RSL10 SDK Release Notes - 2.3

December 7, 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
Apply the correct trim values is divided into two for buck coverter and LDO mode in the Sys_RFFE_SetTXPower()	The `SYS_RFFE_SetTXPower()` function, which is responsible for the VCC trim settings, now extracts the correct VCC trim value for boards that were made with manufacturing calibration process version 22 and higher. It also correctly uses LDO trim values for process versions 22 and up. For more information, see the "Manufacturing Records" section of the *RSL10 Firmware Reference*.
Added Keil Support to the CMSIS-Pack, including sample code	The CMSIS-Pack is now updated to integrate Keil support and a subset of Bluetooth Low Energy and non-Bluetooth Low Energy sample applications. For more information about the sample applications available for Keil, see the *RSL10 Sample Code User's Guide*. For information about importing the Keil CMSIS-Pack, see *Getting Started with RSL10*.
CMSIS Pack - provide links to documentation in component description	Links to RSL10 documentation are now available in each CMSIS-Pack's component description in the RTE Configuration Wizard.
Android Audio Streaming for Hearing Aid (ASHA)	Added support for the Android Audio Streaming for Hearing Aid (ASHA) service. A new sample application named *ble_android_asha* has been added to the RSL10 SDK.
Removed Bluetooth Low Energy reset from initialization	To reduce the boot time of the app, the `BLE_Reset()` function was removed from the `BLE_Initialize()` functions of all Bluetooth Low Energy applications.
An API lets the application set the maximum data length for RX	By default, the maximum RX octet is set to 251. Now the application can set its own desired maximum supported RX length by calling the following API after the `BLE_InitNoTL(0)` function: `uint8_t BLE_DeviceParam_Set_MaxRxOctet(uint8_t maxRxOctet, uint16_t maxRxTime)` The first argument is set in byte units and the second is set in microseconds.
Exposed platform reset function	In the stack, kernel memory issues a platform reset when it cannot allocate a memory space. Now you can define your own platform reset implementation with this format: `void platform_reset(uint32_t error)` By default, it calls the ARM Cortex-M3 system reset.

7 issues

Bug Fixes

Summary	Release Notes
Timeout might happen when two link layer procedures collide	Both *peripheral_server_uart* and *central_client_uart* sample applications send physical layer (PHY) change commands right after connection, causing a link layer timeout and losing the link. To fix it on the peripheral side, remove the following:

```
Connection_SetPHY(GAP_RATE_LE_2MBPS,
GAP_RATE_LE_2MBPS);
```

