RSL10 CMSIS-Pack Release Notes - 3.8

29 Jun 2023

RSL10 is a Bluetooth® 5.2 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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- onsemi IDE version 4.4
- Arm® Keil® uVision® IDE (tested with version V5.37.0.0)
- IAR™ Embedded Workbench® (tested with version 9.20.2)

This document lists the significant new features, bug fixes, and known issues for this release.

What's New

Firmware Changes

Summary	Release Notes	Key
LPDSP32 integration scripts updated	LPDSP32 integration scripts have been updated and now they support Python 3.	SWMCOL-4712
Vehice Access sample applications added	To support Vehicle Access Applications, two new sample applications have been added: *peripheral_server_anchor* for the anchor (i.e., the device that establishes Bluetooth Low Energy connections with the user' s mobile devices and provides sniffers with required connection-specific data); and *ble_rssi_sniffer* for sniffers (i.e., devices that sniff packets from specific Bluetooth connections). These two applications demonstrate how RSL10 can be used to implement keyless entry for a vehicle. Further details can be found in the readme files of these sample applications.	SWMCOL-4697

2 issues

Documentation Changes

Summary	Release Notes	Key
Documentation updates to RF-related voltages	The *RSL10 Hardware Reference* and the *RSL10 Firmware Reference* have been updated to reflect correct RF-related voltages.	SWMCOL-4829
The readme of print_nvr_info sample application is unclear	Verification instructions have been clarified in the readme for the *print_nvr_info* sample application.	SWMCOL-4814
Hardware Reference: NVR gain and offset value use specified for LSAD	Information on using the NVR gain and offset values for the LSAD are specified in the Interfaces chapter of the *RSL10 Hardware Reference*.	SWMCOL-4800
Hardware Reference: ADC SLOWCLK rate specified	The necessary ADC SLOWCLK rate has now been specified as exactly 1 MHz. **NOTE:** This change was implemented in SDK 3.7.	SWMCOL-4710

FOTA User Guide: Note on secure implementation	A cautionary note has been added in the Image chapter of the *RSL10 Firmware Over-the-Air User Guide*, regarding the use of public keys for signature validation when security is used.	SWMCOL-4533
Hardware Reference: DIO configuration clarification	Configuration of `DIO_MODE_INPUT` and `DIO_MODE_DISABLE` has been clarified in the Digital Input /Output chapter.	SWMCOL-3563

6 issues

Bug Fixes

Summary	Release Notes	Key
Sleep Mode sample crashes when 2Mbps support is disabled	When 2Mbps support was disabled, the sleep mode sample would not wakeup after entering sleep for the first time because the wakeup routine was missing. *sys_powermodes_wakeup* has been added to the linker script for all Sleep Mode samples as the default wakeup routine when 2Mbps wakeup support is disabled.	SWMCOL-4852
Issues when the system is transitioning from Run Mode to Sleep Mode	If an interrupt becomes pending and/or a wake up source is triggered when the system is transitioning from Run to Sleep Mode, the interrupt might not be serviced or the system might crash. To resolve those issues, in the case when sleep mode entering procedure is aborted due to a wakeup event, in `Sys_PowerModes_Sleep()` function, after the program execution falls through `SYS_WAIT_FOR_INTERRUPT` statement, code has been added to perform wakeup procedure and then call `Continue_Application()` function.	SWMCOL-4815
M3/LPDSP32 shared buffer location mismatch	The location of the DSP DRAM region used as a shared buffer between the Arm Cortex-M3 and LPDSP32 cores in audio samples has been updated to match codec source firmware available within the LPDSP32 package.	SWMCOL-4813
Primary bootloader descriptor returns wrong value	The `ValidateImage` function in *sys_boot.c* for the *bootloader* sample has been fixed, to return the image size instead of the address.	SWMCOL-4778
IRQ error and link loss	A state machine in the baseband controller can sometimes get stuck when a link is encrypted and the ongoing event is aborted by another event. Error handling has now been applied, so that when this issue occurs, the hardware can recover and the events are reprogrammed in such a way that the links are not affected.	SWMCOL-4715
Default charge pump clock updated	The default charge pump clock in a number of sample applications had been set to 125kHz. It is now set to 166kHz across all applications for better frequency deviation performance on the transmitted RF signal.	SWMCOL-4684
DFU firmware hardfaults when execution passes from FreeRTOS to Boodloader /DFU	The SysTick timer now gets disabled before the DFU is started, when `Sys_Fota_StartDfu()` is called. This avoids the hardfault when execution passes from FreeRTOS to the Boodloader/DFU.	SWMCOL-4668
Conflicting resolution to audio streams mixing	Some remote microphone protocol settings are now set based on the modulation index to give better stream discrimination.	SWMCOL-4614
JTAG can become unlocked when exiting Sleep Mode	The JTAG interface could sometimes become unlocked when returning from Sleep Mode if the `nReset` Line was not connected on the JTAG debug interface. This issue has been fixed.	SWMCOL-4207

