RSL10 SDK Release Notes - 2.1

July 13, 2018

RSL10 is a Bluetooth® 5 certified multi-protocol radio System-on-Chip (SoC) that brings the industry's lowest power Bluetooth low energy technology to wireless applications. The RSL10 Software Development Kit (SDK) includes the firmware, software, and documentation needed to evaluate this chip and develop applications based on RSL10.

This document lists new features in this release, describes known issues, and points you to more information so you can start using the product.

What's New

Summary	Release Notes
Added Bluetooth low energy abstraction sample code	Added two new applications: *ble_central_client_bond* and *ble_peripheral_server_bond*. These sample projects are enhanced versions of existing sample code. They are equipped with a Bluetooth low energy abstraction layer, which provides a higher level application programming interface (API) to abstract the Bluetooth GAP and GATT layers. The abstraction layer has been designed to improve flexibility and simplicity by providing the following features: • An event subscription mechanism that allows the application to subscribe and receive callback notifications for any Kernel or Bluetooth event. This improves encapsulation/integration of different modules, as they can be implemented when isolated in their own files. • Generic definition of custom services with callback notification support. • Security handling and bond list implementation in RSL10 flash. • Code structure and API naming aligned with RivieraWaves documentation, so you can map the code into the documentation more easily. • Sample code refactored to separate the generic abstraction layer code and application-specific code in different files. This increases flexibility, maintainability and reusability of components between distinct applications. For more information about these sample applications, see the *RSL10 Sample Code User's Guide* and the readme text files that accompany each sample.
Ability to disable the automatic addition of flags in advertising	The GAP layer automatically inserts flags in every advertisement. Now the automatic insertion of flags can be disabled. This must be disabled for Bluetooth low energy Mesh. To disable the insertion of flags, add the following line to the `BLE_Initialize()` function *after* `BLE_InitNoTL(0);` : `BLE_ADV_Flags_Set(1);`
Using peripheral bus (PBUS) for radio initialization	The boot time of Bluetooth low energy initialization has been decreased, which helps with applications to be used in energy harvesting.
Integrated new CEVA host layer in the stack	The new host layer provides the following benefits: • The preferred slave latency can be set from an application without needing to inform the master device. • GAP APIs can be used to start DTM from the application. Do not send DTM HCI commands to the stack anymore to start DTM; use the GAP APIs instead. • Bluetooth low energy light stack can support 2 Mbps.
Bluetooth low energy stack API document update	*RW BLE Audio Mode 0 - Hearing Aid Service (HAS) Interface Specification* (*RW-BLE-PRF-AM0-HAS-IS.pdf*) from CEVA has changed since the last release. The `ntf_cfg` field of `AM0_NTF_CFG_UPDATE_IND` is documented.
ON Semiconductor company identifier is used in the Bluetooth low energy stack	The stack is now clearly identified with the ON Semiconductor company ID. Unfortunately, some protocol analyzers might be unable to decode it, and it might be shown as an unknown company ID.

