 sec
STB Power Up fading	6 sec (white LED fading in-out): 2 per day

Table 8: RF4CE Daily Usage Table

In RF4CE mode a GetPendingMessages command is sent to STB upon specific key releases as defined in section 8.8.7.1, at each auto polling period and after each proximity detection. See sections 8.8.7.2.1 and 8.8.7.2.2 for further details on STB polling.

5.4 RF Factory Test Mode

This remote shall support TSI.
TSI functionality will be used to test sample unit has been produced correctly.
It will include key numbering assignment, proximity sensor test and many other tests that will be described in detail in the EATP document.

The white LED will be considered as LED 1 whereas the red LED will be considered as LED2.

It is possible to activate TSI within 10s after POR through the dedicated enabling UART command (see TSI API document for further details).

You can refer to the UEI Test Specification for details.

5.4.1 TSI all key test

For all key test TSI will implement the following key scan event table:

Key Label	Scan code	Key Label	Scan code	Key Label	Scan code
TvPower	1	Blue	14	Epg	29
NavUp	2	Home	15	NavDown	30
Context	3	Select	16	VolDown	31
ChanDown	4	Voice	17	Digit1	32
Digit3	5	PlayPause	18	Digit7	33
Digit9	6	Digit5	19	Red	34
Yellow	7	Digit0	20	LiveTv	36
StbPower	8	Side key/Profile	22	Back	37
NavLeft	9	NavRight	23	Record	38
VolUp	10	ChanUp	24	Digit2	39
Rewind	11	FastFwd	25	Digit8	40
Digit4	12	Digit6	26	Green	41

Mute	13	Input	27		

Table 9: TSI Key Event Table

5.4.2 TSI Proximity Sensor Test

Proximity sensor should be tested with TSI through TSI Platform Specific Messages and should consists in TSI Platform Specific Commands and Events.
TSI Platform Specific Command and Events are formatted as follow:

Offset	Length	Name	Description
0	1	STX	= 0x02 (ASCII Start of Text)
1	2	Command + Length	= 0xA0, varies (Command 0x0A)
3	1	Platform subcommand	A one-byte subcommand specifying a specific platform specific functionality or test.
4	Varies1	Subcommand payload	Optional payload to support the platform subcommand.
3+length	1	ETX	= 0x03 (ASCII End of Text)

Table 10: TSI Platform Specific Commands and Events Format

5.4.2.1 TSI Proximity Sensor Test Commands, Responses and Events

TSI Proximity Test Commands should be sent by the Host in order to start or stop the proximity sensor test session.
Each command should be acknowledged by the RCU through corresponding events.
While the session is active, an event indicating a proximity detection should be sent by the RCU to the Host whenever a detection is triggered by the sensor.
TSI Proximity Test Commands and Events should be implemented as follow:

Test Commands, Responses and Events	Platform Subcommand	Subcommand payload	Description

	Value		
TSI Proximity Test Start Command	0xF1	N/A	Sent from Host to RCU. This command starts the Proximity Test session.
TSI Proximity Test Stop Command	0xF0	N/A	Sent from Host to RCU. This command stops the Proximity Test session.
TSI Proximity Test Start Response	0xF1	0x00 for success 0x01 for generic error code	Sent from RCU to Host. This event is an acknowledgement for the start command.
TSI Proximity Test Stop Response	0xF0	0x00 for success 0x01 for generic error code	Sent from RCU to Host. This event is an acknowledgement for the stop command.
TSI Proximity Test Detection Event	0xF2	N/A	Sent from RCU to Host. This event indicates a detection has been triggered by the sensor.

5.5 IR Factory Test Mode

Factory Test Mode shall be accessed only within 6 seconds of applying power to the remote control by pressing Key #22 & key #24 simultaneously (after applying power) which are specified in the product specification (depending on the key matrix).
Factory Test Mode shall not be accessed if Low Voltage is detected after apply power.
Once entered, the FDRA or E2 will be immediately reset with 4 blinks.
Then every key will produce a unique IR data output.
The IR will start sending with a key press and continue sending as long as the key is held down (no stuck key timeout).

The software will have NEC protocol or a custom protocol (for other UEI manufacturers/customers).
The IR code is determined when the software is compiled for each project. The remote control will remain in Factory Test Mode for 30 seconds, after which the remote returns to the normal default state.
Pressing the two keys after the six second period will have no effect.
Upon proximity events the RCU should issue a fading red blink as defined in section 8.3.2.4.
Factory Test Mode can be re-entered, if needed, by removing the battery and following the same procedure again.
Stuck key time out does not apply to FTM.

FTM table – NEC Protocol with Prefix based on last digit of SW version and X-release number

Key Number	Key Labels	Physical Key Switch Number	Input Pin	Output Pin	Data (Hex)
1	TV Power	S00	0	0	0x00
2	STB Power	S01	0	1	0x01
3	Home	S02	0	2	0x02
4	EPG	S04	0	4	0x04
5	Live TV	S05	0	5	0x05
6	Up Arrow (D-Pad)	S10	1	0	0x10
7	Left Arrow (D-Pad)	S11	1	1	0x11
8	Ok (D-Pad)	S12	1	2	0x12

9	Right Arrow (D-Pad)	S13	1	3	0x13
10	Down Arrow (D-Pad)	S14	1	4	0x14
11	Back	S15	1	5	0x15
12	Context Key	S20	2	0	0x20
13	Voice/Search	S22	2	2	0x22
14	Vol+	S21	2	1	0x21
15	Vol-	S24	2	4	0x24
16	CH+	S23	2	3	0x23
17	CH-	S30	3	0	0x30
18	Record	S25	2	5	0x25
19	Rewind	S31	3	1	0x31
20	Play/Pause	S32	3	2	0x32
21	Fast Forward	S33	3	3	0x33
22	1	S34	3	4	0x34
23	2	S35	3	5	0x35
24	3	S40	4	0	0x40
25	4	S41	4	1	0x41

38	Blue	S61	6	1	0x61
26	5	S42	4	2	0x42
27	6	S43	4	3	0x43
28	7	S44	4	4	0x44
29	8	S45	4	5	0x45
30	9	S50	5	0	0x50
31	0	S52	5	2	0x52
32	Mute	S51	5	1	0x51
33	TV Input	S53	5	3	0x53
34	Side Key (Profile)	S03	0	3	0x03
35	Red	S54	5	4	0x54
36	Green	S55	5	5	0x55
37	Yellow	S60	6	0	0x60

Table 11: FTM Key Table

6 QA Requirements

6.1 Quality Information

Quality/Testing Information	
Information Description	Standard (Yes) , Custom, No
1) ESD Protection	1) Yes
2) Environmental Testing	2) Yes
3) FCC or CE Certification Requirements	3) Yes, CE and FCC (including Chile)
4) Bluetooth Certification	4) Yes
5) Bluetooth Listing	5) No (customer responsibility).
6) Reliability Testing	6) Yes
7) Environmental Requirements a. RoHS Certification type (RoHS, Green, Sony Green, WEEE, REACH, Eup, Erp, Prop 65, etc.)	7) Yes a. WEEE, REACH, RoHS

Moreover, RCU is designed and tested to comply with the following standard revisions:

Test standard name	FCC & CE Standards and version numbers[1]
FCC Part 15 Subpart B (digital device;	

(digital device, unintentional radiator)	
FCC Part 15 Subpart C (intentional radiator)	15.205, 15.207, 15.209, 15.247
Safety	EN 62368-1:2014+A11:2017
Human exposure testing	EN 62479:2010
Radio testing (BLE/RF4CE)	EN 300 328 V2.2.2 (2019-07)
EMC testing	EN 55032:2015
EMC testing	EN 55035: 2017
EMC testing	EN 301 489-1 V1.9.2
EMC testing	EN 301 489-17 V3.1.1

[1] Version numbers are indicated for reference. Product should comply with latest official published revisions that are available at the project stage the product pass certifications.

6.2 Operating Temperature

For RCUs, please refer to Spec 0003 rev 26 for UEI standard operating temperatures.

7 Packaging and Marking Requirements

Packaging Requirements	
Information Description	Standard (Yes), Custom, No
1) Packaging a. Type (Bulk, Blister, Clamshell, Giftbox)	1) Yes. a. Bulk.
2) Master Carton a. Units per carton	2) Yes. a. 70 units.
3) Packaging bag a. Type (Poly bag, Bubble bag, etc.)	4) Yes a. Poly bag
4) User Manual	5) No

Marking Requirements and Miscellaneous Information	
Information Description	Standard (Yes), Custom, No
1) URC Part number and Date Code a. Type b. Location	1) Yes a. Laser-etched b. Battery compartment
2) "Made in China"	2) No
3) Customer/Serial Part number Label	3) No
4) Device Code label	4) No
5) Battery pull-out Tab	6) Yes
6) Other (Describe)	7) Manufacturer name and

	address to cover EN62368 cert

7.1 Date code format

Refer to CRS file for date code format.

7.2 Single Use Plastic Free Packaging Testing

The attached document below is describing the test methodology which applies to Single Use Plastic Free packaging type.

Packaging Test method for single us

8 Software Requirements

This software will be fully customized. The software development for this project will be captured under SW Project #: **7060** as for Apollo project. This will ensure the dual stack firmware can be uploaded on Apollo samples running on X9 version of the firmware.
One software release will serve all Liberty Global countries, and both color keys and non-color keys variations of this RCU. No country specific customizations or variances are envisioned in the software release.

Supported features:

- Voice over UAPI for both BLE and RF4CE
- Quickset with OTV
- OTA
- Modeless operation*
- User Profile selection (side key)
- Proximity Sensor
- Run Time Configuration parameters.
- TSI for automated testing.

*Modeless remote does not have separate STB and TV modes that users should switch manually in order to control each device (e.g. switch to TV mode to control TV volume and TV inputs, switch to STB mode to control STB channels, etc.). In modeless remote all functions are automatically mapped and switched conveniently and intuitively to the right device for greater user experience.

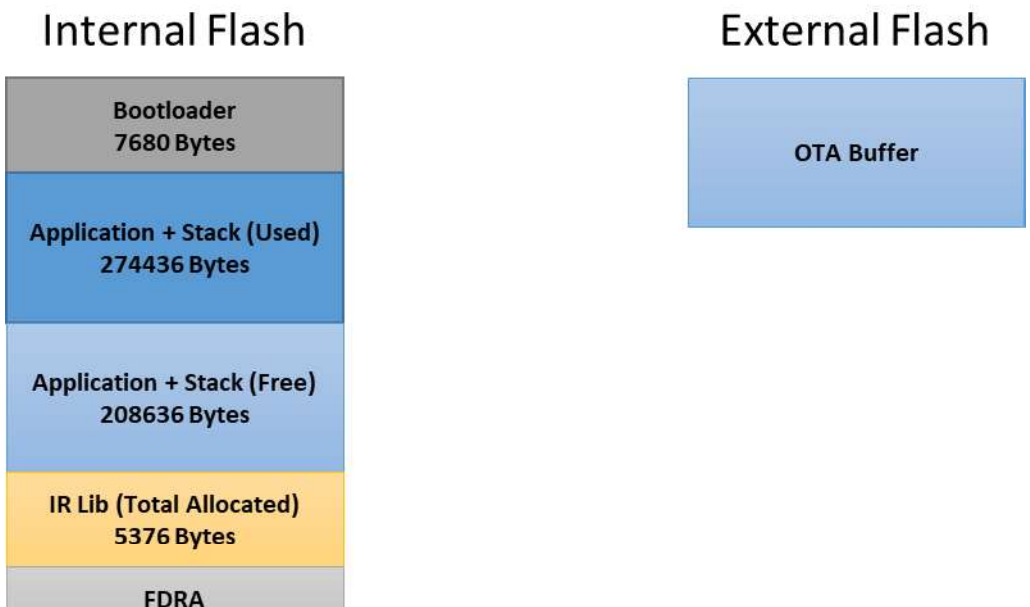
8.1 Specification / Firmware revisions

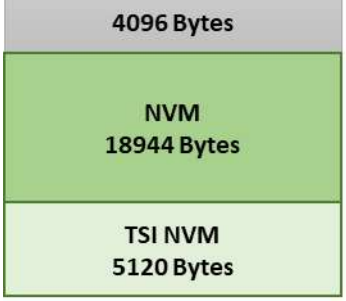
The table below is showing the relation between the revisions of a firmware and a product specification.
Note that the table is showing only relations where a specification change caused a firmware change.

Product Specification Version	Firmware Revision	Comments
V0.8	UE7060 – X11.02	First intermediate dual stack release is matching revision 0.8 + implement Quickset through App Id 0x13 for RF4CE + audio nibble order is reversed
v1.2	UE7060 – X12	First official dual stack release
v1.3	UE7060 – X13	
v1.8, v1.9	UE7060 – X18	
V2.0, v2.1, v2.2	UE7060 – X19	FW adaptation for voice compatibility with Selene STB
V2.3	UE7060 – X21	FW adaptation for Selene OTA specificities
V2.4	UE7060 – X24	Handling of multiple HW configurations
V2.5	UE7060 – X25	Do not exit RF4CE STB Power state machine when STB is in cold standby and does not ACK the RCU messages
V2.8	UE7060 – X26	NVM manager update + handling of 0xF7 in RAM to address volume control loss

8.2 Software Architecture

The following figure illustrates the RCU Software architecture. Note that the below memory mapping and used sizes depend on the FW version may be subject to change based on feature implementation and code occupancy.
The external flash is fully dedicated to OTA image buffering with OTA image containing application and stack for both BLE and RF4CE protocols. However, the maximum size that should be used in external flash for OTA buffering is equal to the maximum size allowed in internal memory for application and stack so OTA image can be fully copied from external to internal memory.





8.3 Definitions

8.3.1 Timings

Stuck Key Timeout	70s	The remote control can continuously send IR or RF signal for this duration of time in normal operating mode when a key is held down. After this duration, the RCU will stop transmitting IR and requires all keys to be released in order to process the next key press. This feature is intended to save the battery life and protect the IR diode.
Programming Timeout	10s	The remote will exit programming state if no user action is performed for this duration.
Voice Key Timeout	20s	Maximum duration of audio streaming.

Table 12: Timings Definition Table

8.3.2 Visual Feedback Timings

Below are the general LED feedback definitions.

8.3.2.1 LED Colors

White and Red LEDs are available.

8.3.2.2 Short blink white

Short blink white is defined as 100ms ON time and 100ms OFF time for the white LED.

8.3.2.3 Long blink or Error blink

A long or error blink is used to indicate an error or timeout condition.
Long Blink or Error blink is defined as 1s ON time and 200ms OFF time for LED.

8.3.2.4 Fading blink red

Red LED:
- 0.5 second linear fade up.
- 0.5 seconds ON (red).
- 0.5 second linear fade down.
- 0.5 seconds OFF.

The above cycle is executed **5 times** for a total of 10 seconds.

8.3.2.5 Fading blink white

White LED:
- 0.5 second linear fade up.
- 0.5 seconds ON.
- 0.5 second linear fade down.
- 0.5 seconds OFF.

The above cycle is executed **3 times** for a total of 6 seconds.

8.3.3 Visual Feedback Scenarios

This section describes all the scenarios where product should implement LED activity. Any scenario that is not described in this section should not implement LED activity.
Note that the product specification term “fading” is designated as “pulse” by LGI.

8.3.3.1 Boot-up

Four short white blinks first time booting from an image.
Two short white blinks when power is applied then after including after an OTA image activation.

External Flash IC availability is tested during RCU boot-up. If Flash access is failing, then the RCU will issue 1 error red blink as defined in section 8.3.2.3. The external flash IC access error blink should occur after the boot-up white blink above.

8.3.3.2 Pairing

During pairing fading white blink will be displayed until pairing success or failure.
Unsuccessful Pairing is indicated with 10 short red blinks for a total duration of 2 seconds.
Successful Pairing is not indicated by any LED activity.
If a key is pressed while RCU is in pairing state, then RCU will issue one short white blink (as defined in 8.3.3.6) and will resume with fading white blink.
In case of pairing failure, if a key is pressed before the end of the 10 short red blinks, then it is ignored.

8.3.3.3 Unpairing

There is no specific visual feedback while unpairing and on unpairing success. The visual feedback following the unpairing operation will depend on the operation that follow the unpairing.

For example, in manual pairing operation the success of unpairing will be indicated by the pairing operation following the unpairing.

Upon unpairing failure in normal operation mode, the visual feedback is addressed by the pairing failure visual feedback which is detailed section 8.3.3.2.

Upon unpairing failure in factory reset operation, the visual feedback is addressed by the factory reset failure visual feedback which is detailed section 8.9.2.

8.3.3.4 Reconnection

There is no visual feedback for reconnection scenario. Success or failure should be transparent and should not cause any LED activity.
However, if any type of key is pressed while RCU is in advertising state, then it is indicated by a short blink on white LED.

8.3.3.5 Probing

See sections 8.7.15.1 and 8.8.13.1 for further details.

8.3.3.6 Key Presses

All key presses are followed by one short white blink.
Upon long keypresses the LED should issue one short white blink when key is pushed.

8.3.3.7 Low Battery condition

The unit will not display any visual feedback (i.e. LED blinks) for Low Battery condition. All user warnings in this remote

will be handled via GUI of the STB, based on the messages that RCU will transmit to STB in this case.

8.3.3.8 Feature Reset

Please refer to section 8.9.1 for further details on feature reset.

8.3.3.9 Factory Reset

Please refer to section 8.9.2 for further details on factory reset.

8.3.3.10 Software version verification

LED indication will be made through red LED if RCU is paired to an RF4CE STB. Otherwise, it should be indicated though white LED.

For further details refer to section 8.9.3 for further detail on software version verification.

8.3.3.11 Short software version verification

LED indication will be made through red LED if RCU is paired to an RF4CE STB. Otherwise, it should be indicated though white LED.

For further details on short software version verification, refer to section 8.9.4.

8.3.3.12 Enable/Disable automatic polling

For further details on Enabling/Disabling automatic polling visual feedback refer to section 8.8.7.2.2.

8.3.3.13 Status/diagnostics reporting to STB

For further details on Status/Diagnostic reporting visual feedback refer to section 8.8.10.

8.3.4 Key Press Definitions

Key Sequence Legend:

Item	Symbol	Description
1	< >	Press and release key
2	<< >>	Press and hold key down for a minimum of 5 seconds
3	<<< >>>	Press and keep holding down until desired results are obtained
4	→	Continue on to next step

8.3.4.1 Simultaneous Double Key Press

Please refer to section 15 for simultaneous key presses.

8.3.4.2 Setup Key Definition

Key or Key Combo	Description
<<LIVE TV + 0>>	When pressed and held for 5 seconds, this key combination will set RCU in pairing mode and will trigger interleaved pairing attempts if RCU is unpaired. If RCU is paired, pairing attempts will be initiated on the same

	operation mode the RCU is associated to. For further information refer to sections 8.5.4, 8.5.5 and 8.5.8.3.
<<LIVE TV + REWIND>>	When pressed and held for 5 seconds, Clear Pairing + Manufacturing Reset
<<VOLUME DOWN + MUTE>>→<9>→<7>→<7>	Feature Reset
<<VOLUME DOWN + MUTE>>→<9>→<8>→<3>	SW version Verification (see section 8.9.3)
<<VOLUME DOWN + MUTE>>→<9>→<8>→<4>	Short SW version Verification (see section 8.9.4)

Table 13: Setup Key Definition

8.4 Hardware and Software Identification

8.4.1 Software Identification

In order to maintain backward compatibility and to allow OTA update on Apollo and Babylon samples that are already in the field and running on single stack firmware the Software revision will be unique and set to value 7060 for both single stack and dual stack projects.

This choice has been made in order to allow an OTA upload with dual stack firmware on samples that are in the field and that runs on single stack implementation.

8.4.2 Hardware Identification

A model ID will be stored in memory by factory and will be used by the firmware to identify if the RCU is a Babylon or Apollo device.
Based on the read value the firmware will apply the Apollo or Babylon audio settings.
The model ID will not be upgradable through OTA and will be stored at address: 0x407FE00.
The model ID will be set to:

- 0x01 for Apollo model (a.k.a. Babylon Short Form Factor)
- 0x02 for Babylon model (a.k.a. Babylon Long Form Factor) with Adesto external flash memory
- 0x03 for Babylon model (a.k.a. Babylon Long Form Factor) with GigaDevice external flash memory

By default, a value of 0 would force the device to be considered as an Apollo device. This is to ensure backward compatibility of the firmware with the previous revisions.

Moreover, due to global worldwide shortage situation on electronic components, Hardware identification will also be achieved through of UAPI HW Info parameter ID and more specifically through the 4th byte which is referring to HW version.
Below table is defining the 12 bytes of data contained in the UAPI HW Info parameter ID.

Byte	Bit	Name	Description
1		CPU Clock	CPU Clock in 4 MHz units
2		RF Support	
	7:5	Reserved	
	4	BLE	
	3	Wi-Fi	Wi-Fi supported if set
	2	Z-Wave	Z-Wave supported if set
	1	Bluetooth	Bluetooth supported if set

	0	RF4CE	RF4CE supported if set
3		IR Features	IR/XMP1/XMP2/XMP4
	7:4	Reserved	
	3	XMP-4	XMP-4 supported if set
	2	XMP-2	XMP-2 supported if set
	1	XMP-1	XMP-1 supported if set
	0	IR	IR supported if set
4		HW Version	Hardware version number
5:11		Reserved	

Before X23, HW version (byte 4) default value is set to 0x01 and resides in OTA reprogrammable Flash area.

From X23 release, HW version has changed and has been defined as a bit map with the below format :

Hw version	**Bit 0**	**Bit 1**	**Bit 2**	**Bit 3**	**Bit 4**	**Bit 5**	**Bit 6**	**Bit 7**
	External Flash	Booster / Charge Pump configuration		Proximity Sensor Package Type		Reserved	UE878 revision	

Hw version	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	0: GigaDevice 1: Adesto	0: Booster TI 1: reserved 2: Charge Pump 3: reserved		0: WLCSP 1: TSOT 2: DFN 3: reserved			0: Rev H 1: Rev J 2: reserved 3: reserved	

The reason for this change is to track the changes of the most critical electronic parts.

At production, HW version value will be set by factory according to the table above. This value will be located at the last page of UE878 internal flash memory at address **0x407FE02** which is not reprogrammable through an OTA update.
At RCU boot-up the firmware will read the last memory page and the value of the UAPI hardware configuration will set the HW version byte accordingly.

HW configurations are listed in annex 17.2 and are laser printed in the battery compartment.
RCUs produced before HW version format has changed, cannot report the correct HW version value after they have been upgraded through OTA to any FW version ≥ X23. These units will report an HW version 0x00. So, for units reporting HW version 0x00, the actual HW configuration can be identified by checking the model ID in combination of the HW version:

- HW version 0x00 with model ID 2 will correspond to HW configuration 1.
- HW version 0x00 with model ID 3 will correspond to HW configuration 2.