Sample code updates for latest ATE test program (ADC calibration points)	Two new APIs, `Sys_ADC_CalibratedConfig` and `Sys_ADC_GetGain`, have been added to facilitate getting calibrated values from the ADC. Without these APIs, applications prior to release 3.8 were using default trim values for the ADC. To keep the behavior unchanged in current products, `Sys_ADC_CalibratedConfig` is still not called anywhere in the sample. Users who want to get calibrated ADC values must add the code to call `Sys_ADC_CalibratedConfig`. The existing sample application *ADC_UART* now has extra functionality that demonstrates how to convert a value reported by the ADC to a voltage, and how to apply a gain factor to get an accurate, calibrated voltage value. **Notes:** 1. These APIs rely on the following: • Valid trim records exist in NVR4. • VCC Trim value is applied prior to calling `Sys_ADC_CalibratedConfig`. 2. No action is needed for ADC offset, as it is subtracted automatically by hardware.	SWMCOL-3802

10 issues

Known Issues

The following issues are known to be outstanding in release 3.8:

> **Acknowledge inappropriate terminology**
>
> onsemi plans to lead in replacing the terms “white list", "master" and "slave” as noted in this product release. We have a plan to work with other companies to identify an industry wide solution that can eradicate non-inclusive terminology but maintains the technical relationship of the original wording. Once new terminologies are agreed upon, we will update all documentation live on the website and in all future released documents.

Summary	Release Notes	Key
Issue in ASHA audio streaming when devices are out then back in the RF range	For the *ble_android_asha* sample application, with the hearing aid at the edge of the RF range, the audio would sometimes output either muted or choppy, even after moving the hearing aid closer to the phone. The *ble_android_asha* sample application's audio queue processing code has been improved. As a result, the amount of choppy audio occurring in this scenario has been reduced. However, the overall amount of perceived choppy audio varies between Android mobile device models.	SWMCOL-4811
I2C_CMSIS_Driver sample code does not report bus error when DMA is enabled	In the *i2c_cmsis_driver* sample application, when starting communication in the reverse direction immediately after a communication transaction from one device to the other, the receiving device shows correct data received even when there is a known break in the connection such as a removed wire. It might be that a buffer is not being cleared correctly.	SWMCOL-4662
freertos_ble_peripheral_server_bond might disconnect during bonding	During bonding, *freertos_ble_peripheral_server_bond* occasionally drops the Bluetooth connection.	SWMCOL-4653
Keil IDE: Intermittent instability with sleep and standby sample applications	Bluetooth Low Energy sleep and standby sample applications in the Keil IDE sometimes do not send Bluetooth Low Energy advertising (or connection) messages as expected after waking up.	SWMCOL-4365
FOTA instability at 48 MHz	In the heart rate profile (HRP) Firmware-Over-The-Air (FOTA) sample, `Sys_Initialize()` can crash when SYSCLK is 48 MHz. To work around this, use SYSCLK at 8 MHz.	SWMCOL-4225
UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it must account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.	SWMCOL-3248
Only one DIO interrupt debounce filter can be enabled at a time	Only a single DIO interrupt debounce filter can be enabled at a time.	SWMCOL-3021

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* sample application.	SWMCOL-2823
The BootROM CRC is limited to 88 vector table entries	The CRC function of the BootROM application is not valid if more than 88 vectors are in use.	SWMCOL-2736
Controller privacy cannot be used with some smartphones	Smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used, due to a limitation of RSL10. 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.	SWMCOL-2684

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`
		SWMCOL-2435

11 issues

RSL10 CMSIS-Pack Release Notes - 3.7

08 Aug 2022

RSL10 is a Bluetooth® 5.2 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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- onsemi IDE
- Arm® Keil® uVision® IDE (tested with version V5.37.0.0)
- IAR™ Embedded Workbench® (tested with version 9.20.2)