6 issues

Bug Fixes

Summary	Release Notes
Defragmentation bug no longer causes stack to hang	A bug in the handling of defragmentation of L2CAP caused the stack to hang if the peer device sent an L2CAP packet for the attribute protocol but with a zero payload. This bug has been resolved.
Spurious emissions reduced to an acceptable level	Spurious emissions have been reduced to an acceptable level by modifying the remote microphone protocol.
Removed assert information from HCI variant stack	The HCI variant of the stack supports four connections and now matches with the normal variant.
Sys_RFFE_SetTXPower can be called in any order in custom protocol applications	The order of calling `Sys_RFFE_SetTXPower` used to matter in custom protocol applications because it was reprogramming a register. Now that it does not change the register value, it can be called from any location in your code.
Reduced processing time for device parameters that are called often	New APIs (functions) are available to change the following parameters, and the previous method that was used in `Device_Param_Prepare()` is no longer valid: • `adv_ifs` use `BLE_DeviceParam_Set_ADV_IFS(desired value);` • radio clock accuracy uses `BLE_DeviceParam_Set_ClockAccuracy(RADIO_CLOCK_ACCURACY);` • forced clock accuracy uses `BLE_DeviceParam_Set_ForcedClockAccuracy(desired value);` By default, `adv_ifs` is the value that the stack decides based on advertisement modes, clock accuracy is 20 ppm, and forced clock accuracy is zero (disabled).
Increased the number of devices in white list	The maximum number of devices in the white list is 10; for the light stack it is 3. The maximum number of devices in the resolving address list is six (three for light stack); however, if the core clock is 8 MHz, then only three devices can be added to the resolving list. To have any number bigger than 3 and less than 7 (4, 5, and 6), the core clock must be at least 16 MHz. This feature can be used by calling `BLE_DeviceParam_Set_MaxNumRAL(desired value);` to set `maxNumRAL`. The default is 3.
SDK launches with Java SE 10	RSL10 SDK 2.1 supports versions of Java 8, 9 and 10, all using 64-bit JVMs. This applies to 64-bit Windows operating systems only.
Adjust the Sys_RFFE_SetTXPower function for uncalibrated boards	The `Sys_RFFE_SetTXPower` function works on uncalibrated boards using the nominal trim values. For optimum operations, we recommend using calibrated boards.
Pairing works when controller privacy is used	When controller privacy is used, the address identity type is now set correctly to enable successful pairing with devices that are using private resolvable addresses.
CEVA GAP documentation updated	The DTM API as described in the GAP interface specification documentation (*RW-BLE-GAP-IS_2mbps.pdf*) is available in this release. Also, the document has been updated since the RSL10 SDK 2.0 release.

10 issues

Migrating from RSL10 SDK 2.0 to 2.1

Some of the changes to your Release 2.0 code or projects to work with them in Release 2.1 are:

- In Bluetooth low energy applications, remove the following lines from the *Device_Param_Prepare* function:

```
param->clockAccuracy = RADIO_CLOCK_ACCURACY;
param->adv_ifs = 0;
```

Instead, add this function to adjust the clock accuracy for the stack in the peripheral application in the *BLE_Initialize* function:

```
/* Set radio clock accuracy in ppm */
BLE_DeviceParam_Set_ClockAccuracy(RADIO_CLOCK_ACCURACY);
```

- In the *peripheral_server_PRA* application, we changed the `RENEW_DUR` parameter from 15000 to 150 to reflect the change to the renew

duration granularity in the Bluetooth stack, which was 10ms and is now 1s.
- Compatibility of the Remote Microphone protocol is not guaranteed between SDK 2.0 and SDK 2.1. In other words, an application compiled as TX in SDK 2.0 will not necessarily work with an RX application compiled in SDK 2.1.

Important
This is not an exhaustive list of the necessary changes to migrate from SDK 2.0 to SDK 2.1.

Known Issues

The following issues are known to be outstanding in release 2.1

Summary	Release Notes

Sys_ASRC_Enable function not functioning	The `Sys_ASRC_Enable` function does not function as intended. Use these methods instead:
	`// Enable ASRC`
	`ASRC_CTRL->ASRC_ENABLE_ALIAS = ASRC_ENABLE_BITBAND;`
	`// Disable ASRC`
	`ASRC_CTRL->ASRC_DISABLE_ALIAS = ASRC_DISABLE_BITBAND;`