The firmware also need to be able to identify if the hardware is based on a booster or charge pump configuration. In order to simplify the code execution the firmware decision will be based on the byte content of the last memory page location at address **0x407FE01**.
In order to reflect the charge pump hardware configuration, the factory must set the **0x407FE01** to the value 0xAA. Any

other value will reflect a booster configuration.

8.5 IR Configuration

8.5.1 Default Devices & Settings

Device	Default Code
Cable	C5367 (RF4CE ID for LGI Nextgen STB)
TV	T2051 (Samsung)
AUD	No Default Code

Table 14: Default IR Device Codeset Configuration Table

8.5.2 Library Requirements

QuickSet Cloud IR DB.

1. Target Market:
   - Babylon: EU (including NL, BE, CH, UK, IE, PL, SK and AT) + LATAM.

2. Onboard IR database:
   a. Target Device Mode Types: T2051
   b. 10 keys IR Codes (are by default on the QS enabled keys see section 3.1.1)

8.5.3 IR Function Key Chart

See Section 3.1.1

8.5.4 RCU Pairing

This section is defining the general behavior of pairing procedure and is detailing the different pairing modes triggered by the different pairing key combinations that are defined in section 8.3.4.2.

Note: the pairing feature implementation has been modified from X27 FW version. This section is detailing the implementation of X27 and above revisions. For a description of the previous implementation please refer to the revision 2.8 of the product specification.

If the RCU is unpaired:

- RCU shall not go to pairing mode automatically after power up.
- The pairing mode should always be triggered by an RF key press.

Here are the different possible scenarios when RCU is not paired:

- If a single RF key is pressed, then pairing should be initiated in similar conditions as if <<LIVE TV + 0>> for pressed for 5s long.
- If a combination of key is pressed, then RCU should immediately stop the ongoing pairing procedure and after the key combination has been pressed for 5s RCU should start pairing as defined by Table 13: Setup Key Definition and document attached in section 8.5.10.
- If the key combination is not defined by Table 13 then key combination will be ignored.

If RCU is paired and a key combination is detected pressed for 5s then RCU should initiate manual pairing procedures as defined in sections 8.7.9 and 8.8.4.

The table that is included in the excel document at section 8.5.10, describes the actions that are performed by the RCU upon pairing key combination depending on the RCU state.

If the pairing procedure fails then RCU should indicate the failure through the error blink, go to idle and should remain paired to the previously paired device.

For further details on RCU behavior in pairing state please refer to section 8.7.5 and 8.8.3.
Pairing implements IR assisted feature which is described in section 8.5.8.
Visual feedback for pairing is detailed in section 8.3.3.2.

8.5.5 Dual mode pairing

Dual mode pairing is defining the scenarios where RCU will attempt interleaved BLE and RF4CE pairing attempts.
Dual mode pairing is only possible if the RCU is unpaired before the key combo is pressed.

So, if RCU is unpaired, dual mode with IR assisted pairing will be activated upon pressing <<LIVE TV + 0>>.

RCU will enter pairing state 5s after pairing key combination has been pressed and for a total duration of PAIR_TOTAL_DURATION.
Then RCU will interleave RF4CE and BLE pairing attempts in RF4CE bursts of PAIR_RF4CE_BURST_LENGTH followed by BLE bursts of PAIR_BLE_BURST_LENGTH. RF4CE bursts will precede BLE bursts.

Once pairing is initiated by the host device (STB), by responding to RCU advertising packets, the RCU should stop interleaving the pairing attempts and should complete the pairing procedure until it succeeds or fails.

The pairing implementation details for manual BLE or RF4CE pairing are described respectively in sections 8.7.9 and 8.8.3. The timing definitions are defined in section 8.5.9.

8.5.6 BLE only Pairing

If RCU is paired to a BLE Set Top Box, BLE mode with IR assisted pairing will be activated upon pressing <<LIVE TV + 0>> keys.
Existing pairing information will be erased only if the triggered pairing procedure is successful.
Otherwise, RCU should remain paired with the previously paired Set Top Box.

The details of BLE pairing are described in section 8.7.9.

8.5.7 RF4CE only Pairing

If RCU is paired to an RF4CE Set Top Box when the key combo is detected, RF4CE mode with IR assisted pairing will be activated upon pressing <<LIVE TV + 0>>.
Existing pairing information will be erased only if the triggered pairing procedure is successful.
Otherwise, RCU should remain paired with the previously paired Set Top Box.

The details of RF4CE pairing are described in section 8.8.3.

8.5.8 IR assisted Pairing

IR assisted pairing consists in transmission of specific IR codes that indicate the STB the RCU is trying to initiate a pairing procedure.
Upon reception of the IR assisted data the STB should enter pairing state.
The table below defines the different IR codes that are supported for IR assisted pairing and the sections below defines the implementation of IR assisted pairing.

Protocol Frequency	*UEI ID*	*UEI Exec*	*Hex Data*	*Host Device*	*Host type*
38kHz (Samsung 36-	C3477	549	0xa5b1	Selene	RF4CE

38kHz (Samsung 36-bit)	C3477	349	0xa3b1	Selene	RF4CE
36kHz (RCMM)	C1582	371	0xed	EoS	RF4CE
36kHz (RCMM)	C6247	371	0xec	EoS v2	BLE

Table 15: IR Assisted Pairing Protocols

Protocol	*Frequency*	*UEI ID*	*Prefix1*	*Prefix2*	*Prefix3*	*Data1*	*Data2*
Samsung 36-bit	38kHz	C3477	00001100B	01010100B	0101B	10100101B	10110001B

Table 16: Samsung 36 bits Protocol details

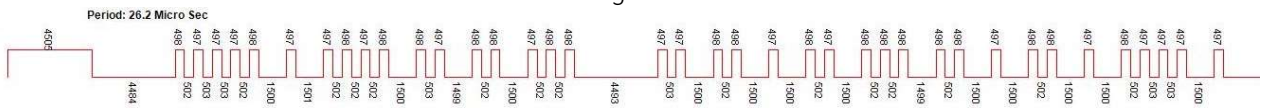


Figure 1: Visualization of Samsung 36-bit IR signal which is blasted during the pairing

Protocol	Frequency	UEI ID	Prefix1	Prefix2	Prefix3	Data
RCMM	36kHz	C1582	00100000B	11000000B	10100110B	11101101B
RCMM	36kHz	C6247	00100000B	11000000B	10100110B	11101100B

Table 17: RCMM Protocol Details

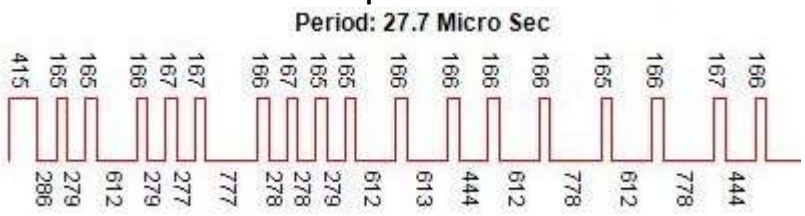


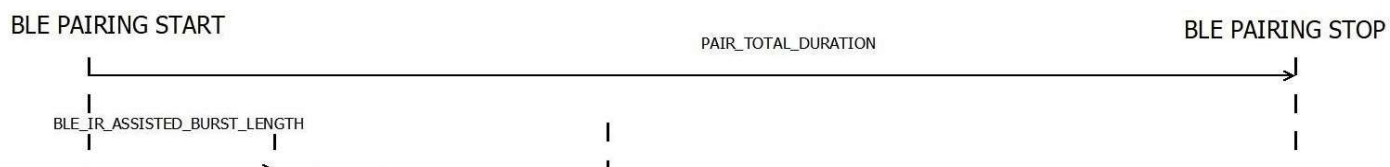
Figure 2: Visualization of EOS RCMM IR signal which is blasted during the pairing

8.5.8.1 BLE only IR assisted Pairing

With IR assisted feature, BLE pairing burst will be preceded by IR codes corresponding to BLE Host Type as defined in Table 15.

The first IR burst transmission and BLE advertising will start in parallel.
IR assisted burst should be transmitted for a duration of BLE_IR_ASSISTED_BURST_LENGTH (5 frames) and should be repeated at each BLE_IR_ASSISTED_BURST_INTERVAL.
BLE advertising and IR assisted burst repetitions will not stop before a STB initiates a BLE connection or until PAIR_TOTAL_DURATION.

The diagram below is illustrating a complete sequence of BLE pairing with IR assisted feature:



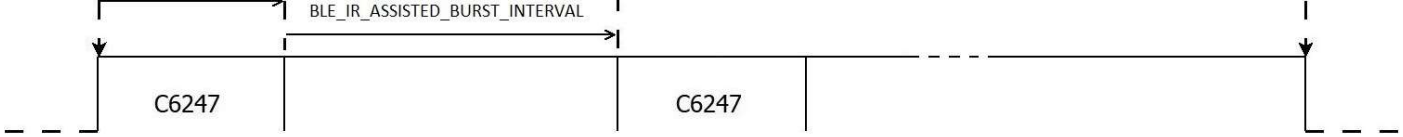


Figure 3: Sequence of BLE Only Pairing with IR Assist Enabled

When STB initiates the pairing connection through connection request indication then IR assisted signal transmission should be stopped.

8.5.8.2 RF4CE only IR assisted Pairing

With IR assisted feature, RF4CE pairing burst will be preceded by IR codes corresponding to RF4CE Host Type as defined in Table 15.

Each IR signal should be repeated for RF4CE_IR_ASSISTED_BURST_LENGTH as soon as RCU enters the pairing mode. First RCMM IR signal is emitted RF4CE_IR_ASSISTED_BURST_LENGTH, followed up by RF4CE_IR_ASSISTED_BURST_LENGTH of the Samsung 36-bit IR signal.

A delay of RF4CE_IR_ASSISTED_BURST_GAP to separate the 2 protocols shall be maintained in accordance with existing Horizon implementation. The IR pairing burst shall be transmitted as soon as pairing is initiated.

The same IR assisted sequence will be repeated every RF4CE_IR_ASSISTED_BURST_INTERVAL.

In parallel of the first IR burst transmission RF4CE discovery should start for a duration of PAIR_TOTAL_DURATION.

The diagram below is illustrating a complete sequence of RF4CE pairing with IR assisted feature:

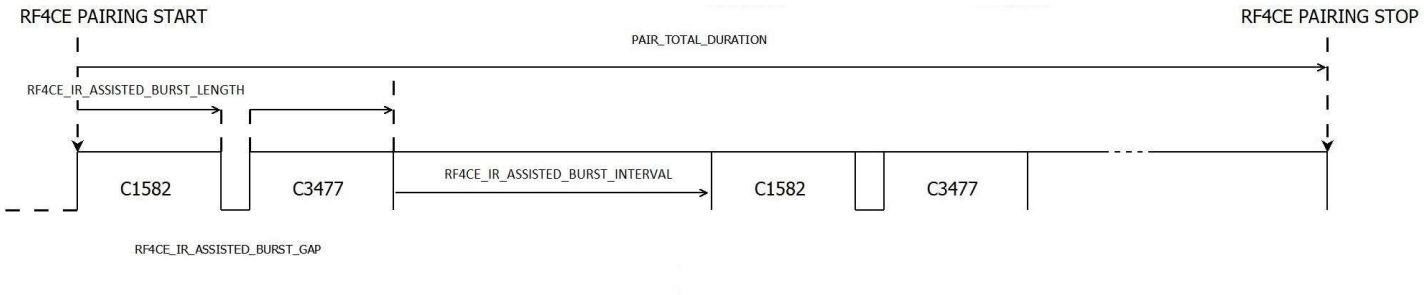


Figure 4: Sequence of RF4CE Only Pairing with IR Assist Enabled

When STB initiates the pairing connection through discovery response then IR assisted signals transmission should be stopped.

8.5.8.3 Dual mode IR assisted Pairing

Dual mode will consist in successive executions of RF4CE only and BLE only sequences as defined in sections 8.5.8.1 and 8.5.8.2. Each sequence will last PAIR_RF4CE_BURST_LENGTH and PAIR_BLE_BURST_ LENGTH and for a total duration of PAIR_TOTAL_DURATION.

The diagram below is illustrating a single sequence of dual mode pairing with IR assisted feature:

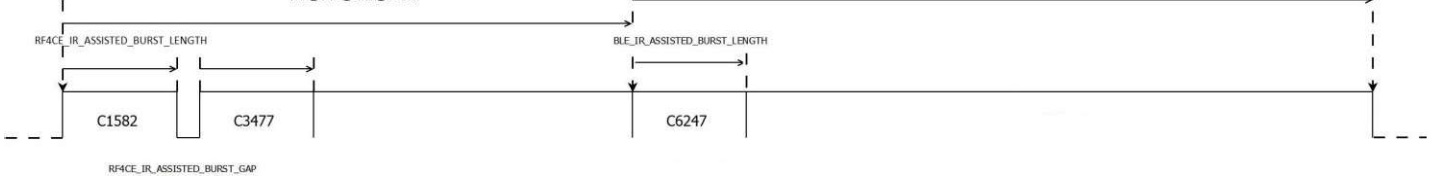


Figure 5: Sequence of Dual Mode Pairing with IR Assist Enabled

The sequence above will be repeated up to PAIR_TOTAL_DURATION.

8.5.9 Pairing Timing definition

Parameter	Value	Description
PAIR_TOTAL_DURATION	60s	Duration of RCU being in pairing state
PAIR_BLE_BURST_ LENGTH	5s	Duration of a BLE pairing burst in dual mode pairing
PAIR_RF4CE_BURST_LENGTH	5s	Duration of an RF4CE pairing burst in dual mode pairing
BLE_IR_ASSISTED_BURST_LENGTH	5 frames	Duration for BLE IR assist
RF4CE_IR_ASSISTED_BURST_LENGTH	5 frames	Duration for RF4CE IR assist
BLE_IR_ASSISTED_BURST_INTERVAL	5s	Repetition interval for BLE IR assist burst
RF4CE_IR_ASSISTED_BURST_INTERVAL	5s	Repetition interval for RF4CE IR assist burst
RF4CE_IR_ASSISTED_BURST_GAP	300ms	Gap separating the different IR assisted bursts from each other

Table 18: Pairing Timing Definition

8.5.10 Pairing State Operations

The table in the following attached document defines the RCU behavior when a user presses a combo key that is defined in Table 13: Setup Key Definition according to the RCU pairing state.

STB state initial conditions	Unpaired	Paired BLE	Paired RF4CE	RCU is unpaired AND in pairing state (pairing is ongoing)

Auto Pair upon single RF keypress	Start Interleaved Pairing with IR assisted enabled	N/A	N/A	Do nothing if no key combo detected
LIVE TV +0	Start Auto Pair upon the first keypress and ignore second keypress	5s after the combo key is detected then: Step 1 - Start BLE Pairing	5s after the combo key is detected then: Step 1 - Start RF4CE Pairing	Do nothing
LIVE TV + RWD	Step 1 - Start Auto Pair upon the first keypress Step 2 - As soon the 2 keys are detected as pressed simultaneously then stop interleaved pairing - no error blink Step 3 - 5s after the combo key is detected then perform Factory reset	5s after the combo key is detected then perform Factory reset	5s after the combo key is detected then perform Factory reset	Step 1 - Stop ongoing pairing activity with no error blink as soon the 2 keys are detected as pressed simultaneously Step 2 - 5s after the combo key is detected then perform Factory reset
VOLUME DOWN + MUTE	5s after the combo key is detected then enter programming mode	5s after the combo key is detected then enter programming mode	5s after the combo key is detected then enter programming mode	Step 1 - Stop ongoing pairing activity with no error blink as soon the 2 keys are detected as pressed simultaneously Step 2 - 5s after the combo key is detected then enter programming mode

NOTES:
#1 - RCU should not enter pairing state automatically after POR
#2 - In case of <<LIVE TV +RWD>> the result of factory reset is indicated as detailed in Factory Reset section
#3 - Pairing LED indication should start by fading up. Visual feedback should be visible 5s after the key combo is pressed.

8.6 Battery Level Operations

The table below defines the levels and thresholds of the battery according to the voltage level that is measured by the micro-controller of the RCU with an accuracy of +/-60mV. The RCU will enable and disable features based on the measured voltage level.

However, the way the RCU notifies the battery level to the Host is protocol dependent and is defined in section 8.7.11 for BLE and in section 8.8.9 for RF4CE.

Voltage State	VDD Value	Battery Condition	IR Xmit	RF Xmit	RCU Behavior
$V_{RESTART}$	2.7V	VDD > VRESTART	Yes	Yes	Lower voltage can be switched back to Good or OK state upon detecting VRESTART.

V_{OK}	>2.3V	VDD > $V_{WARNING}$			All operations are allowed
$V_{WARNING}$	2.3V	$V_{WARNING}$ ≥ VDD > $V_{CRITICAL}$	Yes	Yes	All operations are allowed including reconnection, except pairing and OTA.
$V_{CRITICAL}$	2.2V	$V_{CRITICAL}$ ≥ VDD > V_{STOP}	See RCU behavior	See RCU behavior	In BLE mode: BLE and IR transmission are allowed. Voice is allowed. In RF4CE mode: RF4CE and IR transmission are not allowed. Voice is not allowed. In both modes: other operations (OTA, pairing, Quickset configuration) are not allowed.
V_{STOP}	2.0V	VDD ≤ V_{STOP}	No	No	RCU operation should be stopped

Table 19: Battery thresholds

The flag "*Vrestart detected*" is intended to notify the Host the RCU batteries have been removed and reinserted or replaced.
In order to achieve this purpose:

- "**V***restart* detected" should be set when a voltage below V_{OK} is detected followed by the detection of a battery level higher than $V_{RESTART}$. In this situation, RCU will issue a Software Reset and at next reconnection STB should read "**V***restart* detected" flag.
- "**V***restart* detected" should also be set after a boot-up triggered by a POR of the RCU regardless the VDD level. This is to ensure the flag is set in every situation where RCU can detect the batteries have been removed.

The flag "*Vrestart detected*" will be read by STB through the UAPI message "Read RCU internal flags". See section 8.10.2.1 for further details.

Remote checks the battery level at +/-60mV accuracy in the following conditions:

- Upon key press events
- Upon key release
- During inter frame delay of IR/RF transmission
- At Boot-up
- Periodically during Voice and OTA activity
- Upon proximity sensor events if proximity sensor is enabled. In case proximity sensor is disabled battery level is not read.
- For RF4CE protocol the battery level will be also assessed by the firmware upon UAPI polling.

8.7 BLE Operating Mode and Configuration

8.7.1 BLE Local Device Configuration

This section defines the configuration of the Bluetooth remote control.

8.7.1.1 BD or MAC Address

Each Bluetooth device is identified by a unique MAC address called BD address (Bluetooth Device address) which is taken from ranges assigned by IEEE. Unless otherwise noted, the BD address shall be as assigned by the silicon vendor.

8.7.1.2 PID and VID

The PID and VID values are defined in the table below.

Vendor ID (VID)	**Product ID (PID)**
0x06E7	0x8140

Table 20: PID/VID Table

8.7.1.3 BLE PHY Configuration

BLE chipset should be configured to support 2M and 1M PHY options. BLE PHY bitrate will be selected in function of capabilities that are supported by STB.
If STB supports 2M PHY then the RCU should select this option and RCU should fall back to 1M PHY if STB does not support the feature.

8.7.2 BLE GATT Configuration

The relevant RCU GATT characteristics are defined in the table below.

Name	**Assigned Number**	**Length**	**Value**	**Description**
System ID	0x2A23	8 bytes	<OUI> 0xFFFE<MI>	*Where <OUI> is equal to the 3 MSBs of the RCU's IEEE address and <MI> is equal to the 3 LSBs of the IEEE address*
Model Number String	0x2A24	10 bytes	"RemoteUnit"	*Model name of the product in ASCII. Normally 16 – 20 Byte String*

Firmware Revision String	0x2A26	6 bytes	"BL 195"	*This is the boot loader version.*
Hardware Revision String	0x2A27	2 bytes	"01"	*Hardware version (P = PCB version).*
Software Revision String	0x2A28	10 bytes	"ABCD.XX.YY_M.N"	*ABCD = 4 digits software project number (7060 for Apollo, Babylon single and dual stack projects)* *XX = X-release revision (e.g. 01 for X1)* *YY = release minor revision. 'M' is RTC major version and 'N' is RTC minor version.*
Manufacturer Name String	0x2A29	27 Bytes	"Universal Electronics, Inc."	*RCU Manufacturer Name: "Universal Electronics, Inc."*
PnP ID	0x2A50	7 bytes	0x02 <0x06E7, PID, Product Version>	*0x02; 0x06E7; 0x8140; 0x0110* • *Vendor ID Source: 0x02 (corresponds to USB Implementer's forum)* • *VID=Vendor ID: 0x06E7 (corresponds to UEI's USB vendor ID 0x06E7)* • *PID=Product ID: 0x8140 (Product ID can be vendor-specific)* • *Product Version (JJ.M.N): 0x0110 (corresponds to JJ= 01, M=1, N=0. i.e., ver. 01.1.0)*

Table 21: GATT Configuration Table

8.7.2.1 GAP Characteristics:

RCU shall support the following characteristics in the GAP Service.

Characteristic	**UUID**	**Value**	**Description**
Device Name	0x2A00	"RemoteUnit"	RCU Device Name
Appearance	0x2A01	0x0180	Generic Remote Control
Peripheral Preferred Connection parameters	0x2A04	0x0009 (9 => 11.25ms), 0x0009 (9 => 11.25ms), 0x01F3 (499), 0x06A4 (1700 => 17sec)	Min Connection Interval, Max Connection Interval, Slave Latency, Connection Supervision Timeout Multiplier

Table 22: GAP Characteristics Table

8.7.3 BLE Profile

8.7.3.1 HID over GATT Profile (HOGP)

RCU shall support the following characteristics in the HOGP service.