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

Firmware Changes

Summary	Release Notes	Key
Additional trims can now be used in flash	Added functionality to make use of additional trim values, if they exist in flash.	SWMCOL-4707
Improved support for RF Tools	RSLxx vendor-specific HCI messages are now supported by the RSL10 Bluetooth stack. The HCI application has been updated so you can use RF Tools to enable TX or RX continuous wave modes (CW). In addition, now RF output power and memory addresses can be read and written with RF Tools.	SWMCOL-4667

2 issues

Documentation Changes

Summary	Release Notes	Key
GAPC_PAIRING_FAILED reason documented	The CEVA document *RW-BLE_GAP_IS_2mbps.pdf* has been updated to point to the correct table showing all the possible reasons for a `GAPC_PAIRING_FAILED` error message.	SWMCOL-4718
Firmware Reference: BLE_Set_RxWinSize_Max missing the instant_change_include description	Added a description of the input variable to the reference material in the *Firmware Reference*.	SWMCOL-4711
Getting Started: updated recommend J-Link minimum version	The recommended J-Link version has been updated to version 7.66b or higher.	SWMCOL-4699
Firmware Reference: Scan response is not documented in advertisement report type	In the *Documentation/CEVA/GAPI-IS.pdf* document, in table 23 where the values for `evt_type` in the `GAPM_ADV_REPORT_IND` message are listed, an entry for `SCAN_RSP` (= 0x4) was missing. This has now been added.	SWMCOL-4689

Firmware Reference: Encryption key length clarified	The encryption key length for has been clarified for LE secure connection pairing/bonding and legacy pairing /bonding.	SWMCOL-4678
Firmware Reference: Improved clarity on when to backup modified RF regsiters before sleep	Some RF registers are backed up by sleep/standby initialization functions. When an application changes RF transmission power, it reconfigure those RF registers. Therefore, the application must back up those registers again by calling sleep/standby initialization functions. This ensures that the correct RF transmission power is loaded at wakeup. The Description sections for the related APIs in the *RSL10 Firmware Reference* have been updated with this information.	SWMCOL-4582
Hardware Reference: wakeup pin functionality clarified	It is now explicitly stated that the wakeup pin cannot be used as a GPIO.	SWMCOL-4401
Hardware Reference: DCCLK configuration updated	Some aspects of the DCCLK configuration have been clarified in the documentation.	SWMCOL-4248

8 issues

Bug Fixes

Summary	Release Notes	Key
Improved comments associated with ADV intervals in Bluetooth Low Energy profile samples	Comments regarding the fast and reduced connection modes in the CGM and the HRP profiles have been improved for clarity.	SWMCOL-4690
Fixed ability to reject encryption key size while pairing	When a slave device sends a `GAPC_BOND_CFM` message to a master device, it can use the `accept` field to accept or reject the requested operation. This field, however, did not work when the master requested the exchange of the Long Term Key (i.e., `request` field is `GAPC_LTK_EXCH`). This bug has been fixed and now the slave device can accept or reject the Long Term Key exchange request by setting the `accept` field to `true` or `false`, respectively. For example, if the key size is small, the slave device can reject the request. When the `accept` field is set to `false`, the key size error is sent to the peer device in case there is legacy pairing. Note that the LTK is generated by the stack when there is LE secure pairing, and a key size error is not possible with a secure connection.	SWMCOL-4687
Battery service in Bluetooth Low Energy peripheral sample applications reports wrong values	Bugs associated with battery level measurements have been fixed in Bluetooth Low Energy peripheral sample applications. The battery levels reported to the central devices are now correct.	SWMCOL-4659
sleep_RAM_retention clearing of reset status conflicts with documentation	In the *sleep_RAM_retention* application, the order of resetting the `DIG_RESET_STATUS` and `ACS_RESET_STATUS` registers has been corrected (i.e. `DIG_RESET_STATUS` register has to be reset before `ACS_RESET_STATUS` register).	SWMCOL-4651