Getting started guide missing on the .pdsc file (CMSIS-Pack)	The *Getting Started Guide* can now be accessed from the list of books in the CMSIS-Pack.
hci_app doesn't work when imported through CMSIS-Pack	When imported through the CMSIS-Pack, the *hci_app* sample application did not work because it was configured with incorrect library variants. This issue is now fixed.
Excessively high standby current value for peripheral_server_standby app	Fixed a bug where there was excessive current during standby mode in the *peripheral_server_standby* application. This was too high to be considered a standby current. The fix that was used in this standby code was ensuring that the VDDPA trim values were being turned off at the end of the standby loop, and then turned back on at the beginning of one. This ensures that the VDDPA is not active during standby, which was the cause of the excess current.
Issue with peripheral_server_standby app not advertising at higher TX power	Fixed a bug where the *peripheral_server_standby* application would not advertise nor go to standby at higher dBm levels. This issue was fixed by ensuring that the VDDPA trim values were being turned off at the end of the standby loop, and then turned back on at the beginning of one.
SDK debug is not working with J-Link V6.40	Updated the RSL10 J-Link reset script to fix issues introduced in the SEGGER J-Link v6.40 release. This bug fix will be included in the next SEGGER J-Link version. Until an update is released by SEGGER, we recommend that you downgrade your J-Link version to V6.34h.
A redundant wakeup event in the first sleep-wakeup cycle	Fixed a redundant wake-up event that happened around 180 ms after the first sleep cycle in the *peripheral_server_sleep* application.
Fix the project build settings for the Proximity Profile	A proximity profile task now allows an application to add a proximity profile.
Bandwidth calculation doesn't apply data length extension parameters dynamically	In scatter-net use cases after the maximum length negotiation between the stack and peer device, the bandwidth for the event scheduler is not updated correctly. Now that it is resolved, the Bluetooth Low Energy links will be more stable when more than one connection is ongoing.
Wrong values are set for offsets of parameter update messages	When parameter update requests or responses were sent, fields of offsets and periodicity could be set to invalid values. The Bluetooth stack has been modified to fix this issue.
Providing a workaround solution when RC oscillator is used in sleep mode	The stability of Bluetooth Low Energy link for *peripheral_server_sleep* and *peripheral_server_standby* applications has been improved when RC OSC is used as the source of the low power clock. The *peripheral_server_sleep* app specifically will also now force `LOW_POWER_CLK_UPDATE` as enabled for any case that uses the `RTC_CLK_SRC_RC_OSC` clock.
peripheral_server_sleep: Connection drop when using external clock with LOW_POWER_CLK_UPDATE disabled	The *peripheral_server_sleep* sample application no longer drops the connection between any central server and an RSL10 board if an external clock is used with `LOW_POWER_CLK_UPDATE` disabled.
Sync_pulse DIO initialization for remote_mic_coded app	Initialize `DIO_SYNC_PULSE` as a `GPIO_OUT` in the *remote_mic_trx_coded* sample application. `Sys_DIO_Config(DIO_SYNC_PULSE, DIO_MODE_GPIO_OUT_0);` This allows the synchronization mechanism to be active on the TX side.
peripheral_server_sleep: calculation of RC OSC update time should be modified	The *peripheral_server_sleep* application can work without connection loss when the slave latency is not zero.
peripheral_server_sleep: the first dynamic update of RC OSC should be forced once transitioned to connected mode	In the *peripheral_server_sleep* application, the first dynamic update of RC OSC is forced right away once transitioned to connected mode to avoid possible connection loss due to a late RC OSC period value update.
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 *peripheral_server_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 1.5 ms by calling `BLE_LLD_Sleep_Params_Set` with the right parameter. Previously Link establishment with short intervals less than 10 ms was not able while now it is possible.

16 issues

Migrating from RSL10 SDK 2.2 to 2.3

If you use Release 2.2 code or projects in Release 2.3, remember the following:

- Compatibility of the Remote Microphone protocol is not guaranteed between SDK 2.2 and SDK 2.3. In other words, an application compiled as TX in SDK 2.2 will not necessarily work with an RX application compiled in SDK 2.3.
- Although the Release 2.2 sample code builds successfully in Release 2.3, backwards compatibility is not guaranteed.

Known Issues

The following issues are known to be outstanding in release 2.3:

Summary	Release Notes
peripheral_server_sleep app might crash with pin wakeups	The *peripheral_server_sleep* application might crash if wakeup by DIO pins is selected.
Fix bugs in peripheral_server_standby app with pin wakeups	The *peripheral_server_standby* application randomly crashes if wakeup by DIO pins is selected.
Only one DIO interrupt debounce filter can be enabled at a time	Due to a hardware limitation, only a single DIO interrupt debounce filter can be enabled at a time.
FOTA_PC_APP: Exception: STATUS_ACCESS_VIOLATION stackdump	Occasional crashes have been observed in the Firmware Over-the-Air PC application during the scan procedure.
I2C CMSIS Driver "xfer_pending" not supported	The I2C CMSIS-Driver does not support the "xfer_pending" feature due to a hardware limitation. Therefore, generating a repeated start condition on the I2C interface using the I2C CMSIS-Driver is not possible.
Remote Mic TX with Audio Shield does not work after Supplemental Calibrate app	To make the *remote_mic_tx_raw* application 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.

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	If the installation path is longer than 50 characters, the installer fails with an error specifying which path is too long. When the

long	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.

14 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.RSL10.2.3.xx.pack	Arm CMSIS-Pack for Oxygen and Keil
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).