Characteristic	UUID	Value	Description
HID Information	0x2A4A	0x0111, 0x00, 0x01	USB HID version, Country Code, Flags
Report Map	0x2A4B	0x05 0x01 0x09 0x06 0xA1 etc.	HID format information
Protocol Mode	0x2A4E	0x01	Report Protocol Mode

Table 23: HoGP Characteristics Table

8.7.3.1.1 HID Descriptors

HID descriptor is defined as follows:

```
0x05, 0x01,          // Usage Page (Generic Desktop)
0x09, 0x07,          // Usage (Keyboard)
0xA1, 0x01,          // Collection (Application)
0x85, 0x01,          // Report ID (0x01)
0x95, 0x02,          //      Report Count (2)
0x75, 0x10,          //      Report Size (16)
0x05, 0x07,          //      Usage Page (Key Codes)
0x19, 0x00,          //      Usage Minimum (0)
0x2A, 0x00, 0x01,    //      Usage Maximum (0x100)
0x15, 0x00,          //      Logical Minimum (0)
0x26, 0x00, 0x01,    //      Logical Maximum (0x100)
0x81, 0x00,          //       Input (Data, Array)
0xC0,                // End Collection

0x05, 0x0C,          // Usage Page (Consumer Devices)
0x09, 0x01,          // Usage (Consumer Control)
0xA1, 0x01,          // Collection (Application)
0x85, 0x03,          // Report ID (0x03)
0x95, 0x02,          //      Report Count (2)
0x75, 0x10,          //      Report Size (16)
0x19, 0x00,          //      Usage Minimum (0)
0x2A, 0xFF, 0x03,    //      Usage Maximum (0x03FF)
0x15, 0x00,          //      Logical Minimum (0)
0x26, 0xFF, 0x03,    //      Logical Maximum (0x03FF)
0x81, 0x00,          //      Input (Data, Array)
0xC0,                // End collection

0x05, 0x01,          //      Usage Page (Generic Desktop)
```

```
0x09, 0x80,                 //          Usage (System control)
0xA1, 0x01,                 //          Collection (Application)
0x85, 0x02,                 //          Report ID (0x02)
0x95, 0x02,                 //          Report Count (2)
0x75, 0x10,                 //          Report Size (16)
0x19, 0x00,                 //          Usage Minimum (0)
0x2A, 0x00, 0x01,           //          Usage Maximum (0x100)
0x15, 0x00,                 //          Logical Minimum (0)
0x26, 0x00, 0x01,           //          Logical Maximum (0x100)
0x81, 0x00,                 //          Input (Data, Array)
0xC0,                       //          End Collection


// UAPI Input Report, UAPI_INPUT_REPORT_ID
0x06, 0x01, 0xFF,           // USAGE_PAGE (Vendor Specific)
0x09, 0x01,                 // USAGE (Vendor Usage 1)
0xA1, 0x01,                 // COLLECTION (Application)
0xA1, 0x02,                 //   COLLECTION (Logical)
0x85, 0x07,                 //     REPORT_ID (7)
0x09, 0x14,                 //     USAGE (Byte)
0x15, 0x80,                 //     LOGICAL_MINIMUM (-128)
0x25, 0x7F,                 //     LOGICAL_MAXIMUM (127)
0x75, 0x08,                 //     REPORT_SIZE (8)
0x95, 0x14,                 //     REPORT_COUNT (20)
0x81, 0x22,                 //     INPUT (Data, Var, Abs, NPrf)
0xC0,                       //   END_COLLECTION (Logical)
0xC0,                       // END_COLLECTION

// UAPI Output Report, UAPI_OUTPUT_REPORT_ID
0x06, 0x01, 0xFF,           // USAGE_PAGE (Vendor Specific)
0x09, 0x01,                 // USAGE (Vendor Usage 1)
```

```
0xA1, 0x01,          // COLLECTION (Application)
0xA1, 0x02,          //   COLLECTION (Logical)
0x85, 0x08,          //     REPORT_ID (8)
0x09, 0x14,          //     USAGE (Byte)
0x15, 0x80,          //     LOGICAL_MINIMUM (-128)
0x25, 0x7F,          //     LOGICAL_MAXIMUM (127)
0x75, 0x08,          //     REPORT_SIZE (8)
0x95, 0x14,          //     REPORT_COUNT (20)
0x91, 0x22,          //     OUTPUT (Data, Var, Abs, NPrf)
0xC0,                //   END_COLLECTION (Logical)
0xC0,                // END_COLLECTION

// Voice Report, VOICE_INPUT_REPORT_ID
0x06, 0x01, 0xFF,    // USAGE_PAGE (Vendor Specific)
0x09, 0x04,          // USAGE (Vendor Usage 4)
0xA1, 0x01,          // COLLECTION (Application)
0xA1, 0x02,          //   COLLECTION (Logical)
0x85, 0x1F,          //     REPORT_ID (31)
0x09, 0x14,          //     USAGE (Byte)
0x15, 0x80,          //     LOGICAL_MINIMUM (-128)
0x25, 0x7F,          //     LOGICAL_MAXIMUM (127)
0x75, 0x08,          //     REPORT_SIZE (8)
0x95, 0x14,          //     REPORT_COUNT (20)
0x81, 0x22,          //     INPUT (Data, Var, Abs, NPrf)
0xC0,                //   END_COLLECTION (Logical)
0xC0                 // END_COLLECTION
```

8.7.4 Advertising State

In this state RCU and STB are not connected and RCU initiates a connection attempt. In order to reduce connection

latency STB should maximize its BLE Scan activity. If possible, it should also turn off the Bluetooth Classic scan activity.

8.7.4.1 Indirect Advertising

8.7.4.1.1 Advertising Package Data (UE878 BLE)

While trying to pair, the RCU will send advertising packets (ADV_IND) with the following data format:

Table 24: Indire

Data Field			Length (bytes)	Value	Description
Flag	Length		1	0x02	
	Type		1	0x01	
	Value		1	0x05	Limited Discoverable, No BR/EDR
Appearance	Length		1	0x03	
	Type		1	0x19	
	Value		2	0x0180	Generic Remote Control
Service	Length	Option S1	1	0x05	HID and Battery
	Type		1	0x02	
	Value	Option S1	4	0x1812, 0x180F	HID and Battery Service
Shortened Local Name	Length		1	0x0B	11 bytes
	Type		1	0x08	
	Value		10*	RemoteUnit	RCU short name in ASCII
Manufacture Spec	Length		1	0x05	
	Type		1	0xFF	
	Value	Option M1	4	0x0093, 0x8000	UEI Specific Flag, UAPI support via HOGP

*Maximum value for N is dependent on the available bytes remaining from the service value option byte length fields. N = 31 - Service Value (length) – 21 = 10.

8.7.4.1.2 Advertising Package Data for Wake on BLE

The usual way that the STB SoC is woken is by another listening IC – such as BLE chipset - that interrupts the host and brings it out of sleep mode.

When RCU is paired and STB Power key is pressed the extra data "WAKEUP" will be inserted to the manufacturer specific data in order to allow waking up the STB SOC from BLE activity.

So, when STB Power key is pressed, while RCU is paired, the advertising data will be formatted as follows and STB power key report will not be sent:

DATA field	Field Length	Length	Value
Flags	3	0x02	GAP_ADTYPE_FLAGS (0x01), GAP_ADTYPE_FLAGS_BREDR_NOT_SUPPORTED (0x04)
Appearance	4	0x03	GAP_ADTYPE_APPEARANCE (0x19), GAP_APPEAR_GENERIC_RC (0x0180)
Service UUIDs	6	0x05	GAP_ADTYPE_16BIT_MORE (0x02), HID_SERVICE_UUID (0x1812)

			Battery service (0x180F)
Manufacturer Specific Data	10	0x09	GAP_MANUFACTURER_SPEC_DATA (0xFF), BROADCOM_ID_CODE (0x000F), ***WAKEUP (0x57 0x41 0x4B 0x45 0x55 0x50)***

Table 25: Indirect Advertising Data for STB WAKE UP

Notes:

- STB Power key is treated in the context of OTV feature and the full details of how it is handled by RCU are exposed in section 8.10.1.2.
- Limited discoverable flag is not set in the data of advertising packet for reconnection scenarios through indirect advertising. This change request has been applied in order to avoid undesired bonding requests from devices that would detect the RCU (see Bluetooth Core spec 4.2 Vol 3 Part C section 6.5.4 for further details).

8.7.4.1.3 Scan Response PDU Data

When the STB will be in active scan mode it will issue a scan request (SCAN_REQ PDU) upon reception of an ADV_IND from the RCU. The RCU shall respond with a scan response (SCAN_RSP PDU) containing the following Scan Response Packet Format:

Data Field		**Length (bytes)**	**Value**	**Description**
Complete Local Name	Length	1	0x0B	
	Type	1	0x09	
	Value	10	RemoteUnit	RCU complete name in ASCII

Table 26: Scan Response PDU data

8.7.4.2 Direct Advertising

RCU will go to direct advertising in case of Auto Reconnection / Keep Awake scenarios for further details please refer to section 8.7.7.2.
RCU should enter direct advertising state if a RF key (except for STB Power key) is pressed while the RCU is paired and disconnected.

8.7.5 Pairing State

In order to keep the authentication as transparent as possible for the user the pairing procedure will be forced in “Just Works” configuration. To achieve the “Just Works” configuration the STB and the RCU will set its IO capabilities as “No Input/No Output”.

RCU should be considered as in out of box scenario as long as the RCU has no pairing information stored in memory and is in deep sleep mode.
After the user applies power to the RCU (i.e. removes the battery tab) the remotes shall not automatically go into pairing mode. If the remote is in deep sleep and not paired, the user should press any RF key causing the RCU to get out from deep sleep mode and enter the pairing mode.

While the RCU is in deep sleep mode and not paired the first key report should be sent after the connection has been established except for STB Power Key.
Moreover, any additional keys that are pressed, except the key initiating the pairing, before the pairing is completed should be ignored.

While in pairing mode, the RCU transmits general BLE advertisement (ADV_IND) packets with an advertisement interval of PAIR_ADV_INTERVAL and repeatedly transmits for up to PAIR_ADV_DURATION. If no connection is established as a result of the advertising, the RCU returns to deep sleep state after PAIR_ADV_DURATION.

Note: in case pairing fails RCU will show an error blink.

8.7.6 Connected State

8.7.6.1 Idle Mode

When connected, all communication between the project remote and the target will be via Bluetooth Low Energy (BLE). Commands shall be sent as HID reports over GATT using standard HID.

BLE connection parameters have been chosen for minimizing the power consumption, for not overloading the bandwidth and for minimizing the latency.

8.7.6.2 Connection Parameters

The BLE connection parameters as being used by the Babylon RCU are as defined in section 8.7.7.1.

8.7.6.3 Key Transmission

The keys shall be sent as HID reports as described in the BLE key chart.
For the virtual keys such as Power Probe or STB wake up, the key release should be sent 100ms after the key has been pressed.

8.7.6.4 Key Report Transmission

The key reports are sent using HOGP (HID over GATT profile) profile. Key press report shall be sent at the time of key press. No Key repeat reports will be sent if a key is held down for a long time. When the key is released, a key release NULL report will be sent. If the key is held down up to stuck key timeout, the remote will send NULL report and terminate BLE transmission.

For multiple key press scenario details please refer to section 15 which illustrates the different multiple key press scenarios.

The key reports are defined as per HID standards and the key codes are implemented as defined by the HID descriptor.

8.7.6.5 BLE Key Chart

See section 3.1.1

8.7.7 Connection Behaviors

Once a remote and STB are paired, the BLE connection will be kept maintained using the connection parameters in section 8.7.7.1. In order to optimize the RCUs battery life and to support the full feature set, the STB should always accept the connection parameters suggested by the RCU.

Any disconnect scenario is detailed in section 8.7.7.2.

For cases where a reconnection is triggered from a key press then first key report should be sent except for STB power key.
Any additional key press before reconnection is established should be ignored.

8.7.7.1 Connection Timing Definitions

The values in the table below will be used by the Apollo RCU during setting up the connection with the Apollo STB.

Parameter	UE878 BLE	Description
CONNECTION_INTERVAL	11.25ms	Frequency that central (Host) will ask for data from peripheral

		device (RCU).
SLAVE_LATENCY	499	Number of times that peripheral device (RCU) can choose not to answer central (Host)
SUPERVISION_TIMEOUT	17s	Timeout indicates BLE connection link is considered lost from the last data exchange
PAIR_ADV_DURATION	-	Duration of Pairing advertising is defined as PAIR_TOTAL_DURATION in Table 18
PAIR_ADV_INTERVAL_MIN	30ms	Interval between pairing ads
PAIR_ADV_INTERVAL_MAX	50ms	Interval between pairing ads
RECONNECT_ADV_DURATION	60s	Duration of the stage 1 of low duty cycle advertising
RECONNECT_ADV_BURST	1.28s	Burst duration of the stage 2 of low duty cycle advertising
RECONNECT_ADV_BURST_GAP	300s	Gap duration of the stage 2 of low duty cycle advertising
RECONNECT_ADV_INTERVAL_MIN	35ms	
RECONNECT_ADV_INTERVAL_MAX	55ms	

Table 27: Connection Parameters Table

8.7.7.2 Auto-Reconnection / Keep Awake

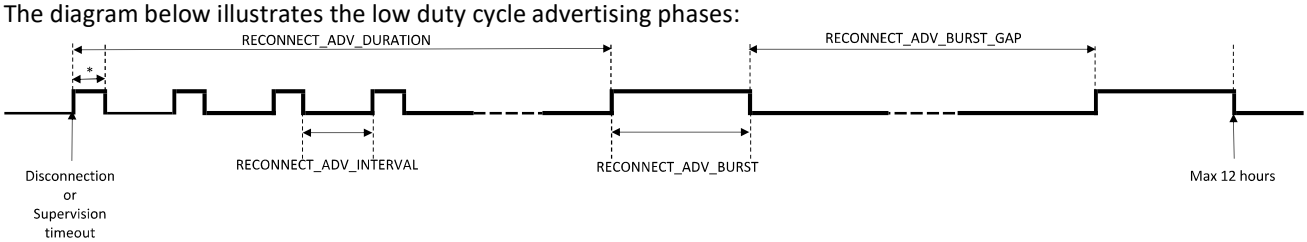
The RCU shall initiate low duty cycle direct advertising for RECONNECT_ADV_DURATION and separated by RECONNECT_ADV_INTERVAL_MIN for any type of disconnections including:

- STB terminates connection by sending LL_TERMINATE_IND.
- Link loss or Supervision timeout occurs.
- There is a POR on the RCU.
- Disconnection triggered by OTA image activation.

In case of unsuccessful reconnection after RECONNECT_ADV_DURATION the RCU will send bursts of direct advertising packets (ADV_DIRECT_IND) for a duration of RECONNECT_ADV_BURST every RECONNECT_ADV_BURST_GAP until a successful reconnection and for a maximum duration of 12 hours.
In case a key is pressed while RCU is in stage 2, then RCU will immediately stops stage 2 and will initiate stage 1.

This ensures that the host (STB) can reconnect with no user intervention. The primary use case is to enable the RCU firmware to be updated during the night. The STB can schedule itself to wake at (say) 3 am and update the RCU firmware. However, this could only work if the RCU and STB can reconnect with no user intervention – hence the periodic ADs.

The diagram below illustrates the low duty cycle advertising phases:



* Phase 1 bursts consist in 1*ADV_DIRECT_IND packet per channel

8.7.8 Auto Pairing Mode

See section 8.5.4 for further details.

8.7.9 Manual Pairing Mode

If the RCU is paired to a BLE STB and the user is simultaneously pressing and holding <<LIVE TV +0>> for at least 5 sec, RCU will enter in BLE pairing state.

If necessary, the link should be disconnected. Below is the sequence of operations that needs to be executed when BLE pairing is initiated:

- RCU should initiate a link disconnection (terminate link) if there is any active connection.
- RCU should attempt to disconnect cleanly and if it fails RCU should force the link disconnection.
- Force pairing sending ADV_IND advertising packets.
- If link is successfully established and pairing procedure was successfully completed, then RCU should be paired and set to BLE operating mode.

The table that is included in the attached document at section 8.5.10, describes the actions that are performed by the RCU upon pairing key combination depending on the RCU state.

RCU should trigger the manual pairing sequence only if <<LIVE TV + 0>> is pressed while all other keys have been released.
In case additional keys are pressed before 5 seconds timeout, then the RCU should not initiate manual pairing mode sequence and stop the timers.

If the pairing keys are still held down for another five or more seconds after pairing already successfully completed, the RCU shall not initiate a new pairing session.

While in manual pairing mode the RCU transmits general BLE advertisement (ADV_IND) packets with an advertisement interval of PAIR_ADV_INTERVAL and repeatedly transmits for up to PAIR_ADV_DURATION.

If pairing procedure succeed, the RCU will operate as a BLE standalone RCU.
If the pairing procedure failed then RCU should indicate the failure through the error blink, go to idle and should remain paired to the previously paired device.

8.7.9.1 BLE IR Assisted Pairing

In Manual Pairing and Auto Pairing mode, the remote shall have the capability of transmitting an IR command that causes the STB to enter pairing mode.
The full details of IR assisted pairing feature are described in sections 8.5.8, 8.5.8.1 and 8.5.8.3.

8.7.10 Clear Pairing

The RCU pairing table can be cleared by issuing a factory reset. The RCU should terminate the link before the pairing entry is cleared. Please refer to section 8.9.2 for the factory reset.

8.7.11 Battery Service

There are two ways of extracting battery information from this RCU. The first method is through UAPI and is detailed in section 8.10.2. The other one defined below is the standard GATT service.

RCU operating voltage is 3.0V, at nominal voltage Remote transmits IR/RF and performs all functionalities. By using Battery Service, Target can ping the battery status from Remote for its processing.
At the operating or Good voltage, the remote allows to program and writes onto the Flash memory.

There are different voltage states as defined in Table 19: Battery thresholds

There are different voltage states as defined in Table 19: Battery thresholds.

Key press & release during low voltage is not indicated differently than key presses during normal battery levels. Further decrease in battery voltage to dead voltage: remote stops working restricting IR/RF transmission as seen in Table 19: Battery thresholds.

BLE Battery service (UUID – 0x2A19) is an ADC based LVD check algorithm. Remote battery algorithm calculates battery level in terms of percentage, wherein 100% is fully charged and 0% is discharged completely.
Below this voltage Remote behavior is erroneous.

Characteristic	UUID		Value	Description
Battery Level	0x2A19		0-100 101-255	RCU Battery Level in percentage Reserved

The standard Battery service includes a single characteristic, Battery Level, which contains the battery level as a percentage from [0, 100].
The percentage needs to be set to (VDD is the measured battery voltage):

Actual Voltage at the chip	Report sent in terms of percentage
3 V ≤ VDD	100 %
3 V < VDD ≤ 2 V	[100 * (VDD – 2)]
VDD < 2 V	0 %

Table 28: Battery Percentages Table

The STB can request the read the Battery Level. Upon receiving a battery level read request from STB RCU will read the current battery level, will report it but will not latch it.

In addition, if Battery Level is configured for notification, the RCU will send a Battery Level notification in the following cases:

- After the RCU has paired
- After the RCU has reconnected
- When the RCU is connected and the power state has changed to one of {V_{OK}, $V_{WARNING}$, $V_{CRITICAL}$}

8.7.12 Data Length Extension (DLE)

DLE feature extends the size of the packets that are transmitted over the air. The size is fixed per report ID and can be different for each report ID.
The purpose of DLE implementation on Apollo product is to specifically extend the size of the HID reports related to audio data (VOICE_INPUT_REPORT_ID = 0x31) and Over the Air Update (OTA) (UAPI_OUTPUT_REPORT_ID = 8) features. The standard HID reports such as the reports related to keypresses or to battery status service should not be impacted.

OTA update is implemented through UAPI. UAPI does not support different sizes for UAPI input (UAPI_INPUT_REPORT_ID = 7) and UAPI output (UAPI_OUTPUT_REPORT_ID = 8) reports. So, all UAPI reports will be impacted by DLE.
Audio data (VOICE_INPUT_REPORT_ID = 31) can be set to a different size.

The DLE should be implemented such a way it can be easily enabled or disabled at code compilation.

By default, the RCU should support packets size that are compliant with non-DLE configuration corresponding to default ATT MTU size of 23 bytes and HID descriptor should be implemented as defined in section 8.7.3.1.1.
RCU should enable DLE feature based on the ATT MTU exchange request sent by the STB. Upon receiving ATT MTU

exchange request, the RCU should set the packet sizes according to the MTU size that has been requested.

Audio packet size should be set to a maximum size of 160 bytes of ADPCM data. If MTU size from STB exceeds this value, then audio data packet size should be set to 160 bytes of ADPCM data. Note that audio data size must be a multiple of 20 bytes.
UAPI_INPUT_REPORT_ID and UAPI_OUTPUT_REPORT_ID size will be set to the value that is requested through the ATT MTU exchange request from STB.

Examples:

- If STB requests an ATT MTU size of 200 bytes, then audio data report size will be set to 160 bytes and OTA report size will be set to 197 bytes (ATT MTU – 3).
- If STB requests an ATT MTU size of 100 bytes, then audio data size will be set to 80 bytes (ATT MTU – 3 with audio being a multiple of 20 bytes) and OTA report size will be set to 97 bytes (ATT MTU – 3).

Minimum value for ATT MTU should be 23 bytes and maximum value for ATT MTU should be 200 bytes.

STB is responsible for:

- reading the RCU HID descriptor whenever ATT MTU size value is changed between 2 consecutive reconnections.
- configuring UAPI MTU size (as part of UAPI Config Param - app ID 0x00, param ID 0x09) to ATT MTU - 4 (3 bytes for ATT header and 1 byte for UAPI length).
- initiating the ATT MTU exchange procedure in case DLE should be enabled.

8.7.13 Quickset implementation

If the RCU operates in BLE mode, then Quickset will be implemented through App Id 0x13.
If a message with App Id 0x11 is received while RCU is operating in BLE mode, then RCU will reject with UAPI_EPERM (0x01) error code.

If UAPI Quickset AppID config command is received while RCU is operating in BLE mode, then RCU will reject the command with the error code UAPI_EINVAL (0x16).

For more details about Quickset App ID 0x13 please refer to section 8.10.1.1.

Key signal data for the universal functions described in Table 1: Functional Key Table shall be sent by STB over the BLE connection to the RCU in ZipIR format and STB shall store them in NVM (FDRA) on the RCU.

8.7.14 Voice over BLE

Please refer first to section 8.10.4 detailing general considerations of voice feature before reading the implementation details of voice feature as described below in this section.

This section describes the implementation details of the audio feature when the RCU has been paired with a BLE Host device and operates as a BLE RCU.

The Voice over BLE audio streaming is controlled by the Host (“Host initiated”) and should be implemented as a “Push-Talk-Release” implementation.

When pressing or releasing the dedicated voice key, the corresponding HID keyboard report will be sent to the host. The host will request audio to start with a **Start Audio Request Message** either after a key press or a key release indication depending on the host implementing a “push to talk” or a “release to talk” feature.

To transfer audio over the underlying BLE link with enough headroom for retransmissions during periods of RF interference, the audio, sampled at 16-bit 16 kHz, is compressed with ADPCM format resulting in a 4:1 compression ratio while maintaining very good audio quality for speech.