I2C CMSIS Driver xfer_pending feature not supported	A firmware workaround has been implemented to overcome the hardware limitation associated with `xfer_pending` feature of the I^2C CMSIS-Driver. Now this feature can prevent the generation of a stop condition, and thus support repeated start conditions, so the `xfer_pending` feature is now supported.	SWMCOL-4571
VDDRF and RF block must be turned on before starting the 48 MHz XTAL oscillator	In the *bootloader* application, the correct sequence of enabling VDDRF before starting the XTAL48 MHz clock has been applied.	SWMCOL-4548
Improve the stability of sleep sample code	Removed code that lowered VDDC trim setting in Bluetooth Low Energy samples with sleep to avoid potential resets and improve stability.	SWMCOL-4540
FOTA samples need DIO5 in known state to initiate FOTA	Bootloader and FOTA sample code need DIO5 in a known state to initiate the FOTA process. In those sample applications, DIO5 is configured as an input to RSL10. In the RSL10 EVB, DIO5 is connected to the pushbutton SW1 and the Atmel(R) ATSAM3U2CA-AU chip (which does USB to JTAG). Take care to ensure that the Atmel chip is always powered in any power supply configuration (by connecting a power source to `VDD_AT` at the P8 header). If the Atmel chip is not powered, it can unexpectedly pull DIO5 low causing the firmware in RSL10 to get stuck in a `while` loop. Other solutions are to remove the resistor R31 (so DIO5 is not connected to the Atmel chip), or to use a different DIO to initiate the FOTA process (this would require a modification to the sample code).	SWMCOL-4513
Add readme files to Keil IDE project configuration	Readme files now show up in the Keil IDE.	SWMCOL-4376
Missing readme files for some IAR sample applications	Added readme files for *peripheral_server_sleep* and *peripheral_server_sleep_ext* sample applications for the IAR configuration.	SWMCOL-4361
Keil IDE: SEGGER RTT files causing warning	Some applications in Keil IDE return a `SEGGER_RTT_ASM_ARMv7M.S`-related warning. This issue is fixed by SEGGER with the new SEGGER RTT file updates.	SWMCOL-4118
FlashStatus enum is defined twice and both definitions are used in bootloader project	The common definitions shared between the *rsl10_flash.h* and *rsl10_flash_rom.h* have been moved to *rsl10_flash_common.h* to avoid semantic warnings thrown by the CDT indexer.	SWMCOL-3997
Wrong signed definition for channel assessment parameters	The fields of `chnlAsses_min_thr`, `chnlAsses_max_thr` and `chnlAsses_noise_thr` that were defined as unsigned byte variables (`uint8_t`) are now defined as signed variables and applications can set them without any need to convert unsigned values to signed values when providing them to the Bluetooth Low Energy stack.	SWMCOL-3991

13 issues

Known Issues

The following issues are known to be outstanding in release 3.7:

Acknowledge inappropriate terminology

onsemi plans to lead in replacing the terms “white list", "master" and "slave” as noted in this product release. We have a plan to work with other companies to identify an industry wide solution that can eradicate non-inclusive terminology but maintains the technical relationship of the original wording. Once new terminologies are agreed upon, we will update all documentation live on the website and in all future released documents.

Summary	Release Notes	Key
Default charge pump clock should be updated	The default charge pump clock in some sample codes is currently set to 125 KHz. It should be set to 166 KHz.	SWMCOL-4684
I2C_CMSIS_Driver sample code does not report bus error when DMA is enabled	When testing the *I2C_CMSIS_Driver* sample application, starting communication in the reverse direction immediately after a communication transaction from one device to the other, the receiving device shows correct data received even when there is a known break in the connection such as a removed wire. It may be that a buffer is not being cleared correctly.	SWMCOL-4662
freertos_ble_peripheral_server_bond may disconnect during bonding	During bonding, *freertos_ble_peripheral_server_bond* may occasionally drop the Bluetooth connection.	SWMCOL-4653
Keil IDE: Intermittent instability with sleep and standby sample applications	Bluetooth Low Energy sleep and standby sample applications in Keil IDE sometimes do not send Bluetooth Low Energy advertising (or connection) messages as expected after waking up.	SWMCOL-4365
FOTA instability at 48 MHz	In the heart rate profile (HRP) Firmware-Over-The-Air (FOTA) sample, `Sys_Initialize()` can crash when SYSCLK is 48 MHz. To work around this, use SYSCLK at 8 MHz.	SWMCOL-4225

JTAG can become unlocked when exiting sleep mode	When using the JTAG locking mechanism, with the current implementation of `Sys_PowerModes_Sleep_Init` and `Sys_PowerModes_Sleep_Init_2Mbps` in *rsl10_sys_power_modes.c* the device becomes unlocked when returning from sleep mode.
	