Remote Mic TX with Audio Shield does not work after Supplemental Calib app	To make the *remote_mic_tx_raw* application to work with the Audio Shield, erase the NVR3 program that is loaded by the supplemental calibrate application.
CRC limitation with 89 vector table entries	The CRC function is not valid if all 88 possible vectors are in use.
Controller privacy cannot be used with some smartphones	Because of a hardware issue, smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used. Therefore, use host privacy in these cases. When host privacy is used, you can decrease power consumption close to the level of controller privacy use by employing a new feature called "power save control". To enable it, see the new API called `GAPM_POWER_SAVE_CTRL_CMD` as specified in the GAP Interface Specification from CEVA.
XTAL 48MHz start up time needs to be increased	When the RC oscillator is used as the source of the low power clock, the Bluetooth low energy connection with the sleep application is not very stable and sometimes a disconnection is observed. To alleviate the issue and also account for board to board variation, the wake-up delay for this case has been increased from 1.1 ms to 1.3 ms by calling `BLE_LLD_Sleep_Params_Set` with the right parameter.
Only pairing and bonding have been implemented in the "ble" driver	For the *ble_peripheral_server_bond* sample application, its “ble” driver can support pairing with a central device to set up an encrypted connection, and bonding by storing that connection information so that the link’s encryption information can be re-used later on. However, other features of this driver (e.g., white-listing, private resolvable addressing, scanning, etc.) have not been fully implemented and tested. They will be available in future releases.

Delay needed in the remote_mic_rx_raw sample application for PCM input	To have good audio quality for PCM input, add a delay in the `ASRC_out` DMA interrupt. Use a delay instruction in the `DMA_IRQ_FUNC(ASRC_OUT_IDX)(void)` function in *app_func.c* as shown below:
	`#if (INPUT_INTRF == PCM_RX_RAW_INPUT)`
	`  Sys_Delay_ProgramROM(100);`
	`#endif`

Remote Mic TX: smaller sub-frame causes noisy right channel	Choosing a SUBFRAME that is less than 32 samples for the DMIC and PCM ports in the Stereo Raw Audio Stream Broadcast Transmitter Custom Protocol sample application (*remote_mic_tx_raw*) causes periodic noise on the right channel for PCM and both channels for DMIC.

Investigating increase in latency for custom_protocol_trx	Configuring the RX output to SPI causes an extra 1.5 ms increase in the end-to-end latency. Now it is 8.6 ms, but it is supposed to be 7 ms.
Investigating spurious emissions for the remote mic protocol	The remote microphone protocol implementation on the transmitter side has spurious emission violation non-conformance with regulatory tests.
Installer fails if the path is too long	If the installation path is longer than 50 characters, the installer fails with an error specifying which path is too long. When the installer tries to install some of the subfolders, the entire path length exceeds the maximum allowed by Windows. If you choose a location that is different from the default installation location, ensure that it does not exceed 50 characters.
Remote microphone performance needs further evaluation	The audio stream broadcast custom protocol has not been fully characterized.

12 issues

Further Reading

See *Getting Started with RSL10* to discover:

- How to install the software development tools
- How to connect to the Evaluation and Development Board
- How to run a simple sample application that shows the board is working
- An overview of the documentation set

Release Organization

The following files are available on the same web page as these release notes:

Release Item	Description
RSL10_getting_started.pdf	Second document you should read after this one. For a full description, see Further Reading.
RSL10_SDK_Installer.msi	Installer for the standard RSL10 SDK (Oxygen Eclipse)
ONSemiconductor.RSL10.2.1.0.pack	Arm CMSIS-Pack for Oxygen
Documentation.zip	All manuals in one folder that can be accessed without installing or starting an IDE
RSL10_Utility_Apps.zip	Contains utility applications and external projects that can be used to help demonstrate and test the functionality of the RSL10 Evaluation and Development Board: ■ Standalone Flash Loader ■ Ezairo 7100-based sample audio applications
RSL10_LPDSP32_Support.zip	Startup files, Eclipse project files, a utility to integrate LPDSP32 output code with Arm Cortex®-M3 processor code, a test suite, and documentation
RSL10_release_note_history.zip	An archive file containing previous versions' release notes.

Copyright © 2018 Semiconductor Components Industries, LLC (d/b/a ON Semiconductor).