Audio is transferred after the **Start Audio Request Message has been received from the host.** Audio will consist in as a stream of messages starting with a **Start Audio Message**, followed by a series of **Audio Data Message**s, and terminated by a **Stop Audio Message**. The Start Audio Message indicates the start of the audio stream and defines the format of the audio data.

If the host is ready to accept audio data, and supports the format defined in the Start Audio Message, it shall respond to the Start Audio Message with a **Confirm Start Audio Message**. Only after receiving the Confirm Start Audio Message, will the first audio data be sent. Each audio data report contains only a sequence of ADPCM compressed audio samples, each pair of 16-bit samples compressed into a single byte.

Audio streaming session should always be initiated when the host send the **Confirm Start Audio Message and the MIC key is pressed** and should be stopped when the host sends a **Stop Audio Message Request which is triggered by receiving MIC key release**.

Upon reception of a **Stop Audio Message Request** from the host the RCU will acknowledge the end of the audio streaming by sending a **Stop Audio Message.**

Voice session shall timeout after voice key timeout as defined in section 8.3.1 if a **Stop Audio Message** was not received by the host. In that case RCU will send a **Stop Audio Message** to indicate the end of the audio streaming.

Multiple keypress scenarios with MIC key will be handled as follow:

- If a key, different than MIC, is pressed before MIC key, then MIC key will be ignored.
- If a key, different than MIC, is pressed after MIC key then it will be ignored.

The sequence chart below illustrates a “Push-Talk-Release” protocol exchange.

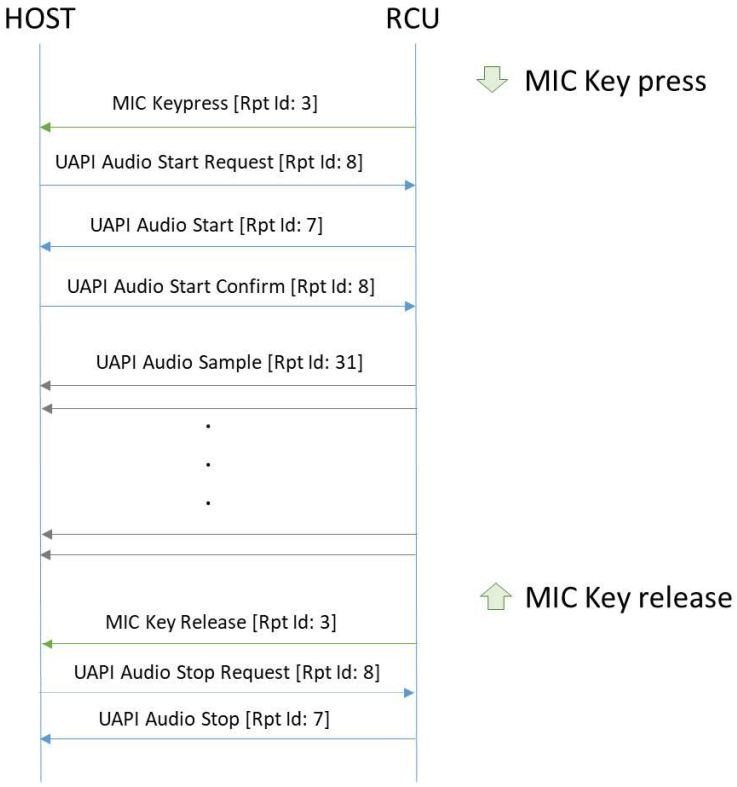


Figure 6: Host Initiated Audio Streaming

8.7.15 BLE STB Power Sequence

8.7.15.1 STB Power States and STB Probing

The STB can be in one of the following states:

- **Power Off State** – STB is powered off (all modules are powered off).
- **Cold-standby** or **Network-standby** - everything is powered off except for the Wifi-BLE combo chip, or network interface, which keeps listening for commands.
- **Hot-standby** - everything is up and running, except for the audio and video outputs.
- **On or Active** – STB is in full operation.

Whenever the STB power key is pressed and RCU is connected to STB, a probe (0x3F) will be sent to the STB. See section 8.7.15.2 for complete details on STB Power sequence.

In order to minimize the response latency, the Slave latency will be set to 0 until the probe response is received.
If STB does not respond to RCU Probe command after a 2s timeout then the slave latency is reset to its default value as defined in section 8.7.7.1 and probe response will be ignored.

The Table below summarizes the conditions that will trigger an STB Probe command and the expected Probe Response.

STB state	BLE Connection state	Probe (0x3F) result
STB Powered Off (No power at all)	SOC is OFF BLE chip is OFF BLE link is disconnected	RCU check whether BLE link is disconnected. Do not try to reconnect. No probe.
Cold-standby / Network standby	SOC is OFF BLE chip is ON BLE link should be disconnected[1]	RCU check whether BLE link is disconnected. Do not try to reconnect. No probe.
Hot-standby (Video Disabled)	SOC is ON BLE chip is ON	ParamID 0xF1: Value 0x00.

	BLE link should be connected[2]	
Active	SOC is ON BLE chip is ON BLE link is connected	ParamID 0xF1: Value 0x01.

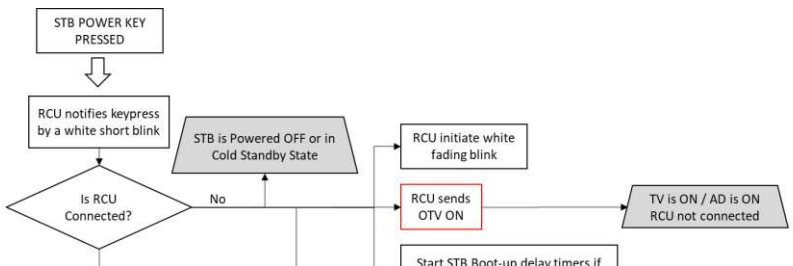
Table 29: STB Power State and BLE STB Probing

[1] Before going to cold standby state the STB should gracefully disconnects the BLE link with the RCU. This will allow the RCU to go into advertising state and to perform the WAKE ON BLE feature.
The STB should also ensure in cold standby state the RCU will not be able to connect the STB from a direct advertising activity.

[2] In connected state a key press on STB power key will send the STB power key report. When STB is in hot standby state, receiving STB Power key will cause the STB to switch from hot standby to active states. When STB is in active state, receiving STB Power key will cause the STB to switch from active state to hot standby.

8.7.15.2 STB Power Key Press Sequence

The diagram below illustrates the power sequence state machine that is implemented when the STB Power key is pressed.



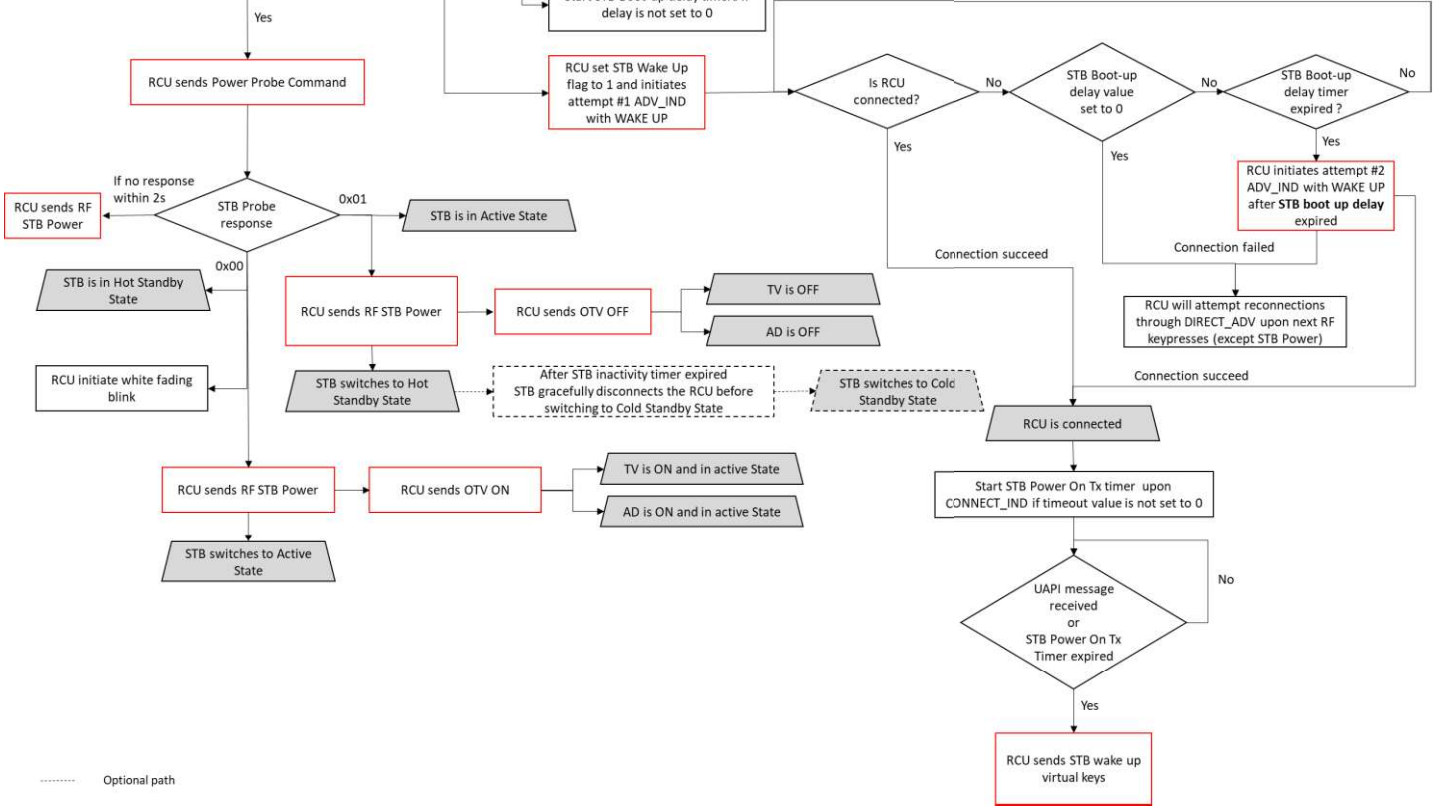


Figure 7: Power Sequence State Machine

Note: RCU should ignore proximity detection events while executing the power sequence state machine.

STB power key down should be indicated by a short white blink as defined in 8.3.2.2.

Case 1: RCU is connected and STB power key is pressed then RCU should send a Probe command to STB.

- STB responds the Probe command with status 0x00, STB is in hot standby state and RCU should:
  - RCU should indicate device powering ON sequence with a fading blink white indication as defined in 8.3.2.5
  - Send RF STB power key press report. When STB is in hot standby state, STB Power key will cause the STB to switch from hot standby to active state.
  - Send OTV ON macro to switch the rest of the devices ON.
  - All devices should be switched ON.

- STB responds the Probe command with status 0x01, STB is in active state and RCU should:
  - Send RF STB power key press report.
    When STB is in active state, STB Power key will cause the STB to switch from active state to hot standby states.
    After a timeout STB will go to cold standby.
    Before going to cold standby state, the STB should gracefully disconnect the BLE link with the RCU. This will allow the RCU to go into advertising state and to perform the WAKE ON BLE feature.
    The STB should also ensure in cold standby state the RCU will not be able to connect the STB from a direct advertising activity.
  - Send OTV OFF macro.

- RCU should implement a 2 second timeout on the probe command. If the probe command timeout and RCU does not receive a response to the command, then the RCU should exit the OTV sequence and send RF STB Power key.

Case 2: RCU is disconnected when STB power key is pressed then STB is either powered OFF or in cold standby state.

- RCU does not send a Probe command and does follow the sequence:
    - RCU should indicate device powering ON sequence with a fading blink white indication as defined in 8.3.2.5
    - Send OTV ON macro and start timer tracking for STB boot-up delay.
    - STB should start scanning for incoming connection from RCU as fast as possible in order to allow the RCU to connect.
    - RCU attempts to Wakeup STB. It set the flag STB Wake Up and reconnect it through indirect advertising scheme (ADV_IND).
      It is required the STB always accept incoming connections from a paired RCU that sends indirect advertising packets.
        - If connection succeed: All devices should be ON, RCU is connected to STB and RCU exits the Power sequence state machine.
        - If connection failed:
            - If STB boot-up delay is not set to 0, then wait for STB boot-up delay timer to expire.
            - If STB boot-up delay is set to 0, then RCU exits the Power sequence state machine.
    - After STB boot-up delay the STB should be scanning for incoming connection from RCU in order to allow the RCU to connect.
    - RCU attempts to Wakeup STB and reconnect it through indirect advertising scheme (ADV_IND).
      It is required the STB always accept incoming connections from a paired RCU that sends indirect advertising packets.
        - If connection succeed: All devices should be ON, RCU is connected to STB. Then RCU should send the virtual key “STB WakeUp”.
        - If connection failed: All devices should be ON, RCU indicates connection failure as defined in section 8.3.3.4. RCU should initiate a reconnection through direct advertising upon the next RF keypress different than STB power key.

- Note that STB boot-up delay are defined through UAPI Configurable parameters as defined in section 8.10.2.

8.7.15.3 STB Wake Up Flag Behavior

STB Wake Up flag is used to indicate whether the STB Power key was pressed while the RCU is disconnected (i.e. while STB is booting).

STB Wake Up flag should be set to True if and only if the STB Power key is pressed while a bonded RCU is disconnected.
STB Wake Up flag should be reset to False under the following circumstances:

- Flag is explicitly written by STB through UAPI RCU internal flags command (see section 8.10.2.1).
- Factory reset of RCU or Unpairing (deletion of bonding data) of RCU.
- Upon a 5 minute timeout after the STB Power key is pressed and wakeup flag set, it will be cleared if still set.

8.7.15.4 TV Power key

IR key code for TV Power key will be mapped to Power Toggle trigger ID (0x01) in the key map record.

8.8 RF4CE Operating Mode and Configuration

8.8.1 RF4CE RF IDs

The following table show the device information that is exchanged between RCU and STB, in typical processes such as binding/pairing.

Parameter	Value	Description
	RCU Parameters	
Vendor ID	0x10EE	Vendor ID send by RCU

Vendor String	UEIC	Vendor String send by RCU
User string	R32845XX	User string send by RCU. It will use RCU model number. X masks different versions (Short & Long form factors) and logo variations.

Table 30:RF4CE Ids

8.8.1.1 User String Details

USER STRING														
BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8	BYTE 9	BYTE 10	BYTE 11	BYTE 12	BYTE 13	BYTE 14
R	3	2	8	4	5	X	X	0x00	0xFF	0xFF	0xE0	0x00	0x00	0x00
APPLICATION SPECIFIC USER STRING								NULL	VENDOR ID FILTER		MIN/MAX CLASS FILTER	MIN LQI FILTER	RESERVED	
									DEFAUT VALUES SPECIFIED IN ZRC 2.0 SPECS					

Table 31: RF4CE User String

The Remote shall pair only with a STB which announced the following device type. RCU should not filter on STB User String.

Parameter	Value
STB Parameters	
Device Type	0x09

8.8.2 Deviation from the standard

Historically LGI's RF4CE implementations use of 2 additional RF4CE channels, to make the total of 5 channels. The five channels are CH11 (2405MHz), CH15 (2425MHz), CH20 (2450MHz), CH25 (2475MHz), CH26 (2480MHz).

8.8.3 RF4CE Pairing

The RCU shall pair with Target devices which support the ZRC2.0 or ZRC1.1 profile.

For ZRC 2.0 pairing the RCU follows the ZRC2.0 discovery and binding procedure as described in ZigBee RF4CE: ZRC Profile Specification v2.0. However, Liberty Global requires that a ZRC2.0 STB is only accepting pairing requests in certain circumstances. For this reason, and to be backwards compatible with legacy STB's, this RCU also supports IR assisted pairing.

In the discovery process, the RCU will disclose the profile identifier 0x01 (ZRC1.x) and 0x03 (ZRC2.0).

The aim is to establish RCU-STB pairing in the most transparent and intuitive to the user manner. Hence pairing will be initiated in two possible scenarios:

1) RCU with empty pairing table (i.e. out-of-the-box scenario) – refer to section 8.5.4 for further details.
2) Already paired remote, i.e. pairing table has an entry. => User executes RF4CE Manual pairing by pressing one of the key combinations defined by 8.3.4.2 and that is supposed to trigger an RF4CE pairing. RF4CE manual pairing is described in section 8.8.4.

Note that only single-target pairing will be supported on the RCU side.

Visual feedback for pairing is detailed in section 8.3.3.3 and IR assisted feature is detailed in section 8.5.8.

Visual feedback for pairing is detailed in section 8.3.3.2 and IR assisted feature is detailed in section 8.5.8.

If Pairing is triggered by a single RF key press (automatic pairing case detailed in section 8.5.4), then RCU will send the key code of the key that originated the pairing in PAIRING_KEY_INFO vendor specific attribute (see Table 32: Vendor Specific ZRC Commands).

The RF keys that are pressed before the pairing is completed should be ignored; except for the key that initiated pairing.

Pairing related mandatory notes:

- After the pairing is successful, the remote will wait 1 second and then initiate a Get Pending Message request, to check for any pending messages.
- Once paired the RCU shall remain in paired state unless it is manually unpaired by the user. Removing batteries should not remove pairing info.
- The remote control can store only one pairing entry in the pairing table.
- Encryption beyond what is provided by RF4CE is not required. The RCU shall encrypt all data provided that the STB supports RF4CE encryption.

Pairing operating voltage is described in Table 19: Battery thresholds. If battery level does not allow pairing to start, then the LED will not display any blinks. By not displaying any visual feedback, the RCU will signal clearly that the batteries are not good.

8.8.3.1 RF4CE only IR Assisted Pairing protocols

The full details of IR assisted pairing feature are described in sections 8.5.8, 8.5.8.2 and 8.5.8.3.

8.8.4 Manual Pairing

If the RCU is paired to an RF4CE STB and the user is simultaneously pressing <<LIVE TV +0>> for at least 5 sec, RCU will enter in RF4CE pairing state.

The table that is included in the attached document at section 8.5.10, describes the actions that are performed by the RCU upon pairing key combination depending on the RCU state.

If pairing procedure succeed, the RCU will operates as an RF4CE standalone RCU.
Upon failure the RCU should indicate the failure through the error blink and go to idle mode. Moreover, RCU should remain paired to the previously paired device (see sections 8.5.4, 8.5.5, 8.5.7 for further details).

In case additional keys are pressed before 5 seconds timeout, then the RCU should not initiate manual pairing mode sequence and stop the timers.

8.8.5 ZRC 2.0 Profile Implementation

The software implementation shall follow the ZRC2.0 Profile Protocol and the Generic Device Profile Version 2.0 specification. Since ZRC 2.0 is backward compatible with ZRC 1.1, the RCU will support ZRC 1.1 as well.

8.8.6 Key Repeat Messages

The RCU shall send key repeat messages every 100ms whilst any key is held down.
This applies only to the RF4CE keys.
IR keys may or may not send repeat frames and it will depend on the IR protocol being used.

8.8.7 UAPI Polling

For maximum battery lifetime, all communication is initiated by the RCU. However, there are multiple scenarios where Host device needs to be able to initiate a request.
For Host device to be able to deliver messages to RCU, the UAPI polling mechanism is implemented through the sending of a “Get Pending Messages” message from RCU to STB.
The polling of the Host device will be managed through 2 different mechanisms: the manual polling and the automatic polling.
Upon UAPI polling the battery level will be assessed by the firmware. If battery level is latched to Vcritical UAPI polling will stop.

8.8.7.1 Manual Polling

Manual polling is a polling of the Host device that is based on a user releasing specific keys. The RCU should poll Host device for pending messages upon the release of the following key list:

- OK key
- Side key
- STB Power key

Upon the key release the RCU should sends “Get Pending Messages” to the Host device until all the pending messages have been received or an error condition has been detected.

The proximity detection will also generate a “Get Pending Messages” command 100ms after sending the STB probe command. Probing upon proximity is described in detail in section 8.10.5.

In case of STB Power key, a “Get Pending Messages” command will be sent only if a probe response indicates the STB is in standby state. If STB is notified as in Active State, then the command should not be sent. For further details please check section 8.8.13.2.

If no response is received from STB 100ms after a "Get Pending Messages" command has been sent, then RCU goes back to idle mode.

8.8.7.2 Automatic Polling

The RCU will poll periodically with Get Pending Messages for the purpose of getting any pending messages from the Host device.
The default period of the polling should be set to 30 seconds.
It should be possible to enable or disable the automatic polling from the Host device.
It should also be possible to configure the automatic polling period from the Host device. The configured value should be stored in NVM and should be persistent to a POR of the RCU.

If no response is received from STB 100ms after a "Get Pending Messages" command has been sent, then RCU goes back to idle mode.
During voice session auto polling should be paused and should restart automatically at the end of the audio session.
During OTA auto polling is paused and should restart after the OTA. Similarly, if STB is not responsive and RCU issued 10 OTA data request attempts with no success then auto polling should be resumed until the next data request response from STB.

8.8.7.2.1 Automatic Polling Period

Automatic Polling Period, also named Polling Period, is configured through UAPI with App ID=0x00 and Param Id =0x03. The polling period value is a multiple of 62.5 milliseconds, and length of this field is 4 bytes.

The complete details are shared in UAPI Protocol Specification .

Once in paired mode, the RCU wakes up periodically and sends a **Get Pending Messages** message to check for **Pending Messages** at the STB

Messages at the STB.
The default wake-up interval, so called “Polling Period”, should be 30s.

Shorter polling periods have a strong negative impact on RCU’s battery life. Longer polling periods will cause a higher latency impact on features such as OTA and Quickset.
A Polling Period set to 0 should stop the automatic polling feature.

8.8.7.2.2 Automatic Polling State

The automatic polling feature can be enabled or disabled with Automatic Polling State configuration parameter. Automatic Polling State, also designated by Auto Poll State, is configured through UAPI with App Id=0x00 and Param Id=0x05.

Auto Poll State must be set to true in order to enable automatic polling. To disable the automatic polling, the parameter will be set to false.

Note that the Automatic Polling Period configuration parameter described in section 8.8.7.2.1 must be set to a value greater than 0 for enabling Automatic Polling.

The Auto Polling can be enabled and disabled also from the RCU by a key-combo <<VOLUME DOWN + MUTE>> <9><3><0/1>

1. Press and hold <<VOLUME DOWN + MUTE>> until 2 Short White LED blinks.
2. Enter <9>→<3>→<0> to Enable the auto polling.
3. Enter <9>→<3>→<1> to Disable the auto polling.
4. If success: The LED will display 3 Short White blinks.
   If failure: The LED will give an error blink as defined in section 8.3.2.3.