	To fix this, the `Sys_PowerModes_Sleep_Init and Sys_PowerModes_Sleep_Init_2Mbps` routines in *rsl10_sys_power_modes.c* must have the following code added to them:
	
	/* Write lock keys and wake-up restore address */
	
	if (SYSCTRL->DBG_LOCK == DBG_ACCESS_UNLOCKED)
	
	{ *((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x00)) = 0; }
	
	else
	{}}
	*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x00)) = DBG_ACCESS_LOCK;
	}
	*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x04)) = SYSCTRL->DBG_LOCK_KEY[0];
	*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x08)) = SYSCTRL->DBG_LOCK_KEY[1];
	*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x0C)) = SYSCTRL->DBG_LOCK_KEY[2];
	*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x10)) = SYSCTRL->DBG_LOCK_KEY[3];
	*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x14)) = sleep_mode_env->app_addr;
	
	}
		SWMCOL-4207

UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it must account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.	SWMCOL-3248
Only one DIO interrupt debounce filter can be enabled at a time	Only a single DIO interrupt debounce filter can be enabled at a time.	SWMCOL-3021

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.	SWMCOL-2823
The BootROM CRC is limited to 88 vector table entries	The CRC function of the BootROM application is not valid if more than 88 vectors are in use.	SWMCOL-2736
Controller privacy cannot be used with some smartphones	Smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used due to a limitation of RSL10. 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.	SWMCOL-2684

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`
		SWMCOL-2435

12 issues

RSL10 CMSIS-Pack Release Notes - 3.6

14 Dec 2021

RSL10 is a Bluetooth® 5.2 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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- onsemi IDE
- Arm® Keil® uVision® IDE (tested with version V5.28.0.0)
- IAR™ Embedded Workbench® (tested with version 8.32.4)

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

Firmware Changes

Summary	Release Notes
hci_app refactored to use RF_Tools	To use the *RF_Tools* application for RSL10, the *hci_app* sample application has been renamed to *hci*, and the DTM scripts have been removed. Download the *RF_Tools.zip* package to access *RF_Tools*. Note that CW is not supported at this time for RSL10.
Light stack is no longer supported	The light stack is no longer supported and has been removed from the SDK. The samples *peripheral_server_sleep* and *peripheral_server_sleep_ext* no longer demonstrate the light stack.

Option of setting offsets of param update to app now provided	When some smart phone models start ASHA stereo streaming (as seen for Broadcom-based devices), they program left and right anchor points very close to each other, causing audio artifact issues for one side.
	Through this API,
	`BLE_Set_ParmUpdtReqOffsets(bool enable, uint16_t *desired_offsets)`
	an application can set offsets 0 to 5, sent in a connection parameter update request over the air, to its own desired values, to trigger smart devices to change the anchor points such that events are programmed with a longer space of time between them.
	By default this is disabled. To enable it, an application can call this API any time after Bluetooth Low Energy initialization, as in this example:
	`uint16_t offsets[6] = {2,4,7,0xffff, 0xffff,0xffff};`
	`BLE_Set_ParmUpdtReqOffsets(true, desired_offsets);`

Event duration calculation and event priority handling have been improved	The calculation of events duration in central and peripheral roles has been changed, to improve the programming of events in multi-link use cases. Even priority arbitration logic has been updated such that if no packet is received for six successive events, the priority of the related link is set to maximum.

Operation across full temperature range	For proper operation across your RSL10 product's full operating temperature, specific settings must be applied to the RC oscillator frequency setting and retention regulator trims before entering sleep mode. The RC oscillator must be set to 3 MHz. Non-automotive RSL10 products must apply retention regulator trims of 0x01. Automotive RSL10 products must apply retention regulator trims of 0x03. The applicable sample applications now set the RC oscillator 3 MHz and apply retention regulator trims of 0x01 before entering sleep mode. This default configuration is appropriate for all non-automotive RSL10 products. Users of automotive RSL10 must manually apply retention regulator trims of 0x03. This can easily be done by selecting 0x3 for VDDTRet Trim value, VDDMRet Trim value, and VDDCRet Trim value under Retention Regulator Trim Configuration in the *RTE_Device.h* file of applicable sample applications. Note that if the low power clock source is RC 32 kHz oscillator (i.e., RTC_CLK_SRC is defined to RTC_CLK_SRC_RC_OSC in the *app.h* file), VDDT retention regulator trim must always be set to 0x3 for all RSL10 products. Failure to use the correct RC oscillator frequency and retention regulator trims can result in incorrect operation across the temperature range when sleep mode is used. See the sample code *readme* files for additional detail.
RSL10 now supports Bluetooth Low Energy 5.2	The Bluetooth SIG deprecated RSL10 QDID 92528 due to an outdated profile. A new RSL10 QDID of 168840 has been created against test case specification TCRL 2020-1. With the new QDID and Bluetooth stack in this SDK release, RSL10 now supports the Bluetooth Low Energy 5.2 core specification. No new Bluetooth features have been added, but asymmetric PHY support for 1 and 2 Mbps is now certified. Also note that in LE Security Mode 1 Level 4, the device completes pairing only if the encryption key size is not less than 16 bytes. After Feb 2022, the RSL10 QDID 92528 can no longer be used to qualify new designs with the Bluetooth SIG. We recommend that all customers who have not previously qualified a design with the Bluetooth SIG update to the stack in this SDK release and use the new RSL10 QDID of 168840 for qualification with the SIG. This QDID does not contain profiles. A new Profile Subsystem QDID will be available soon. Any customers who have qualified designs with the Bluetooth SIG, and wish to use the stack in this SDK release, are advised to contact the Bluetooth SIG to determine if any declaration updates are required. No action is required for customers who have qualified designs with the Bluetooth SIG and do not use the stack in this SDK release. QDID 92528 is still valid for previously qualified designs with the Bluetooth SIG.
Bluetooth Low Energy event priorities	Any Bluetooth Low Energy event has a priority set by default that can be found in `enum rwip_prio_dft` in file *rwip_config.h*. If an event cannot be programmed because of conflict or overlap with another higher-priority event, its priority is incremented to the default priority increment per event as found in `enum rwip_incr_dft`. Now the API has a new capability: `uint8_t BLE_Set_EventPriority(uint8_t eventIndex, uint8_t priorityValue, uint8_t incrementValue)` Priority `priorityValue` and increment value `incrementValue` can be set for an event where the index to its event is specified through `eventIndex`. The meaning of the index can be found in `enum rwip_prio_idx` defined in file *rwip_config.h*.

7 issues

Documentation Changes

Summary	Release Notes
Hardware Reference: Flash_WriteBuffer() has no maximum word limit	Made changes to Section 7.2.2 "Flash Memory" in the Hardware Reference to clarify the organization of the flash memory.
Added VDDxRet Trim Value Setting Information to READMEs	Important notes have been been added into the readme files of code samples using Sleep Mode, regarding how trim values of retention regulators need to be configured when Sleep Mode is used.
Getting Started Guide: documented new default path to RSL10 CMSIS-Pack	The default path to the RSL10 CMSIS-Pack, which used to be *C:\Users\<username>\ON_Semiconductor\PACK*, is now *C:\Users\<username>\AppData\Local\Arm\Packs (or %LOCALAPPDATA%\Arm\Packs* on Windows). This path has changed so that the customer can always find the RSL10 CMSIS-Pack in the same main folder, whether they use the onsemi IDE, Keil, or IAR.
Firmware Reference: specified which priority is high and which is low for BLE_Set_EventPriority	Which end of the priority scale represents high priority and which represents low has been clarified in the *RSL10 Firmware Reference*. The higher the number, the higher the priority.
RSL10 documentation rebranded	The RSL10 documentation has been rebranded to match onsemi's new name, logo, etc.

Firmware Reference: BLE_Set_ScanConIndStatusCallBack documented	Documentation on a new API function, `BLE_Set_ScanConIndStatusCallBack`, is now available in the *RSL10 Firmware Reference*.
Firmware Reference: BLE_Set_ParmUpdtReqOffsets documented	Documentation for a new API function, `BLE_Set_ParmUpdtReqOffsets`, is now available in the *RSL 10 Firmware Reference*.
Firmware Reference: Bluetooth Security Notice added	An important notice of a possible Bluetooth security concern has been added to the *RSL10 Firmware Reference*.
Hardware Reference: RTC alarm interrupt clarified	The functionality of the RTC Alarm has now been explained in the *RSL10 Hardware Reference*.
Hardware Reference: RSL10 Does Not Support Changing SPI Clock Phase (CPHA)	Lack of support for CPHA has now been documented in the *RSL10 Hardware Reference*.
Remove "!!!WORKAROUND!!!" Text from SPI Driver	Removed misleading comments indicating that some code was a workaround instead of a proper solution.
Hardware Reference: TIMEOUT_RESET_FLAG bit field of ACS_RESET_STATUS register logic described	The logic behind setting the `TIMEOUT_RESET_FLAG` bit in the `ACS_RESET_STATUS` register, which affects timeout duration and power management state/s, is now described in the *RSL10 Hardware Reference*.
Hardware Reference: Describe ACS_WAKEUP_GP_DATA	The `ACS_WAKEUP_GP_DATA` register is now fully described in the *RSL10 Hardware Reference*.