8.8.8 Vendor specific ZRC commands

The following table specifies the vendor specific ZRC commands that shall be used by the remote. The commands shall be sent in the vendor reserved areas of the ZRC command protocols. The vendor specific ZRC approach is implemented in order to maintain backward compatibility with legacy LGI Horizon STB. See section 8.8.9 and 8.8.10 for further details on command usages.

Command					***Payload***			
Function	**Transmission Type**	**Purpose**	**ZRC Control**	**RC Cmd Code**	**Byte2**	**Byte3**	**Byte4**	**Byte5**
BATTERY_LOW	Unicast	Send at certain battery voltage level. See section 8.8.9.	0x01	0xe0	2	0	0	0
BATTERY_TOO_LOW	Unicast	Send at certain battery voltage level. See section 8.8.9.	0x01	0xe0	3	0	0	0
PAIRING_KEY_INFO	Unicast	Sent by RCU to STB after pairing completed. Key numbers are as per Table 1: Functional Key Table.	0x01	0xe0	4	[Key Nr.]	0	0
SHOW_PAIRING_PARAMETERS	Unicast	Show RCU's pairing table	0x01	0xe0	11	0	0	0
SHOW_SOFTWARE_VERSION	Unicast	Show RCU Firmware Software Version (i.e. 7060)	0x01	0xe0	13	0x1B	0xAB	Point release
SHOW_CURRENT_TV_CODESET_NUMBER	Unicast	Show TV Code ID that is currently downloaded on RCU	0x01	0xe0	17	XX	XX	0
SHOW_CURRENT_AV_CODESET_NUMBER	Unicast	Show AMP Code ID that is currently downloaded on RCU	0x01	0xe0	18	YY	YY	0

_CODESET_NUMBER		currently downloaded on RCU						
SHOW_POLLING_PERIOD	Unicast	Show RCU current polling period (in seconds)	0x01	0xe0	20	[0-30]	0	0

Table 32: Vendor Specific ZRC Commands

8.8.9 Power Status reporting

The RCU detects battery voltage level states according to Table 19: Battery thresholds
When the RCU is paired with ZRC2.0 or ZRC1.1 STB, the low battery voltage levels are reported to the STB by ZRC Vendor extensions, from Table 32: Vendor Specific ZRC Commands.

The RCU sends low battery warnings over RF4CE, as noted below, after IR- or RF-key is sent and upon the voltage level state conditions that are defined in Table 19: Battery thresholds.

BATTERY_LOW indication is sent when $V_{WARNING}$ threshold has been crossed.
BATTERY_TOO_LOW indication is sent when $V_{CRTICAL}$ threshold has been crossed and is sent to STB only the first time the threshold is crossed.

The RCU shall not send low battery warnings for stuck key condition ~~or pairing key-combo~~.

8.8.10 Status/diagnostics reporting to STB

The RCU will send diagnostics information to STB by ZRC Vendor extensions, according to Table 32: Vendor Specific ZRC Commands.

Programming Sequence:
1. Press and hold <<VOLUME DOWN + MUTE>> until the 2 short white blinks.
2. Enter <9>→<1>→<2> to send diagnostics info to STB.
3. The LED will blink white four times to indicate success.

Results/Conditions:

1) RCU unicasts SHOW_PAIRING_PARAMETERS command to the paired STB.
2) RCU unicasts SHOW_SOFTWARE_VERSION to the paired STB.
3) RCU unicasts SHOW_CURRENT_TV_CODESET_NUMBER to the paired STB.
4) RCU unicast SHOW_CURRENT_AV_CODESET_NUMBER to the paired STB.
5) RCU unicast SHOW_POLLING_PERIOD to the paired STB.

8.8.11 Quickset implementation

If the RCU operates in RF4CE mode, then STB should be able to configure the RCU to operate Quickset through App Id 0x13 or App Id 0x11 by sending the UAPI configuration message: Quickset AppID config.

UAPI Quickset AppID config command is defined in Table 35: UAPI App ID 0 Payload Table.

If RCU is not latched and a Quickset AppID message is received before the UAPI Quickset AppID config command, then RCU should latch to the application ID received in the first Quickset message from STB.

If UAPI Quickset AppID config command is received after RCU latched to Quickset application ID:

- If the command set the same AppID then command will be ignored.
- If the command requests a switch from AppID 0x11 to 0x13, then RCU should erase the existing data related to AppID 0x11 from memory and start to be functional with AppId 0x13.
- If the command requests a switch from AppID 0x13 to 0x11, then RCU should reject the command with error code UAPI_EPERM (0x01).
- If RCU receives a different Quickset App Id that the one the RCU is latched, then the RCU will reject the message with error code UAPI_EPERM (0x01).

This will ensure the backward compatibility with the previous STB implementation which does not implement the UAPI Quickset AppID config command.

The RCU will unlatch the Quickset App Id in case:

- RCU is unpaired.
- factory reset is applied.

The Quickset database coverage is independent of Quickset implementation. Quickset flow through App ID 0x13 is detailed in section 8.10.1.

The sections from 8.8.11.1 to 8.8.11.4 will define the topics that are specific to Quickset implementation using Remote API App Id 0x11.

8.8.11.1 Quickset Supported Features

This remote will support the following Quickset features:

- **Code Download-** This feature will download a set of IR codes to the remote.

- **Code Test** – this feature allows the user to test and see whether a selected code-set will work with a specific device prior to assigning it to a mode on the remote.

- **Copy Settings** – This feature will copy settings of the Quickset codes to the Remote.

- **Volume Lock –** This feature allows the user to setup the remote-control volume+, volume- and mute keys to exclusively control volume up and volume down on a user-specified device.

Example: In a system with a TV, STB, and audio receiver, when the remote-control volume lock is set to the audio receiver, pressing volume+ or volume- on the remote will result in the volume changing on the audio receiver, regardless of which mode the remote control is currently in.

- **Auto Setup** - The Quickset library in UAPI provides a feature of automated lookup by identifying the attached devices by means of HDMI-EDID look-up or using CEC control information. Automated-lookup will allow the programming of devices that support EDID/CEC information automatically without going through the standard device-brand-model lookup route. CEC/EDID information of connected device is obtained and passed to respective API. Based on the information passed, API will retrieve code sets for the device connected which can be used for programming the remote.

- **Manual Setup** - Manual setup assists the user with a UI flow and enables to setup RCU manually. The user is instructed to search for brand/model of his device, and the programming sequence is displayed with the correct ID to the user.

- **Optimized Setup Maps -** Optimized setup map assists the user with a UI flow, to setup his TV and AMP without searching for and the mode/brand. After user selects brand, Quickset starts to test RCU keys, and by the user feedback finds the right device and programs it to the RCU.

- **One Touch View (OTV) –** OTV feature will be executed by pressing STB Power key depending on the state of STB.

8.8.11.2 Quickset Initialization

The RCU must be paired to enter Quickset operation. Immediately after each successful pairing, the remote will do the following:

- Remote will send the command “Get Pending Messages “to STB and it will try to download any Quickset

pending messages.

- Optionally, STB may send UAPI Quickset AppID config command to set the Quickset application ID to 0x11 or 0x13. Note that if the command is not received, then the remote will be set to the first received Quickset application ID. The Quickset operation using App Id 0x13 is described in section 8.10.1.1.
- If remote is set to App Id 0x11, it will engage into Quickset operation only if it receives some Remote API message (UAPI AppID 0x11), otherwise it will resume normal operation.
- During Quickset operation, remote will exit back to normal operation if no message is received after "Get Pending Messages" message is sent.

8.8.11.3 Volume Lock feature

This feature allows programming volume lock to a specific mode over Quickset.
Quickset shall program the RCU with the correct codeset for detected devices (automatic or semi-automatic setup).
Since this is a modeless remote control, after a TV code is downloaded and programmed, the volume keys should now get locked to TV.
If an audio code is downloaded and programmed, the volume should get locked to Audio.

The Volume lock will move back to the TV if an existing AVR code is removed (manually or automatically) from the RCU.

It shall be also possible for the user to overwrite the default Quickset Volume lock, via action from the STB GUI, i.e. via an application trigger and a command from the STB.

If the code set is updated or removed using quickset, the volume lock programming should be handled through Quickset as well.

8.8.11.4 Quickset Test Mode (UEI Internal)

Behavior of test keys when in Quickset ID setup test mode are:

- For TV the following keys will be available:

  - Power, Volume Up, Volume Down, Mute.
- For AMP the following keys will be available:
  - Volume Up, Volume Down, Mute.
- OK key will send STB select data and then exit Quickset test mode.
- Up/Down/left/right keys will send STB Up/Down/left/right data and then stay in Quickset Test mode.
- All other keys will be ignored.

Timeout of Quickset Test Mode is set to 30 seconds.

8.8.12 Voice over RF4CE

8.8.12.1 RF4CE audio protocol overview

Please refer first to section 8.10.4 detailing general considerations of voice feature before reading the implementation details of voice feature as described below in this section.

This section describes the implementation details of the audio feature when the RCU has been paired with an RF4CE Host device and operates as an RF4CE RCU.

When pressing or releasing the dedicated voice key, the corresponding UAPI messages will be sent from RCU to the host. Using the messages detailed below, the host can request audio sampled from the built-in microphone to be streamed to the host while the voice key is being held.

When the Voice key is pressed the RCU shall establish a voice session with the STB as illustrated by Figure 8: RCU_Initiated Audio Streaming.

After the audio session is successfully established, the RCU shall start capturing audio at the first **Audio State Message**

After the audio session is successfully established, the RCU shall start capturing audio at the first **Audio State Message** and shall stream audio data as long as the Mic button is pressed, or until the Voice stuck-key timeout, as defined in section 8.3.1, expires.
The RCU shall stop streaming audio data when the Mic button is released as illustrated by Figure 8: RCU_Initiated Audio Streaming.

If the Voice button is held longer than Voice button stuck interval. The session will end with a release message, as if the voice button is released.

Audio is transferred as a stream of messages, starting with a **Start Audio Message**, followed by a series of **Audio Data Messages**, and terminated by a **Stop Audio Message**. The Start Audio Message indicates the start of the audio stream and defines the format of the audio data.

If the Host device is ready to accept audio data, and supports the format defined in the Start Audio Message, it shall respond to the Start Audio Message with a **Confirm Start Audio Message**. The Confirm Start Audio Message shall define the same format in its payload as the Start Audio Message it is a response to. If the Host Device is not ready to accept audio data, it will return an appropriate error message:

- ESRCH if Audio feature is not supported at all.
- EINVAL if the audio format defined in the Start Audio Message is not supported.
- EBUSY if Host device does not currently have the resources available to sink the audio stream.

The first Audio Data Message should be sent only after the Confirm Start Audio Message has been received from the Host Device.
This and subsequent Audio Data Messages are comprised of audio data only, adhering to the audio format defined by the preceding Start Audio Message.
The Stop Audio Message indicates the end of the audio stream to the Host Device.

When audio streaming is started from the RCU, the RCU will lock to the current RF4CE channel and it will disable RF4CE channel agility. This common practice is done to control and minimize the delay and jitter in audio packet transmission. When the channel agility is disabled, the RCU will not change channel in case of No ACK.

Since RF4CE channel agility is disabled at both ends, the Host device must be capable of disabling the channel agility.

The audio packets are uploaded at an average inter-packet gap of approximately 10.5ms. Each packet has 84 bytes of data, so the # of packets sent in 6 seconds is approximately 6000/10.5 = ~571 packets. These packets are being transmitted as Unicast single channel with Acknowledgement. Please note that for audio packet transmission a RF retry period of 30ms is applicable instead of the normal 100ms.

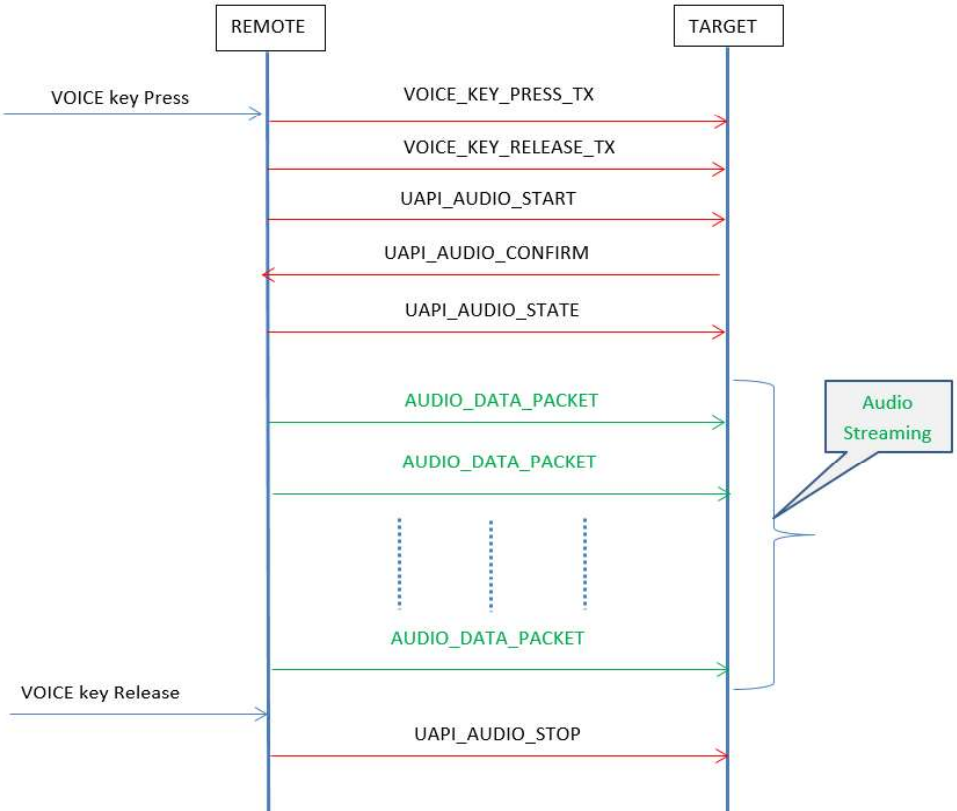

REMOTE
TARGET
VOICE key Press
VOICE_KEY_PRESS_TX
VOICE_KEY_RELEASE_TX
UAPI_AUDIO_START
UAPI_AUDIO_CONFIRM
UAPI_AUDIO_STATE
AUDIO_DATA_PACKET
Audio Streaming
AUDIO_DATA_PACKET
AUDIO_DATA_PACKET
VOICE key Release
UAPI_AUDIO_STOP

Figure 8: RCU_Initiated Audio Streaming

8.8.12.2 UAPI Audio Additional Features

Audio is always 'pushed' from an audio source to an audio sink. However, UAPI allows an audio sink to request an audio transfer using the **Request Audio Message**.
This RCU should not support the **Request Audio Message** command, and the RCU Microphone will send audio only when the audio request originates from the RCU.

Similarly, UAPI allows an audio sink to send a **Request Stop Audio Message** to the source in order to request it to stop sending Audio Data Messages.
However, this feature will not be supported in this product.

The **Query Audio Support Message** can be used to query the audio capabilities of a Host Device which in turn will respond with an **Audio Support Response Message**.

The Audio Data Messages include a sample counter. The sample counter is reset to 0 before the first Audio Data Message following a Start Audio Message is sent. Subsequently, the number of *captured* audio samples is added (modulo 65536) to the sample counter. Note that captured samples are counted, even if they are not transmitted. The sample counter in an Audio Data Message reflects the counter value for the first sample in the payload of this message; the sample counter can be used by the Host Device to conceal lost samples.

8.8.12.3 Host Audio Specifics

From X19 Firmware revision the RCU should be able to determine, from the RF4CE user string - Samsung&UPC, if it is paired to a Samsung Selene STB or to a different STB.
If RCU is paired to Samsung Selene STB then RCU should ensure a minimal gap of 4ms between an audio data packet and the acknowledgement of the previous audio data packet.
This is due to the slowness of the STB which cannot switch between radio transmit and radio receive paths fast enough.
This limitation causes the audio data packet following the acknowledgement of the previous audio data packet to be missed most of the time.

8.8.13 RF4CE STB Probing

8.8.13.1 STB Power States and STB Probing

The STB can be in one of the following states:

- **Power Off State** – STB is powered off (all modules are powered off).
- **Cold-standby** or **Network-standby** - everything is powered off except for the RF4CE & IR receivers, or network interface, which keeps listening for commands.
- **Hot-standby** - everything is up and running, except for the audio and video outputs.
- **On or Active** – STB is in full operation.

STB state	Probe result	RCU LED feedback
STB Powered Off	No response from STB	RCU will issue a white LED fading
Cold-standby / Network standby	STB/ UAPI are unable to process the RCU-probe-packet. The RCU packets are acknowledged by STB at RF4CE level. But no UAPI response message is received within 100ms.	RCU will issue a white LED fading. In this case, the RF4CE ACK would be received from the STB, however, there would not be a UAPI response.
Hot-standby	UAPI shall be informed by the middleware that	RCU will issue a white LED fading.

	the STB is in a standby state. UAPI on the STB will communicate to UAPI on the RCU that the STB is in standby (ParamID 0xF1: Value 0x00).	
Active	UAPI is informed by the middleware that the STB is in an active state. UAPI on the STB will communicate to UAPI on the RCU that the STB is in active state. (ParamID 0xF1: Value 0x01)	No LED blinks.

Table 33: STB Power State and RF4CE STB Probing

8.8.13.2 STB Power Key Press Sequence

Physical Key co-ordinate of STB power key – 0x03 – QS should load POWER OFF macro onto this key.
Virtual key co-ordinate that can be used for Virtual key – 0xF5 – QS should load POWER ON macro on to this key.

For the STB to be able to deliver a feedback message, the RCU sends a Get Pending Messages message 100ms after sending the STB Power Probe command. RCU will set the timeout for the Get Pending message to 100ms i.e. if no response is received within 100ms, RCU will assume that STB is in Standby state.

If the macros are not present, the RCU needs to send only the STB power toggle command upon STB Power key press.
Upon STB Power key release a manual polling will be performed as specified in section 8.8.7.1 except for the case where the STB power probe response indicates the STB is active and should be switched off by the STB power key press.

Each scenario is described according to the STB state in detail below.
Note that LED blink is described in Table 33: STB Power State and RF4CE STB Probing and intentionally not described in the text below.

STB is in cold standby:

- User picks up the remote and presses STB power key.

- The remote requests STB status and STB does not respond to request within 100ms.
- STB power toggle (RF) is sent.
- Depending on the OTV macro availability, RCU sends discrete input change command (IR) to TV and/or Amp followed by discrete Power On (IR) to TV and/or Amp and then the RCU waits for a brief delay and sends discrete input change command (IR) to TV & Amp (The OTV macro is programmed to the RCU after Quickset discover only if the rule for corresponding device is available and discrete IR codes are found in the downloaded rule).
- STB sends CEC ON and CEC Input change to TV/Amp.
- Upon STB power key release RCU should initiate a manual polling.
- In cold standby the STB RF4CE chip activity time is reduced to 20%. In this condition it is possible the STB does not acknowledge packets from RCU. In case of missing acknowledgment, the RCU should jump to the next step of the STB Power key Sequence.
  For example if Power Probe key down/up is not acknowledged the RCU will send GetPendingMessages as it is the step of the state machine following the Power Probe Command.
  Note that if a key press is not acknowledged then the associated key release will not be sent.

STB is in hot standby:

- Same as Cold Standby except STB respond to STB Power Probe command with a status of 0x00.

STB is in network standby:

- Same as Cold Standby.

STB is active:

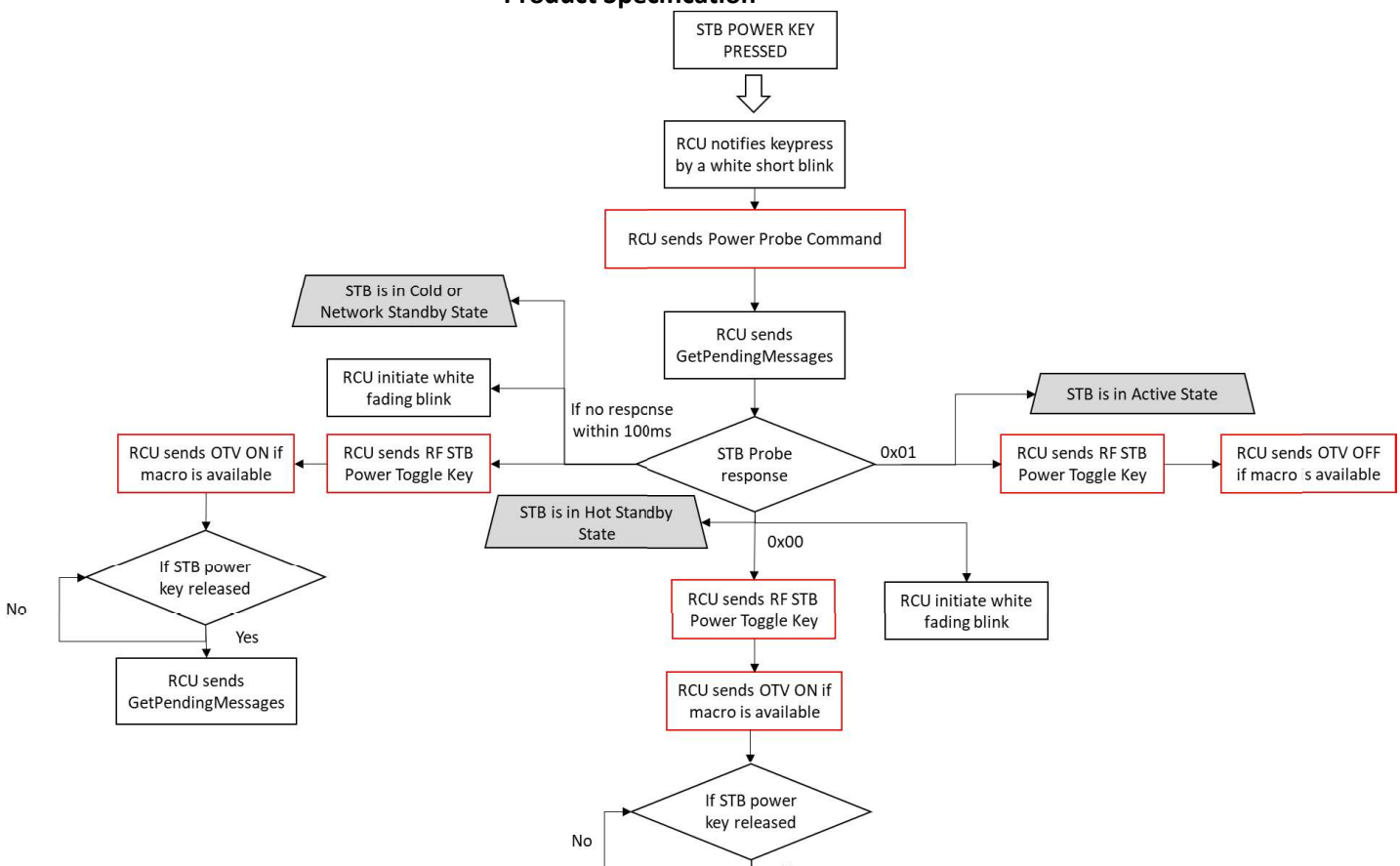
- User picks up remote and presses STB power button.
- The remote checks STB status and STB respond with 0x01 status to STB Power Probe Command.