13 issues

Bug Fixes

Summary	Release Notes
Improved commenting in SPI driver	A comment has been added to the SPI Driver to help clarify the cautionary note in the *RSL10 Hardware Reference* about avoiding data corruption when using DMA with the SPI Driver.
L2C Connection Count updates incorrectly in L2C abstraction	The connection counter in the Bluetooth Low Energy L^2C abstraction is now being updated on an `L2CC_LECB_CONNECT_IND` connection event, to ensure that both central and peripheral roles have the correct connection count.
Changes in how public key is generated and used in pairing	The changes made deal with how the public key is generated and used. The public key cannot be read. For secure LE, the public key is not generated at device config - instead it is generated during the pairing process. For every new secure pairing operation, a new public key is generated.
Connection parameters updates are not repeated by peripheral role	When a central device sent a connection parameter update request, and API `BLE_ConParam_SlaveRspOption(1)` was used, in a case where the peripheral device could not find the right slot for its scheduler, it rejected the parameter request. Now this is fixed, and it always repeats the requested parameters when asked by the application.
Corrected functionality of BLE_ADV_Flags _Set() function	When using the `BLE_ADV_Flags_Set(1)` API, the stack has been checking the length of the ADV data without considering that since ADV flags are not added anymore, the API can accept 3 bytes more of ADV data. This is now fixed.
The values in SCID and DCID of L2CAP disconnect rsp are now corrected	DCID and SCID values in `L2CAP_DISCONNECTION_RSP` have been swapped, and now the stack sends the response with the expected source and destination CID values.

Offset0 value in connection param response is bigger than expectation	The Bluetooth Low Energy stack has been able to send a connection parameter update response even with the `offset0` value in the suggested anchor points being higher than connection interval. This has now been fixed.
Reported RSSI for empty packets is not correct	The reported RSSI for empty packets has been corrected.
Link loss is seen in scatter-net scenarios	When controller privacy is used in parallel to links that are encrypted, in a special HW state machine, Bluetooth Low Energy error interrupts can happen. A mechanism had been added to avoid this possibility. Since, in RSL10, controller privacy is not supposed to be enabled, this mechanism has been removed. If for any reason a Bluetooth Low Energy HE error happens, a Bluetooth Low Energy recovery functionality has been added that recovers the Bluetooth Low Energy HW, and programs Bluetooth Low Energy so that events and links are not lost.

9 issues

Known Issues

The following issues are known to be outstanding in release 3.6:

Acknowledge inappropriate terminology

onsemi plans to lead in replacing the terms “white list", "master" and "slave” as noted in this product release. We have a plan to work with other companies to identify an industry wide solution that can eradicate non-inclusive terminology but maintains the technical relationship of the original wording. Once new terminologies are agreed upon, we will update all documentation live on the website and in all future released documents.

Summary	Release Notes
I2C_CMSIS_Driver sample app doesn't report bus error when DMA is enabled	In the I2C_CMSIS_Driver sample application, when starting communication in the reverse direction immediately after a communication transaction from one device to the other, the receiving device shows correct data received even when there is a known break in the connection such as a removed wire.
Battery service with peripheral_server always reads default value	Battery level readings are erroneous using the battery service with the *ble_peripheral_server* sample application.
freertos_ble_peripheral_server_bond may disconnect during bonding	During bonding, *freertos_ble_peripheral_server_bond* may occasionally drop the Bluetooth connection.
Keil: Intermittent instability with sleep and standby sample applications	Bluetooth Low Energy sleep and standby sample applications in Keil sometimes do not send Bluetooth Low Energy advertising (or connection) messages as expected after waking up.