- The remote checks STB status and STB respond with 0x01 status to STB Power Probe command.
- STB power toggle (RF) is sent.
- RCU should send discrete power off (IR) to TV and Amp.
- STB sends CEC Off to TV and Amp.
- STB turns off (this should happen only once the CEC Off commands are sent).

Notes:

- In the OTV application, the discrete input change command is sent before the discrete Power On. This covers the situation when the TV is already ON. However, in case the TV is off, then it will take some time to Power On. The time needed for this will vary from one TV to another and it may take a few seconds. The timeout in the OTV macro for this can be varied by the customer from the STB. The recommended value is 7 seconds to cover >95% of devices. By that time, the user may be pointing the RCU in some other direction. Also, when the RCU resends the discrete input command after a few seconds, there is a possibility of getting more onscreen input change messages on the TV. In case the user presses the STB Power key again within this timeout, the OTV macro is cancelled and the All Off macro is transmitted to prevent STB and TV getting out-of-sync power states (defined at the beginning of this paragraph as well).
- RCU should ignore proximity detection events while executing the power sequence state machine.
- Keypress scenarios while STB power sequence is ongoing are detailed in section 8.10.1.38.10.1.

The diagram below is illustrating the STB Power key Sequence:


STB POWER KEY PRESSED
RCU notifies keypress by a white short blink
RCU sends Power Probe Command
RCU sends GetPendingMessages
STB Probe response
If no respcnse within 100ms
STB is in Cold or Network Standby State
RCU initiate white fading blink
RCU sends RF STB Power Toggle Key
RCU sends OTV ON if macro is available
If STB power key released
No
Yes
RCU sends GetPendingMessages
0x01
STB is in Active State
RCU sends RF STB Power Toggle Key
RCU sends OTV OFF if macro s available
STB is in Hot Standby State
0x00
RCU sends RF STB Power Toggle Key
RCU initiate white fading blink
RCU sends OTV ON if macro is available
If STB power key released
No

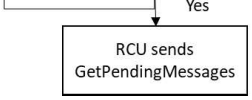


Figure 9: RF4CE STB Power Key Sequence Flow Diagram

8.8.13.3 TV Power key

TV Power key is treated as the other IR keys and will send the corresponding IR TV Power toggle signal.

8.9 Standard Features

8.9.1 Feature Reset

Resets all remote control's configurable features. In the Dual stack FW, the RCU's configurable features are STB boot up delay, STB Power ON Tx Timeout, the last STB Probe Status, polling period, automatic polling state, proximity sensor activation, RCU internal flags.

1. Press and hold <<VOLUME DOWN + MUTE>> until the white LED short blink twice.
2. Enter <9>→<7>→<7> to initiate feature reset.
3. The white LED will blink short blink twice (setup confirmation Blink) to indicate success.
4. Feature reset failure will be indicated by a white error blink.

8.9.2 Factory Reset

Reset remote control to factory defaults.

- If already paired, clear pairing information. If devices are connected a disconnection request will be sent prior to pairing information is cleared.

- All Quickset settings are cleared.
- Universal TV settings are cleared back to the default IR codes.
- All settings are returned to the default values and behavior.

Programming Sequence:
<<LIVE TV + REWIND>> → **Press and hold together for more than 5 seconds**.
If successful: (White LED will blink 4 times to indicate success).
If failed: (White LED will display an Error blink).
After the reset is initiated, there is an additional delay of approximately 2 seconds before feedback. This is due to the additional time required for erasing and writing FDRA.

Note: for Manual Pairing the key combo is mentioned here for completeness:
<<LIVE TV + 0>> → **Press and hold together for more than 5 seconds.**
See section 8.3.4.2 for further key combo definitions.

8.9.3 Software Version Verification

This feature is used to verify the SW version of the remote control.
1. Press and hold <<VOLUME DOWN + MUTE>> until the white LED blinks short twice.
2. Enter <9>→<8>→<3> to initiate software version verification.
3. Enter digits 1 through 8. Count and note the number of blinks displayed by the Red LED if RCU is in RF4CE operating mode or by the white LED otherwise.
4. The number of short blinks after pressing each 1, 2, 3, 4, 5, 6, 7 and 8 keys is the 8 digits (ABCD-XX-YY) SW version of the remote control. Out of these first 4 digits indicate SW # (i.e. ABCD) and digits 5 and 6 (i.e. XX) indicate X-release version. Digits 7 and 8 (i.e. YY) indicate the minor release version.
5. Software version verification failure will be indicated by a white error blink.

8.9.4 Short Software Version Verification

This feature is used to show the short SW version of the remote control.

1. Press and hold <<VOLUME DOWN + MUTE>> until the white LED blinks short twice.
2. Enter <9>→<8>→<4> to initiate software version verification.
3. Enter digits 1 through 4. Count and note the number of blinks displayed by the Red LED if RCU is in RF4CE operating mode or by the white LED otherwise.
4. The number of short blinks after pressing each 1, 2, 3, 4, keys is the 4 digits (XX-YY) SW version of the remote control. Digits 1 and 2 (i.e. XX) indicate X-release version. Digits 3 and 4 (i.e. YY) indicate the minor release version.
5. Short software version verification failure will be indicated by a white error blink.

8.9.5 TV Menu Shift

The feature is used to select the correct HDMI input on the TV, when the TV uses an on-screen display for HDMI input selection. This function applies only when the <TV Input> key is pressed and when the RCU contains a TV model-specific device code supporting navigation of the on-screen Input selection menu.

1. When the <TV Input> key is pressed, the RCU shall send the TV input select function.
2. After this the RCU shall enter TV input mode.
3. In TV input mode, the Navigation keys shall be able to navigate on-screen menus and select on-screen menu items.
4. The RCU exits TV input mode after 10 seconds when no key is pressed.
5. The RCU exits TV input mode when any key other than the menu navigation (<Menu Up>, <Menu Down>, <Menu Right>, <Menu Left>) is pressed.
6. When the <OK> key is pressed the TV OK function is send and the RCU exits TV input mode
7. If RCU is operating in RF4CE mode it should send “GetPendingMessages” once.
8. When one of the menu navigations keys (<Menu Up>, <Menu Down>, <Menu Right>, <Menu Left>) is pressed while in TV input mode and within the 10-second window, a new 10-second timer will start.

8.10 Advanced Features

Babylon project will implement advanced features based on UAPI implementation. Each of the advanced features will be detailed in the sections below.

The table below defines the RCU behavior in case of concurrent usage of the advanced features.

Started First → Started second ↓	Background OTA	Voice	Quickset
Background OTA	N/A	Background OTA is accepted but immediately paused until the end of the audio streaming and then resumed automatically by the RCU	STB should prevent OTA to be started while a QS session is ongoing. In the eventuality an OTA activity is started while a QS session is ongoing then OTA will be rejected with the error code EBUSY
Voice	Background OTA is paused until the end of the audio streaming and then resumed automatically by the RCU	N/A	Quickset will get priority if started first and audio start request from STB will be rejected with error code EBUSY
Quickset	Background OTA is paused until the end of the of the Quickset activity and then resumed automatically by the RCU	Quickset activity is rejected with EBUSY status	N/A

Table 34: Concurrent usage scenario Table

8.10.1 Quickset

QuickSet is a method of programming the remote control with a certain set of database commands used to control different Consumer Electronics devices.

Quickset Library coverage is Europe.
The flowchart below shows an example of flow for different setup methods that will be supported by the RCU implementation on the STB:

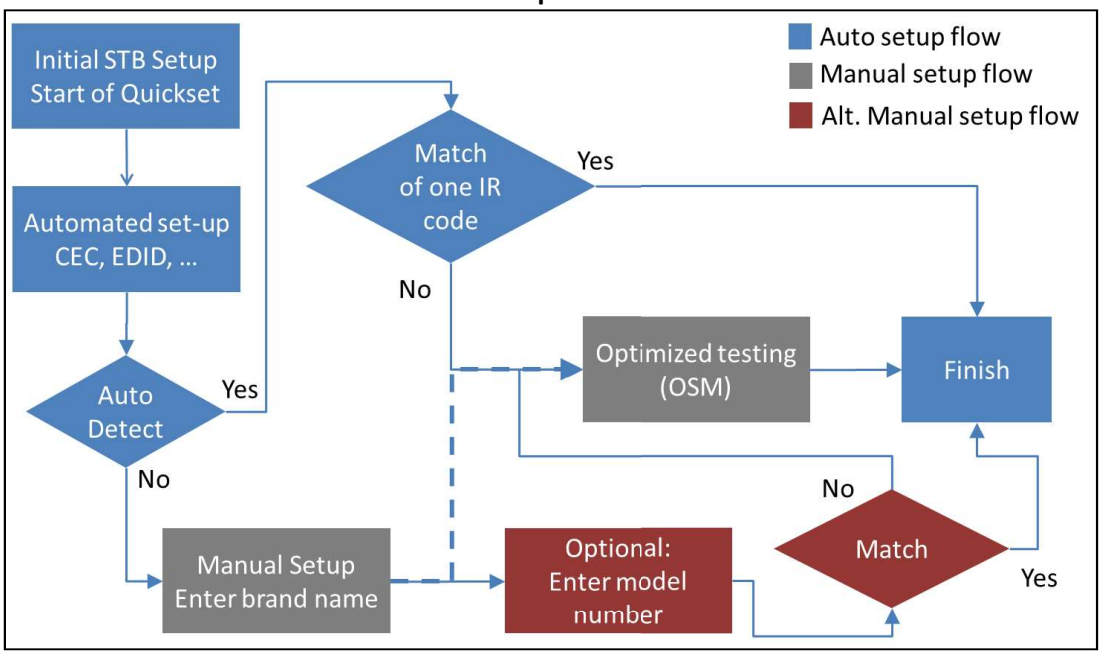


Figure 10: Quickset flowchart

8.10.1.1 Quickset implementation (AppID = 0x13)

This section is describing the Quickset implementation through App Id 0x13.

The following are descriptions of the Quickset features:

- **Key Codes Download-** This feature will download a set of IR codes to the remote in ZipIR format. These codes will be stored in NVM.

- **Code Test** – this feature allows the user to test and see whether a selected code-set will work with a specific device prior to assigning it to a mode on the remote. **Note:** This feature is not planned to be implemented on the STB.

- **Auto Setup** - The Quickset library in UAPI provides a feature of automated lookup by identifying the attached devices by means of HDMI-EDID look-up or using CEC control information. Automated lookup will allow the programming of devices that support EDID/CEC information automatically without going through the standard device-brand-model lookup route. CEC/EDID information of connected device is obtained and passed to respective API. Based on the information passed, API will retrieve code sets for the device connected which can be used for programming the remote.

- **Manual Setup** - Manual setup assists the user with a UI flow and enables to setup RCU manually. The user is instructed to search for brand/model of his device, and the programming sequence is displayed with the correct ID to the user.

- **Optimized Setup Maps – "**Optimized setup maps" assists the user with a UI flow, to setup his TV and AMP without searching for the model. After user selects brand, Quickset starts to test RCU keys, and by the user feedback finds the right device and programs it to the RCU.

  Note: Test keys for OSM are limited to Volume+, Volume- and Mute.

- **One Touch View (OTV) –** OTV feature will be executed by pressing STB Power key or TV power Key depending on the state of STB. See for more details section 8.10.1.3.

8.10.1.2 One Touch View (OTV)

The RCU will store OTV ON and OTV OFF macro and will support the One Touch View feature through the STB Power key.

OTV ON or OTV OFF macro will be transmitted depending on the current state of the STB.
It shall be possible to enable and disable One Touch View feature via STB settings menu. OTV logic shall be governed by the STB.

8.10.1.2.1 One Touch View ON

One Touch View (OTV) ON macro should power ON all the devices and should change all the inputs accordingly for devices included in the video path (TV and Audio device if present).

If Quickset is implemented through App ID 0x13, OTV ON macro will be mapped to Power ON trigger ID (0x02) in the key map record.
If RCU operate in BLE mode, then OTV ON macro will not include the RF STB power key. RF STB Power key transmission will be handled by the RCU outside the OTV macro.

Remote operation sequence on OTV ON macro:

1. RCU sends the discrete IR Input Key for TV (if discrete IR is available for this device).
2. RCU sends the discrete IR Input Key for AVR if present (if discrete IR is available for this device).
3. RCU sends Power ON Key for TV (if discrete IR is available for this device).
4. RCU sends Power ON Key for AVR (if discrete IR is available for this device).
5. RCU wait for TV boot-up delay as defined in the OTV macro.
6. Remote sends Input Key for TV {re-send IR discrete input command}.
7. Remote sends Input Key for AVR {re-send IR discrete input command}.

8.10.1.2.2 One Touch View OFF

One Touch View (OTV) OFF macro should power OFF all devices included in the Activity.

If Quickset is implemented through App ID 0x13, OTV OFF macro will be mapped to Power OFF trigger ID (0x03) in the key map record.
If RCU operate in BLE mode, then OTV OFF macro will not include the RF STB power and RF STB Power key transmission will be handled by the RCU outside the OTV macro.
In BLE mode OTV OFF macro should start by a delay if TV OFF should be delayed after RF STB Power key transmission.

Remote operation sequence example on OTV OFF macro:

1. Wait for OTV OFF macro delay before transmitting first IR command. This delay is optionally set in the macro by the STB in case TV IR command must be delayed from RF Power key.
2. Remote sends Power OFF Key for TV (if discrete IR is available for this device).
3. Remote sends Power OFF Key for AVR (if discrete IR is available for this device).

8.10.1.3 STB Power Button

The STB power key always transmits STB Power Toggle command. However, depending upon the STB state, it should activate one of the following macros:

- STB in Activate State: Transmit OTV OFF macro if programmed (Includes Discrete Power Off for TV/Amp)
- STB in Standby State: Transmit OTV ON macro if programmed (Includes Discrete Input change for TV/Amp and Discrete Power On for TV/Amp)

IR portion of both the above activities are programmed as macros onto the remote after successful Quickset programming and if both the activities are deemed “qualified”.
If the macros are not present, the RCU needs to send only the STB power toggle command.

STB Power Button activity is defined in section 8.7.15 for BLE implementation and in section 8.8.13 for RF4CE implementation.

Below are common rules that apply when STB power key is pressed:

- IR key presses shall not interrupt or influence behavior of the Power Sequence Activity. IR key presses shall be executed if the IR key is maintained or pressed after OTV macro transmission is completed.
- RF key presses shall not interrupt or influence behavior of the Power Sequence Activity. RF key presses shall be ignored during this time.
- After initiating the Power Sequence Activity, upon STB Power key press, then pressing STB power key before the end of the activity will trigger the following actions:
    - if the RCU is not BLE connected, OTV ON macro should be abort and OTV OFF should be executed.
    - if the RCU is BLE connected, any keypress on STB power key while OTV sequence is not completed should be ignored.
    - if RCU operates in RF4CE mode STB power sequence will be aborted and restarted.

8.10.1.4 TV Power Button

Any additional key should be ignored if it is pressed before the end of the TV Power key transmission.
RCU should ignore proximity detection events while executing the TV power toggle sequence.
Implementation details of TV Power Button are described in sections 8.7.15.4 and 8.8.13.3.

8.10.2 UAPI Messaging

8.10.2.1 UAPI Configuration App ID = 0x00

Parameter ID	Name	Type	R/W	Standard Opcode[1]	Length	Description
0x01	Version Info	Struct	R	Config (0x01)	4 bytes	UAPI version information (Version X.Y.Z is encoded as 0xXX, 0xYY, 0xZZZZ).
0x02	Product Info	Struct	R	Config (0x01)	4-N bytes	Product information (This is the application version XX.YY.ABCD). In order to align the manifest files across the platforms, Product Name should be set to VID.PID (06E7.8140).
0x03	Polling Period[2]	Integer	R/W	NVM (0x02)	4 bytes	Period used by app to poll for messages. Default value: 30 seconds.
0x04	HW Info	Struct	R	Config (0x01)	12 bytes	HW platform information.
0x05	Auto Poll State[3]	Boolean	R/W	NVM (0x02)	1 byte	Automatic polling enable/disable. Default value: Enabled.
0x06	Boot Loader Version Info	Struct	R	Config (0x01)	4 bytes	Boot loader version information (Version X.Y.Z is encoded as 0xXX, 0xYY, 0xZZZZ).
0x07	Max Payload	Integer	R	Config (0x01)	2 bytes	Maximum payload size in bytes of incoming and outgoing UAPI messages supported; if not present, assume **1020** byte (i.e. 1024 byte for header + OpCode + payload).
0x08	Secured	Boolean	R	Config (0x01)	1 byte	True if network used to transfer UAPI messages is secured; if not present, assume false.

0x09	MTU size	Integer	R	Config (0x01)	1 byte	MTU size in bytes; if not present assume 20 bytes.
0xF1	STB Probe Value	Integer	R/W	Config (0x01)	1 byte	Probes whether the STB is active or not. 0x01 = Active, 0x00 = Standby.
0xF2	ProfileChangeTimeout (valid in RF4CE mode only)	Integer	R/W	NVM (0x02)	1 byte	ProfileChangeTimeout is only valid in RF4CE mode. Defines the delay after pressing the side key (Profile) for the GetPendingMessages to occur. In 100ms increments. Min (0x00) = 100ms, Max (0xFF) = 25.6s. Default value is 100ms (0x00). Note: This parameter is only valid in RF4CE mode. If this Param ID is received while the RCU operates in BLE mode, then RCU should return an EINVAL error code.
0xF3	STB Boot-up Delay (valid in BLE mode only)	Integer	R/W	NVM (0x02)	1 byte	Configurable STB Boot-up delay between 0 and 255 seconds in increments of 1s. Delay is persistent to battery removal. Default Value = 0s. Note: This parameter is only valid in BLE mode. If this Param ID is received while the RCU operates in RF4CE mode, then RCU should return an EINVAL error code.
0xF4	STB Power On Tx timeout (valid in BLE mode only)	Integer	R/W	NVM (0x02)	1 byte	Configurable maximum timeout value for sending the first virtual key upon STB power key press while RCU is not connected. Timer is started upon CONNECT_IND. The

						timeout value is expressed in steps of 100ms and is persistent to battery removal. Default Value = 10s. A value set at 0 would cause RCU to send virtual keys right after the encryption succeed. Note: This parameter is only valid in BLE mode. If this Param ID is received while the RCU operates in RF4CE mode, then RCU should return an EINVAL error code.

0xF5	RCU Model ID	Integer	R	Config (0x01)	1 byte	When RCU receives this command from STB it will report the Model ID of the RCU through the same Parameter ID. Model ID will be reported as a decimal value. Model will be reported as: 0 for Apollo RCU model that are produced before X10. 1 for Apollo RCU model that are produced from X10. 2 for Babylon RCU model.
0xF6	Proximity Sensor Activation	Boolean	R/W	NVM (0x02)	1 byte	When RCU receives this command, it will enable/disable proximity sensor feature. The command should implement a Boolean argument that should be set to: 0 to disable proximity sensor. 1 to enable proximity sensor. By default, proximity sensor should be disabled.
0xF7	RCU internal flags	Integer	R/W	NVM[5] (0x02)	1 byte	RCU Internal flags command is sent by the STB. If command is indicated as a GET action in the configuration payload[4] then when RCU will respond with a "RCU internal flags" opcode message. The response that is sent by the RCU should include a 1-byte long data which defines a bitfield formed by the RCU internal flags and

						bitfield formed by the RCU internal flags and formatted as follow: Bit 0: Vrestart detected flag. Bit 1: STB WakeUp flag. Bit 2-7: Reserved. If command is indicated as a SET action in the configuration payload[4] then the RCU should reset the internal flags (Vrestart detected and STB Wake Up). The payload data that is included in the SET action should be ignored.
0xF9	Quickset AppID config (valid in RF4CE mode only)	Integer	R/W	NVM (0x02)	1 byte	Configuration of Quickset application ID. This parameter is defined to force the Quickset application ID in RF4CE configuration. By default, RF4CE Quickset is implementing AppID 0x11. Valid values are 0x11 or 0x13. Any other values will be rejected. Default value = 0x11. This configuration is optional, and STB may or may not to send it. Notes: This parameter is relevant only when RCU operates in RF4CE mode. In BLE mode RCU should issue an EINVAL error code upon receiving this command with AppID 0x11.

Table 35: UAPI App ID 0 Payload Table

[1] The standard opcodes are defined by the table below:

Opcode	Name	Description
0x00	Reserved	
0x01	Configuration	Configure application settings (RAM)
0x02	NVM Configuration	Configure parameters in NVM
0x03	State Change	Configuration change notification
0x04	Factory Reset	Reset values to factory defaults
0x05	Get Configuration (All Params)	Get all configuration parameters for an App
0x06-0x3E		Reserved for additional standard OpCodes
0x3F	Error Report	
0x40	APP_BASE	
0x41-0xFE		OpCodes defined by the particular AppID
0xFF	Reserved	

Table 36: UAPI Standard Opcodes Table

[2] The details of UAPI Polling Period are described in section 8.8.7.2.1 and in UAPI Protocol Specification .
[3] The details of UAPI Auto Poll State are described in section 8.8.7.2.2 and in UAPI Protocol Specification .
[4] The following table is detailing the configuration payload format:

Byte	Bit	Name	Description
1			
	7:6	Action	0 = Get the current value of the parameter 1 = Set the parameter to value in message 2 = Copy from … 3 = Error in operation
	5:4	Data Type	0 = Boolean 1 = Character String 2 = Byte array

			2 = Byte array 3 = Integer (1,2,4 byte; defined by Length)
	3:0	Length	0-15
2		Parameter ID	Configuration parameter ID
3-17		Value	0-15 bytes of data
…			Optional: Additional attribute value pairs

Table 37: Configuration Payload Format

[5] From X26 FW release, RCU will handle the read/write operations of RCU Internal Flags (0xF7) from/to RAM regardless the fact STB requests this parameter to be read/write from/to NVM.

8.10.2.2 UAPI Battery Service (App ID = 0x43)

ParamID	Name	Type	R/W	Length	Description
0x01	Battery Status Payload	Integer	R	4	See Table 39: Battery Payload Status Format
0x02	Battery Query Message	Integer	R	1	

Table 38: UAPI App ID 0x43 Payload Table

UAPI battery service is configured to support sending Battery Status Payload upon queries from STB or as notifications. Notifications are sent after pairing and when warning and critical threshold have been crossed.

Upon receiving a battery level read request from STB, RCU will read the current battery level, will report it but will not latch it.

Battery status payload format is defined as follow:

Byte	Bit	Name	Description
1	7	Charging Status	1 = Connected to charging base or adapter
	6:0	Status	0x00 = OK 0x01 = Warning 0x02 = Critical 0x0F = Unknown
2		Level	0 – 100 = Battery Level in %, 101 – 255 = Reserved
3-4		Voltage	Optional: Battery voltage in 1/2048 V

Table 39: Battery Payload Status Format

Percentage value is computed according of the table below:

Actual Voltage at the chip	Report sent in terms of percentage
3 V ≤ VDD	100 %
3 V < VDD ≤ 2 V	[100 * (VDD – 2)]
VDD < 2 V	0 %

Table 40: Percentage Level Reporting

8.10.2.3 Messaging for getting STB Status

When STB power key is pressed, the RCU sends a HID key. This will “probe” the STB and request a response message. The response of STB may be originated outside of UAPI, but the message will be delivered via UAPI.

When the STB receives 0x3F code, it shall respond with an UAPI message in the following format:

OpCode	Action (7:6)	Data Type (5:4)	Length (3:0)	Attribute	Value
0x01	1 (set)	0 (Boolean)	1	0xF1	0x00 (Standby)

Or

OpCode	Action (7:6)	Data Type (5:4)	Length (3:0)	Attribute	Value
0x01	1	0	1	0xF1	0x01 (Active)

OpCode 0x01 is a standard UAPI_CB OpCode for application configuration.

Attribute 0xF1 is a Product-Specific Configuration Parameter in UAPI. ParamID 0xF1 is defined here and specifically for this functions and implementation. ParamID 0xF1 conveys the status of STB. Value 0x00 means “Standby” and Value 0x01 means “Active”. If RCU is disconnected, attribute 0xF1 shall contain 0x00. The timeout value for the STB Probe is 1 second.

8.10.3 Run-time configuration

The Apollo RCU shall implement Run-Time Configuration and shall support all currently available parameters to be configured during run-time.
All parameters mentioned below can be read and/or written from/to NVM section during development without generating a new X-release.
During initialization, the application code will check if an RTC parameter is defined in NVM and, if it is, the application code will overwrite the use the RTC value instead of the default value of the parameter.
RTC should be supported over UAPI and over TSI in order to allow read/write of the RTC parameters through BLE and through UART connections.

1. RTC Format XML version major
2. RTC Format XML version minor
3. RTC Data XML version major
4. RTC Data XML version minor
5. Serial Number
6. BLE MAC address (BD address)
7. Manufacturer Name
8. Product Name
9. USB Product ID
10. USB Vendor ID
11. Product ID
12. Model Number
13. Default Event Map (keymap)
14. ZSF record
15. ZCL record
16. Biquad filter coefficients – 16kHz
17. Volume Scale – 16kHz
18. Volume Shift – 16kHz
19. Biquad filter number of stages – 16kHz
20. BLE slave latency
21. BLE Minimum Connection Interval
22. BLE Maximum Connection Interval
23. BLE Supervision timeout
24. BLE Reconnection Advertisement Interval 1
25. BLE Reconnection Duration 1
26. BLE Reconnection Advertisement Burst Duration 1

27. BLE Reconnection Advertisement Burst Gap 1
28. BLE Reconnection Advertisement Type 1
29. BLE Auto reconnect on disconnect
30. BLE Auto reconnect on bootup
31. BLE Reconnection Advertisement Interval 2
32. BLE Reconnection Duration 2
33. BLE Reconnection Advertisement Burst Duration 2
34. BLE Reconnection Advertisement Burst Gap 2
35. BLE Reconnection Advertisement Interval Manual
36. BLE Reconnection Duration Manual
37. BLE Reconnection Advertisement Burst Duration Manual
38. BLE Reconnection Advertisement Burst Gap Manual
39. BLE Reconnection Advertisement Type Manual
40. BLE Undirected Advertisement Data
41. BLE MAC index
42. BLE Initial Advertisement Data
43. Short Product name index
44. Short Product name length
45. Active unpaired triggermap
46. Active paired triggermap
47. Mode Setup record – STB
48. Mode Setup record - TV

8.10.4 Voice

The Voice feature shall be supported over UAPI and through the App Id 0x036. All messaging and data formats are defined in the UAPI specification.

The voice control will be performed through UAPI messages including a UAPI header whereas voice data will be sent as raw data.

8.10.4.1 Voice Implementation Flow

Audio flow requirements summary:

1. The RCU must be paired to the intended Host Device with support for the Voice feature.
2. The MIC key must be pressed and held down to capture Voice data.
   If MIC key is not pressed when STB requests an audio streaming to be started, then RCU should send an audio stop in response to the audio start request.
3. Audio data is transmitted in ADPCM format (16bits-16kHz-mono).
4. Any key press interrupt during a Voice session should be ignored.
5. The audio streaming should be implemented as a “Push-Talk-Release” implementation.
6. STB should be able to handle an audio stop message from RCU in each of its audio streaming state.
7. It is recommended STB to implement an audio streaming timeout reflecting the maximum allowed duration of an audio session.

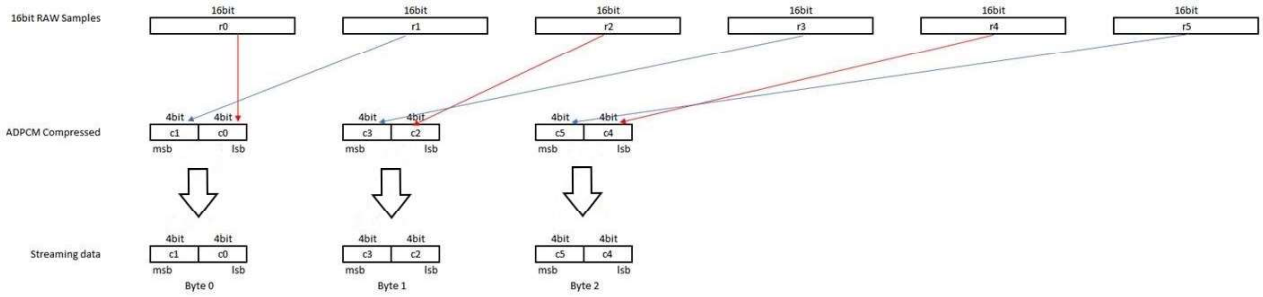
The audio streaming initiation is dependent of the radio protocol that is used to transmit the media. BLE should implement a Host initiated audio streaming whereas RF4CE should implement an RCU initiated audio streaming.
The details of the protocol controlling the audio streaming is described respectively in sections 8.7.14 for BLE and 8.8.12 for RF4CE.

8.10.4.2 Audio Data Format

Requirements:

1. 1 Channel (Mono), 16kHz, 16-bit audio, 256kbps raw compressed 4:1 to 64kbps with IMA ADPCM format.
2. Transmit as a compressed 64kbps ADPCM stream over BLE.

3. ADPCM audio buffer should be able to buffer up to ~300ms. The exact size of the buffer should be tuned according to the audio packet size with a minimal value of 280ms and a maximal value of 360ms.



Note: the ADPCM compression engine stores compressed odd samples in the Most Significant Nibble and even samples in the Least Significant Nibble. Then the nibbles are sent over the air. See the picture below for a visual representation.
The ADPCM audio samples can be decoded using sox utility with the command: sox -t ima -e ima-adpcm -c 1 -r 16k -N input.adpcm output.raw

8.10.5 Proximity Sensor

The Proximity sensor (IQS211B) should detect when the user is picking up the remote. To that extent a proximity sensor is included on the PCB as well as a trace to capture the change in capacitance when a user wraps their hand around the remote.
After the proximity sensors triggers, the virtual key for STB probe will be sent, see sections 8.7.15.1 and 8.8.13.1. As long as the user keeps the RCU in their hand, the Proximity sensor shall not trigger again. Only after the RCU is let loose, the proximity sensor can be triggered again, and a new STB probe will be sent.
The setting value that should be programmed in the proximity sensor is 08806EB6.
Proximity sensor can be disabled/enabled from UAPI command Proximity Sensor Activation. See section 8.10.2.1 for further details on sensor activation.

8.10.6 Over-the-Air (OTA) Update

The OTA feature shall be supported over UAPI.
All messages and data formats are defined in the OTA section of the UAPI Protocol Specification .
Please also refer to section 16 for further details on service desk.

8.10.6.1 OTA Update Requirements:

Background OTA will be implemented for both RF4CE and BLE.

For further information on OTA configuration parameters please refer to customer portal online documentation. See section 16 for further details.

OTA operating voltage is detailed in Table 19: Battery thresholds.

QuickSet data and RCU configuration such as operating mode, pairing information, STB boot up delay, STB Power ON Tx Timeout, polling period should be retained after OTA.
The priorities between the different features, including OTA, are defined in Table 34: Concurrent usage scenario Table.
OTA update mechanism should be fail-safe meaning if an OTA is interrupted for any reason then RCU should remain operational and should keep its current settings.

8.10.6.2 OTA Update Mechanism Overview

This section briefly describes the message exchange between the STB and the RCU and the fail-safe mechanism that is implemented in the RCU.
Complete implementation details are described on customer portal online documentation. See section 16 for further

Complete implementation details are described on customer portal online documentation. See section 10.4 for further details.

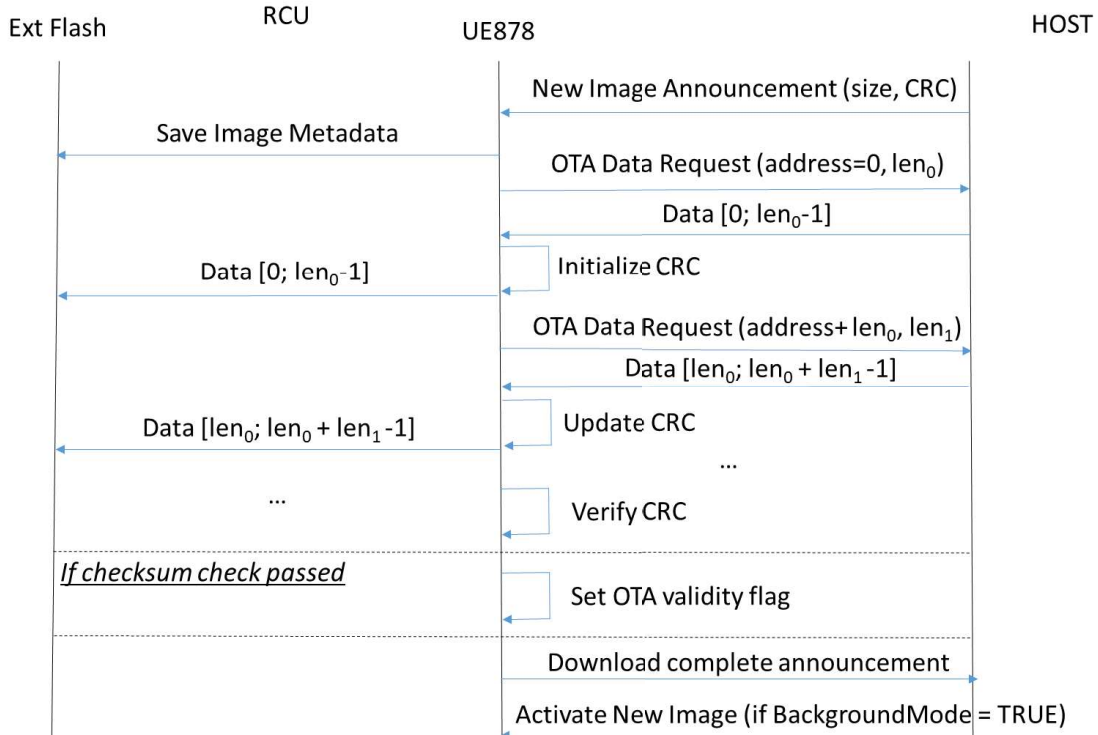
An OTA update will follow the following sequence:

1. If STB has detected it has a new firmware image to upload (by comparing the version numbers of the new image and of the current image running in the RCU), then it should send a New Image Announcement (NIA) message to the remote as soon it is possible. It can be straight forward if there is an active BLE connection between the devices or at the next Get Pending Message if RCU operate as an RF4CE device.
2. After receiving the NIA message, the RCU will use the "OTA Data Request" messages to request the download of the new OTA image through fixed size chunk of data. The new OTA image is then written into the OTA buffer.
The duration of downloading and writing the image from STB to RCU is limited by the connection settings (type of connection, connection parameters…), by the size of the file to download and by the Flash Erase & Write times of the external Flash IC.
3. After the download completion, the remote will perform CRC16 check to make sure the downloaded image is not corrupted.
4. If CRC16 check passes, an OTA validity flag is set by the firmware in the OTA buffer area. The flag indicates a new image is available.
5. Then, when STB wants to activate the new image, it sends an OTA Activate image command that will cause the RCU to reset. Note that a power on reset (POR) will also activate the new image, even when there is no explicit Activate request yet from the STB.
6. The bootloader will check for OTA validity flag and, if it is set, will start updating the application/stack/DB areas based on the address range that is included in the OTA image.
7. The bootloader and application will perform a final checksum validation of all sections.
8. If checksum fails, the remote will stop and will load application again from OTA buffer at next POR.
9. If checksum is Ok, OTA indicator flag is cleared, and bootloader will relinquish control to the RCU application.

For reference an OTA firmware update (steps 1 to 4) takes approximately 90s with UTS test suite in an environment without interference and with DLE enabled.

After the new image activation from STB, the RCU will take maximum 30s to copy the code from external to internal flash and to reboot.

The flow diagram below illustrates the OTA procedure flow.



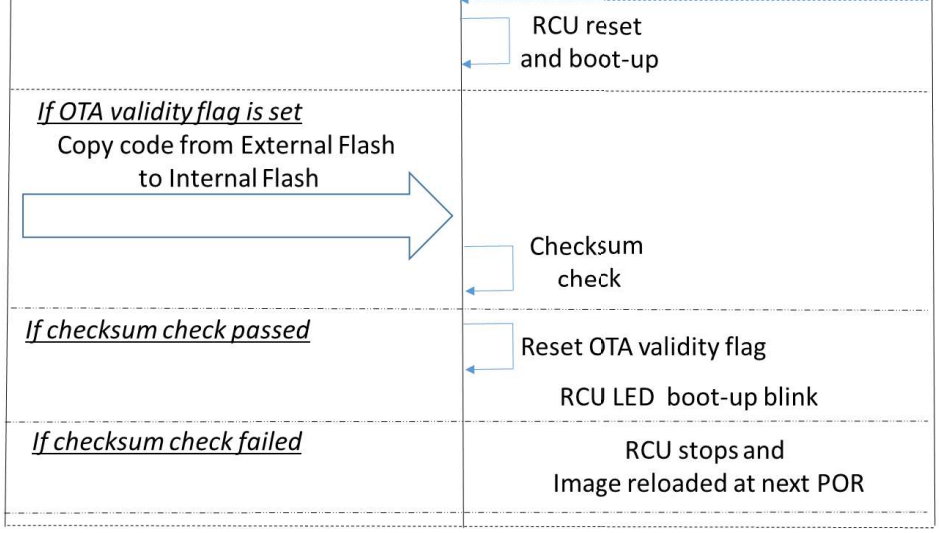


Figure 11: OTA flow diagram

8.10.6.3 OTA Update Selene STB specificities

Due to limitations of Selene set top box, compromises had to be done while performing an OTA update and specific behaviors had to be implemented in order to avoid any interoperability issues.

Consequently the following specific behaviors have been implemented when an OTA is detected to be initiated from a Selene STB:

- A 5 milliseconds delay have been implemented between OTA data requests and the previous data request acknowledge from STB.

- If a corrupted data block is received from STB, then RCU should request the same block at least twice (resulting in 3 attempts maximum) instead of just aborting the OTA session.

8.10.6.4 OTA Update Version Control Specificities

Firmware revisions before the release of X22 are implementing Adesto Flash driver and are intended to work with Adesto external Flash only.
With the introduction of GigaDevice Flash, X22 implements the detection of which external flash is sitting on the PCB in order to select the proper Flash driver.
In order to guarantee the selection of the right driver, the RCU will reject any OTA firmware update attempt to a FW versions below X22 if RCU detects a GigaDevice external Flash on the PCB. This implementation will take precedence on the usage of OTA update force flag.
In case RCU firmware rejects an update attempt due to flash driver incompatibility the RCU will reject the update with the error code 0x01 which corresponds to *"No permission error"*.

9 Customer Requirements on Testing

9.1 Test Support Interface Requirements

- After sample testing the factory will not set TSI_PERMANENTLY_DISABLED to TRUE so customer can use TSI for their internal testing.
- For further information on TSI refer to section 14 in annex.

9.2 Test firmware

9.2.1 Voice

A dedicated test firmware should be delivered and should be able to send a fixed number of samples in the conditions of an audio test.

The purpose of the test firmware is to allow LGI to reproduce a stable test environment for testing radio or coexistence issues.
So, it is required the audio packets are sent exactly as production firmware would do in the same environment (same bitrate, packet interval, retransmissions...).

It is required the test firmware should support RF4CE and BLE with the same RF performances and throughput than the production firmware.
RCU should support DLE and non DLE enabled STBs.

The test firmware should be based on the latest available software release and any fix impacting the RCU behavior or RCU audio performances should be ported on the test firmware.

The test firmware should reproduce the production firmware behavior with the only following differences:

- Upon pressing the MIC key, voice streaming should start for exactly 10 seconds which will result in 80.000 bytes of a known pattern data.
- Upon releasing the MIC key, the voice streaming should not be stopped if the 10 seconds of data have not been sent.
- The data should be not come from ADC but a known data pattern should automatically be generated within the firmware and will be implemented by the following code snippet:

```
void AudioTestPattern_Generate(uint8_t* buff, uint16_t length)
{
    uint8_t upper, lower, number;
    uint16_t i = 0;

    for (upper = 'A'; upper <= 'Z'; upper++)
        for (lower = 'a'; lower <= 'z'; lower++)
            for (number = '0'; number <= '9'; number++)
            {
                if (i >= length)
                {
                   return;
                }
                buff[i++] = upper;

                if (i >= length)
                {
                   return;
                }
                buff[i++] = lower;

                if (i >= length)
                {
                   return;
```

```
                }
                buff[i++] = number;
            }
}
```

The audio session start should follow the same mechanism than is implemented in the production firmware.
However, audio stop requests from STB should be ignored so the tester does not have to maintain the key pressed during the test.
The audio streaming should only stop after the expected amount of data has been transmitted and RCU should issue an UAPI Audio Stop message while stopping the streaming session.

The audio data should be sent as raw data within HID reports with report ID 31.

10 Sustainable Babylon Specifics

10.1 Overview

This section will describe all the specificities of the derivative product called Sustainable Babylon product which is defined as an eco responsible version of the Babylon product made of recycled plastic and delivered in sustainable packaging.

Sustainable Babylon products have been setup with the product family number R328453 versus R328451 for Babylon base model giving the following table:

Variant	Part Number	Color Keys?
Ziggo	R328453B99-00001	No
UPC Slovakia	R328453B99-00003	No
Sunrise	R328453B97-00001	No
Virgin Media Ireland	R328453B99-00002	Yes

10.2 Housing Material and Finish Details

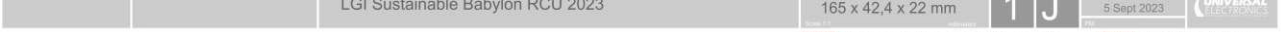
Item #	Part	Material	Colour	Texture / Finish
1	Top case	95% PCR ABS	NCS S 9000-N	Matte/ YS 20043-B Gloss/SPI A-2 for upper part
2	Bottom case	95% PCR ABS IR Transparent ABS	NCS S 9000-N	Matte/MT-11520 Gloss/SPI A-2 (side band

		IR Transparent ABS for IR window		Gloss/SPI A-2 (side band and IR window)
3	Battery Door	95% PCR ABS	NCS S 9000-N	Matte/MT-11520
4	Logo plate	95% PCR ABS	NCS S 9000-N	Matte/ YS 20043-B
5	Light Guide	PC+RTP pigment	Milky White	MT-11007 outside SPI A-2 inside
6	Key Caps	95% PCR ABS	NCS S 9000-N	Matte/ YS 20043-B
7	Home, STB POWER, TV POWER and Voice Key Caps	95% PCR ABS	NCS S 9000-N	Gloss/SPI A-2
8	Side Key Cap	95% PCR ABS	NCS S 9000-N	Matte/ YS 20043-B
9	OK key ring	ABS inside PC outside	NCS S 9000-N Chrome plated	Matte/ YS 20043-B Gloss

10.3 CMFs

Here are CMFs related to sustainable version of Babylon detailing the materials but also the printing location updates made to the product.

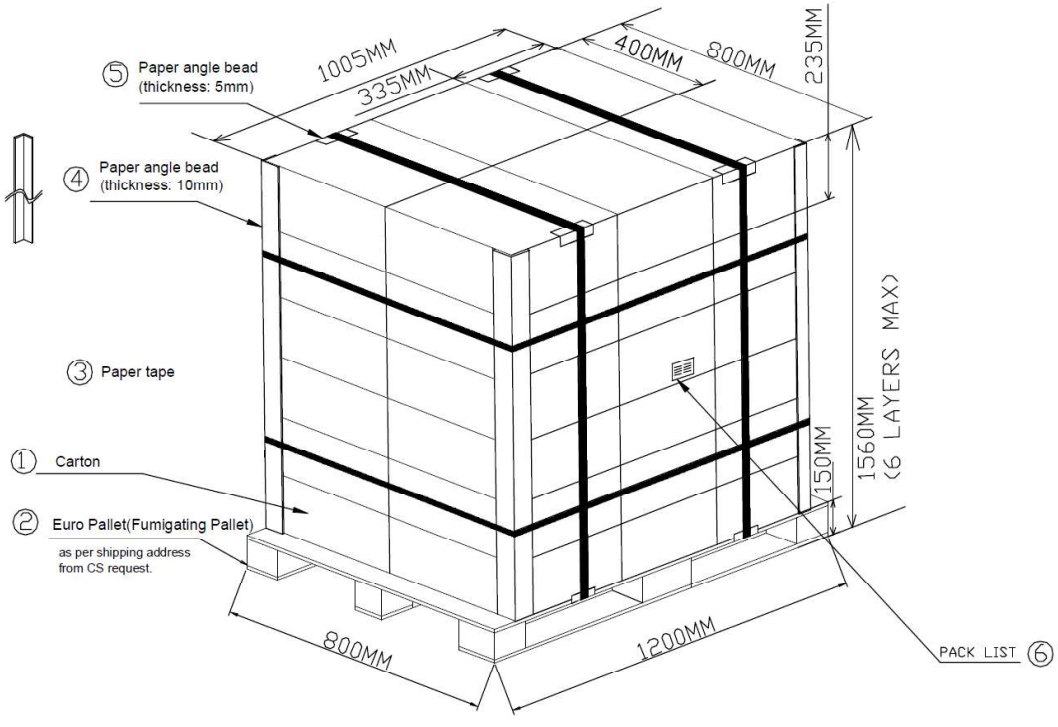

PANTONE
Topcase - No Paint
NCS S 9000-N
95% PCR ABS
Bottomcase - No Paint
R Translucent ABS
Batterydoor - No Paint
Hard key caps - All keys
Double shot key
Side-key cap - Includes internal rubber barrier
Lightguide
Milky White
Opaque white
Logoplate - Printed logo
Keypad printing - Pad printing
Pure White
Pad printing
Logoplate printing - Pad printing - Brand related
2x AAA Batteries
All keys metaldomes
1x Red LED
1 x White LED
NOTE: Colors on rendering are for reference only. Please refer to Pantone color book for true colors.
Microphone
OK
R328453B9X-000XX
Y23xx00
Universal Electronics BV
Colosseum 2 -7521 PT Enschede
The Netherlands
Laser parameters 3mm dot:
55W, 800mm/s, 20 HZ / ConfigX
FCC ID & Made in China
FCC ID code & Made in China
for FFC requirred countries


LGI Sustainable Babylon RCU 2023
165 x 42,4 x 22 mm
1
J
5 Sept 2023
CONCEPTUAL
CONFIDENTIAL


PANTONE
Topcase - No Paint
NCS S 9000-N
95% PCR ABS
Bottomcase - No Paint
R Translucent ABS
Batterydoor - No Paint
Hard key caps - All keys
Double shot key
Side-key cap - Includes internal rubber barrier
Lightguide
Milky White
Opaque white
Logoplate - Printed logo
Keypad printing - Pad printing
Pure White
Pad printing
Yellow PMS 7404U
Blue PMS 2995U
Logoplate printing - Pad printing - Brand related
2x AAA Batteries
All keys metaldomes
1x Red LED
1 x White LED
NOTE: Colors on rendering are for reference only. Please refer to Pantone color book for true colors.
Microphone
OK
R328453B9X-000XX
Y23xx00
Universal Electronics BV
Colosseum 2 -7521 PT Enschede
The Netherlands
Laser parameters 3mm dot:
55W, 800mm/s, 20 HZ / ConfigX
FCC ID & Made in China
FCC ID code & Made in China
for FFC requirred countries

10.4 Packaging

Sustainable Babylon product will be delivered in paper bag versus standard polybag solution used for Babylon base product.
Palletization will also avoid usage of plastic materials and use paper base tape and paper angle bead as illustrated below:


⑤ Paper angle bead
(thickness: 5mm)
④ Paper angle bead
(thickness: 10mm)
③ Paper tape
① Carton
② Euro Pallet(Fumigating Pallet)
as per shipping address
from CS request.
1005MM
335MM
400MM
800MM
235MM
1560MM
(6 LAYERS MAX)
150MM
800MM
1200MM
PACK LIST ⑥

11 Appendix A – Abbreviations and Acronyms

ABS	Acrylonitrile Butadiene Styrene
AMP	Amplifier
AV	Audio-Video
CRS	Customer Reference Sheet
FTM	Factory Test Mode
IR	Infrared
IRDB	Infrared Database
LED	Light Emitting Diode
MAC	Media Access Control
MSO	Multiple System Operator
OTA	Over the Air Update
OTV	One-Touch-View; QuickSet function
PCB	Printed Circuit Board
PCBA	Printed Circuit Board Assembly
PHY	Physical
PMMA	Poly(methyl methacrylate)
QS	QuickSet
RCU	Remote Control Unit
RF	Radio Frequency
RF4CE	Radio Frequency for Consumer Electronics
RGB	Red Green Blue
STB	Set-top box
TLA	Three Letter Abbreviation

TPE	Thermoplastic Elastomer
UAPI	Universal API
UAPI_CB	Universal API Communication Bridge

12 Appendix B – CMF Details

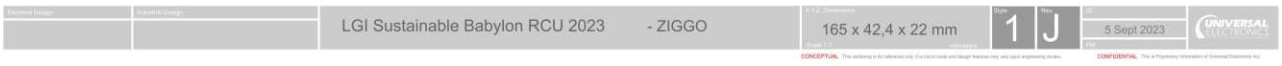
12.1 CMF derivatives

Here are few derivate CMFs files illustrating different logo plates.
Note that the following pictures are provided for reference only and that the complete and final set of derivative models are treated separately in their respective CRS files.

12.1.1 UPC model


Logoplate printing - Pad printing - UPC P
Gray PMS 6C
Pad printing
Matte
2x AAA Batteries
All keys metaldomes
1x Red LED
1 x White LED
NOTE: Colors on rendering are for reference only. Please refer to Pantone color book for true colors.
LGI Sustainable Babylon RCU 2023
- UPC
165 x 42,4 x 22 mm
1
J
5 Sept 2023

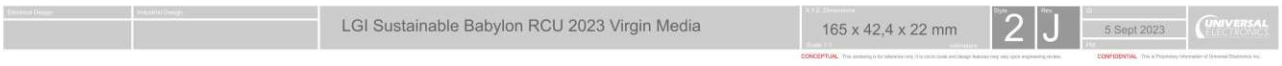
12.1.2 Ziggo Model


Electrical Design
Industrial Design
LGI Sustainable Babylon RCU 2023
- ZIGGO
X Y Z Dimensions
165 x 42,4 x 22 mm
Sheet
1
Rev
J
ID
5 Sept 2023
UNIVERSAL ELECTRONICS
CONCEPTUAL
CONFIDENTIAL

12.1.3 Sunrise Model


Electrical Design
Industrial Design
LGI Sustainable Babylon RCU 2023
- SUNRISE
X Y Z Dimensions
165 x 42,4 x 22 mm
Scale 1:1
millimeters
Model
1
Rev
J
ED
5 Sept 2023
PM
CONCEPTUAL The rendering is for reference only. It is not to scale and design features may vary upon engineering review.
CONFIDENTIAL This is Proprietary Information of Universal Electronics Inc.

12.1.4 Virgin Media CMF


LGI Sustainable Babylon RCU 2023 Virgin Media
165 x 42,4 x 22 mm
2
J
5 Sept 2023

13 Appendix C – OTA Manifest

Manifest file is generated by OTA packaging tool with Product Option selected for Apollo project.
It should contain the information of the following skeleton:

```
<?xml version="1.0" encoding="UTF-8"?>
<RcuOtaManifestFile>
     <!-- manifest file version e.g 1.2 -->
     <version>1.2</version>

      <image type="application">
          <!-- firmware version in the format major.minor.revision -->
         <version>major.minor.revision</version>

         <!-- firmware file name -->
         <location_url>filename.bin.enc</location_url>

         <!-- firmware file size -->
         <size>size_in_bytes</size>

         <!-- firmware file FCS16 checksum -->
         <checksum>checksum</checksum>

         <!--force update flag -->
         <force_update>force_update_flag</force_update>

      </image>
```

```
        <target-version>
            <!-- product id -->
            <target_id>product_id</target_id>

            <!-- hw revision -->
            <hw_revision>hw_rev</hw_revision>
        </target-version>

</RcuOtaManifestFile>
```

With the following values:

<version>**major.minor.revision**</version>
major: Major release version. This is the decimal value converted from the hexadecimal value set in major release version of OTA packaging tool.
minor: Minor release version. This is the decimal value converted from the hexadecimal value set in minor release version of OTA packaging tool.
revision: Correspond to the SW product number defined in section **Error! Reference source not found.**. It is a fixed decimal value set to 7060 for Babylon green project.
Revision value is translated from the Product ID of the OTA packaging tool.
Hexadecimal value of the Product ID is converted in decimal and then the bytes are reversed.

For example:
Product ID is set to 941B representing 7060

<location_url>**filename.bin.enc**</location_url>
Filename: filename of the FW encrypted file

<size>**size_in_bytes**</size>
size_in_bytes: size of the encrypted OTA file including file header

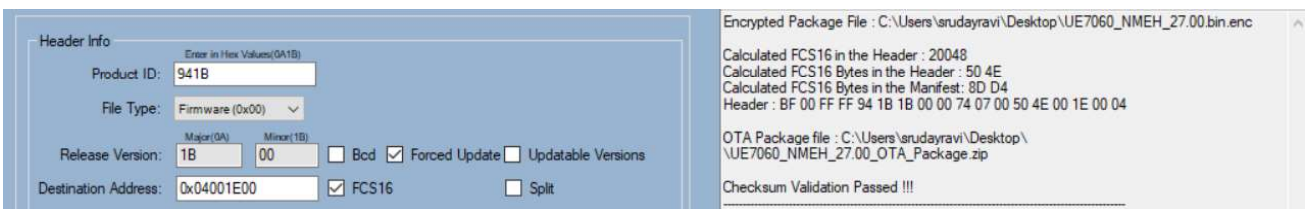
<checksum>**checksum**</checksum>
Checksum: FCS16 checksum computed on the final encrypted OTA file.
Encrypted OTA file header should include the CRC value in the header of the file. CRC value in the header should be set to the FCS16 value of the OTA file computed with CRC value set to 0xFF.

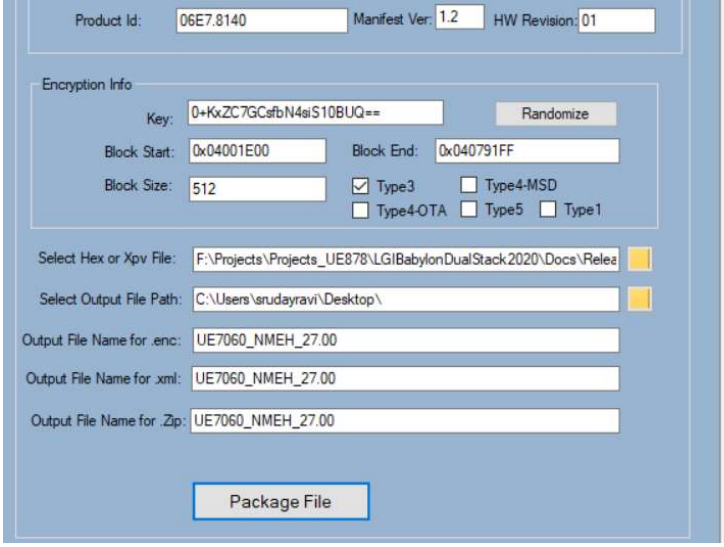
<target_id>**product_id**</target_id>
product_id: should be set to Vid.Pid which is 06E7.8140 for Apollo project

<hw_revision>**hw_rev**</hw_revision>
hw_rev: hardware revision preceded by '0' and reflecting the hardware revision of the product as defined in 8.7.2 – ex: 01 for HW revision 1.

<force_update>**force_update_flag**</force_update>
force_update_flag: Should be set to 1 to ignore current firmware revision in the RCU and to force the OTA update of the file. Otherwise, 0.

Below is an example of how to configure the OTA Packaging tool and the resulting manifest file:






```
<?xml version="1.0" encoding="UTF-8"?>
<RcuOtaManifestFile>
     <!-- manifest file version e.g 1.1 -->
     <version>1.2</version>

      <image type="application">
          <!-- firmware version in the format major.minor.revision -->
         <version>27.00.7060</version>

         <!-- firmware file name -->
         <location_url>UE7060_NMEH_27.00.bin.enc</location_url>
```

```
        <!-- firmware file size -->
        <size>506592</size>

        <!-- firmware file FCS16 checksum -->
        <checksum>0x5E31</checksum>

        <!--force update flag -->
        <force_update>0</force_update>
    </image>

    <target-version>
        <!-- product id -->
        <target_id>06E7.8140</target_id>

        <!-- hw revision -->
        <hw_revision>01</hw_revision>
    </target-version>

</RcuOtaManifestFile>
```

14 Appendix E – TSI

14.1 TSI specification & White paper

UEI TSI WhiteF

14.2 Apollo project TSI specificities

As a TSI specificity, Apollo project will include the proximity detection testing. The proximity detection testing will be done by TSI and more specifically using TSI Platform Specific Commands and Events.
For further details on proximity detection testing please refer to section 5.4.2.

Moreover, it has been asked, in the context of Apollo project, to be able to put in place a large-scale test setups that would automate sequences of key presses.
To achieve that, the suggested approach is to use TSI key inject test command that allows to request the RCU to transmit a key from a laptop executing a script.

14.3 How to connect RCU TSI port

The recommended equipment that is necessary to connect the RCU is:

- A variable power supply
- An FTDI TTL-232R-3V3 USB-Serial TTL Cable

- A PC

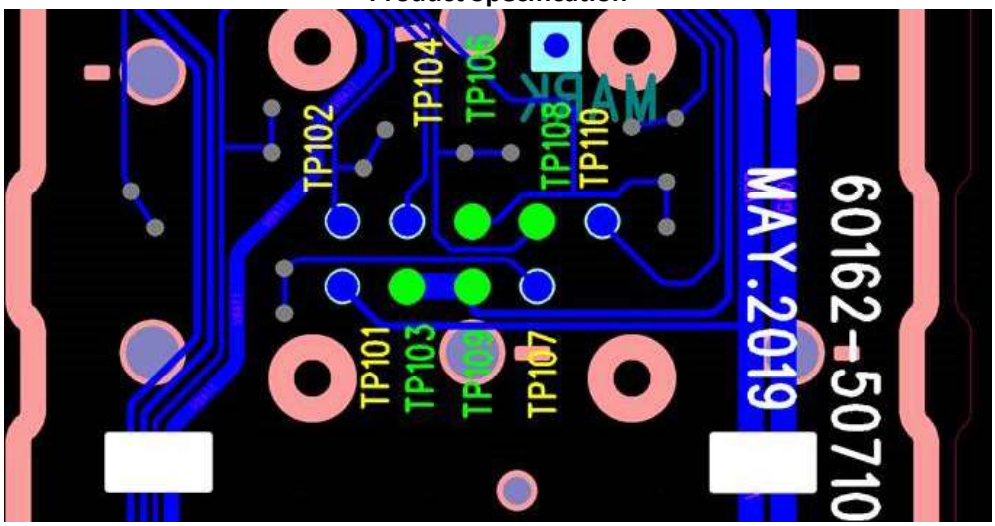
RCU is connected through its programming pads located in the battery compartment.
TSI UART interface uses 3 signals:

- UART RX
- UART TX
- GND

And with the following settings:

- 8 Bits
- 1 Stop Bit
- 115200 Baud
- No Hardware flow control

The signals can be accessed through TP106 (RX relative to DUT), TP108 (TX relative to DUT), and TP103 or TP109 (GND).
See below the PCB layout description of the pads and TPs:



14.4 TSI Key Inject Test

Below is an example how to proceed a simple TSI Key Inject Test:

- Set the power supply voltage level to 3.0 VDC
- Connect power through the DMM
- Wait until the LED blinks indicating boot completion.
- Using the TSI Host script:
    - Initialize the UUT serial port connected to the USB-to-Serial cable
    - Enable TSI on the UUT by sending the “Enable TSI” command (0x02 10 05 01 54 45 53 54 03). The UUT should respond with (0x02 10 06 02 01 ** ** ** ** 03).

- Send the Keypress Inject command (0x02 20 02 00 xx 03). Where "xx" represents the desired key code to be pressed.
  The key code should be set in hexadecimal and its value represents the coordinates of the key on the key matrix. The scan code table is defining the key code value that should be set and is detailed in 5.4.1.
  The UUT should respond with (0x02 00 01 00 03).
- Send the Key release Inject command (0x02 20 02 01 xx 03). Where "xx" represents the desired key code to be pressed. The UUT should respond with (0x02 00 01 00 03).

15 Appendix F – Key Press Actions

15.1 BLE Key Press Action Table

The following table illustrates how the remote-control behaves upon the different types of keypresses according to the its current state:

RCU state	RF key	IR key	OTV key	MIC key
Disconnected paired	Start reconnection	IR key sent	Start OTV sequence[1]	Start reconnection
Disconnected not paired	Start pairing	IR key sent	Start Pairing	Start pairing
Connected paired and idle	RF key transmitted	IR key sent	Start OTV sequence[1]	MIC key transmitted
Paired + Indirect advertising (wake-up)	Ignored	IR key sent	Abort Indirect Advertising Start OTV sequence[1]	Ignored
Not paired + Indirect advertising	Ignored	IR key sent	Ignored	Ignored
Direct advertising	Ignored	IR key sent	Abort Direct Advertising Start OTV sequence[1]	Ignored
Connected + Blasting OTV ON macro / Connected + Blasting OTV OFF macro	Ignored	Ignored[2]	Ignored	Ignored

Disconnected + Blasting OTV ON macro	Ignored	Ignored[2]	OTV ON aborted	Ignored
			OTV OFF transmitted	
Streaming Audio	Ignored	Ignored	Ignored	Streaming Audio
Downloading OTA	RF key sent	OTA paused	Start OTV sequence[1] (OTV OFF sequence)	MIC key transmitted
		IR key sent		OTA paused upon STB starting the audio streaming
Quickset download	RF key sent	IR key sent if programmed	Start OTV sequence[1] (OTV OFF sequence)	MIC key transmitted but audio request rejected

[1] refer to section 8.7.15.2 and 8.8.13.2 for complete sequence description
[2] IR key is ignored during macro transmission but will be transmitted if it is pressed/maintained after the macro transmission

15.2 Multiple Key Press Scenarios

This section describes the remote-control behavior upon keypresses according to the keypad state.
Remote-control behavior is based on the user actions on the keypad and described as a state transition table.

Here is the notation that has been used to elaborate the table and for the state transition description:

{} - no key is pressed
{A} - key A is pressed
{A, B} - key A and B are pressed
DPAD(A) – true if and only if A is in {Cursor Up, Cursor Down, Cursor Left, Cursor Right, Select/OK}
IR(A) - true if and only if A is an IR key

RF(A) - true if and only if A is an RF key
Press A - transmit a key down event, optionally followed by key repeat events over RF if RF(A), and start transmitting IR if IR(A)
Release A - transmit a key up event if RF(A), and stop transmitting IR for IR(A)
Press A, B - Press A and Press B in an undefined order
Release A, B - Release A and Release B in an undefined order

The following table illustrates the state transitions for multiple key press scenarios when RCU operates in BLE mode.

From	To	Condition	Action
{}	{A}	TRUE	Press A
{A}	{}	TRUE	Release A
{A}	{A, B}	(DPAD(A) ^ DPAD(B)) ^ IR(A) ^ IR(B)	-
		(DPAD(A) ^ DPAD(B)) ^ ¬ (IR(A) ^ IR(B))	Press B
		¬(DPAD(A) ^ DPAD(B)) ^ IR(A) ^ IR(B)	Release A
		¬(DPAD(A) ^ DPAD(B)) ^ ¬ (IR(A) ^ IR(B))	Press B
{A, B}	{}	IR(A) ^ IR(B)	Release A if pressed
		¬ (IR(A) ^ IR(B))	Release A, B
{A, B}	{A}	IR(A) ^ IR(B)	Press A if not already pressed
		¬ (IR(A) ^ IR(B))	Release B
{A, B}	{A, B, C}	IR(A) ^ IR(B)	Release A if pressed
		¬ (IR(A) ^ IR(B))	Release A, B
		(DPAD(A) ^ DPAD(B) ^ DPAD(C)) ^ IR(A) ^	

		IR(B) ^ IR(C)	-
{A, B, C}	{}	TRUE	-
{A, B, C}	{A}	TRUE	Press A
{A, B, C}	{A, B}	IR(A) ^ IR(B)	-
		¬ (IR(A) ^ IR(B))	-

Table 41: Multiple Keypress Scenarios in BLE mode

The following table illustrates the state transitions for multiple key press scenarios when RCU operates in RF4CE mode. Note in this case IR and RF keys are treated similarly.

From	To	Condition	Action
{}	{A}	TRUE	Press A
{A}	{}	TRUE	Release A
{A}	{A, B}	(DPAD(A) ^ DPAD(B)) ^ IR(A) ^ IR(B)	-
		(DPAD(A) ^ DPAD(B)) ^ ¬ (IR(A) ^ IR(B))	-
		¬(DPAD(A) ^ DPAD(B)) ^ IR(A) ^ IR(B)	Release A
		¬(DPAD(A) ^ DPAD(B)) ^ ¬ (IR(A) ^ IR(B))	Release A
{A, B}	{}	IR(A) ^ IR(B)	-
		¬ (IR(A) ^ IR(B))	-
{A, B}	{A}	IR(A) ^ IR(B)	Press A
		¬ (IR(A) ^ IR(B))	Press A
{A, B}	{A, B, C}	IR(A) ^ IR(B)	-
		¬ (IR(A) ^ IR(B))	-

		(DPAD(A) ^ DPAD(B) ^ DPAD(C)) ^ IR(A) ^ IR(B) ^ IR(C)	-
{A, B, C}	{}	TRUE	-
{A, B, C}	{A}	TRUE	Press A
{A, B, C}	{A, B}	IR(A) ^ IR(B)	-
		¬ (IR(A) ^ IR(B))	-

Table 42:Multiple Keypress Scearios in RF4CE mode

16 Service desk links

UEI Service desk is a web-based customer portal that includes programming manuals and documents for our Quickset SDK and UAPI library.
A user can easily find the documents related to a specific content by using the search bar on the main page of the desk. Search bar is accessible by clicking on the lens on the top right corner of the page.
Access to service desk can be requested to UEI team in charge of the project (FAE/PM) by providing a name and mail address list.

Here are some useful links that can be accessed through a browser:

- Service desk home page: https://universal.atlassian.net/servicedesk/customer/portal/22
- UAPI library programming manual : https://universal.atlassian.net/servicedesk/customer/portal/22/article/125153432
- UAPI main page: https://universal.atlassian.net/servicedesk/customer/portal/22/article/121474589
- UAPI OTA update : https://universal.atlassian.net/servicedesk/customer/portal/22/article/122747104
- UAPI Audio: https://universal.atlassian.net/servicedesk/customer/portal/22/article/122747812
- UAPI Quickset: https://universal.atlassian.net/servicedesk/customer/portal/22/article/121553880

17 Other annexes

17.1 FIPS SP 800-22 test reports

Below are test reports of UE878 testing related to random number generation BLE specification requirements.


CtrDRBG_19May2020.zip


HashDRBG_18May2020.zip

17.2 Hardware Configurations

HWconfig_v3.txt