Printing APP_DIS_MANUFACTURER_NAME_LEN prints extra characters	In the *app_diss.h* file of the *ble_peripheral_server_cgm* sample application, on line 25, `#define APP_DIS_MANUFACTURER_NAME_LEN (20)` is incorrect. Corrective Action: change the line to read `#define APP_DIS_MANUFACTURER_NAME_LEN (14)` to avoid the few extra characters printed out after the intended string.
FOTA instability at 48 MHz	In the heart rate profile (HRP) Firmware-Over-The-Air (FOTA) sample, `Sys_Initialize()` can crash when SYCLOCK is 48 MHz. To work around this, use SYCLOCK at 8 MHz.

JTAG can become unlocked when exiting sleep mode	When using the JTAG locking mechanism, with the current implementation of `Sys_PowerModes_Sleep_Init` and `Sys_PowerModes_Sleep_Init_2Mbps` in *rsl10_sys_power_modes.c* the device becomes unlocked when returning from sleep mode.
	
	To fix this, the `Sys_PowerModes_Sleep_Init and Sys_PowerModes_Sleep_Init_2Mbps` routines in *rsl10_sys_power_modes.c* must have the following code added to them:
	
	`/* Write lock keys and wake-up restore address */`
	`if (SYSCTRL->DBG_LOCK == DBG_ACCESS_UNLOCKED)`
	`{`
	`  *((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x00)) = 0;`
	`}`
	`else`
	`{`
	`  *((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x00)) = DBG_ACCESS_LOCK;`
	`}`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x04)) = SYSCTRL->DBG_LOCK_KEY[0];`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x08)) = SYSCTRL->DBG_LOCK_KEY[1];`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x0C)) = SYSCTRL->DBG_LOCK_KEY[2];`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x10)) = SYSCTRL->DBG_LOCK_KEY[3];`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x14)) = sleep_mode_env->app_addr;`

Keil: SEGGER RTT files causing warning	Some applications in Keil return a `SEGGER_RTT_ASM_ARMv7M.S`-related warning. This issue is fixed by SEGGER, but the version of the Keil Arm compiler used is incompatible with the new SEGGER RTT file updates.
Wrong signed definition for channel assessment parameters	The fields of `chnlAsses_min_thr`, `chnlAsses_max_thr`, and `chnlAsses_noise_thr` are defined as unsigned byte variables (`uint8_t`), when they need to be signed byte variables (`int8_t`) to match with the channel assessment parameters used in the Bluetooth Low Energy stack. To avoid the mismatch, users need to define the values in a way that can be aligned with the Bluetooth Low Energy stack. For example, to have a value -10, in a parameter passed in the function `Device_Param_Read()`, when `chnlAsses_param_src_type` is `APP_PROVIDED`, a value of 0xF6 can be used. (Values need to be between -128 (0x80) and 127 (0x7F).)
UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it must account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.
Only one DIO interrupt debounce filter can be enabled at a time	Only a single DIO interrupt debounce filter can be enabled at a time.
I2C CMSIS Driver xfer_pending feature not supported	The I^2C CMSIS-Driver does not support the `xfer_pending` feature due to a hardware limitation. Therefore, generating a repeated start condition on the I^2C interface using the I^2C 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.

The BootROM CRC is limited to 88 vector table entries	The CRC function of the BootROM application is not valid if more than 88 vectors are in use.
Controller privacy cannot be used with some smartphones	Smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used, due to a limitation of RSL10. 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.

17 issues

RSL10 CMSIS-Pack Release Notes - 3.5

03 Mar 2021

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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- ON Semiconductor IDE
- Arm® Keil® uVision® IDE (tested with version V5.28.0.0)
- IAR™ Embedded Workbench® (tested with version 8.32.4)

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

Firmware Changes

Summary	Release Notes
Bluetooth Low Energy scan on a fixed channel	The Bluetooth Low Energy specification does not have a standard way to set a scan channel, and it always scans on three channels. Now scanning on a single channel is possible through this API: `BLE_Set_ScanChannel(uint8_t single_chnl, uint8_t desired_channel)` To enable this feature, call the API with `single_chnl = 1` and pass argument `desired_channel` to one of three channels: 37, 38, or 39.

Adding API to extract Bluetooth Low Energy connection parameters and timing	Connection parameters and timing can be extracted through the API function:
	
	`void BLE_Set_ConEstablishCallBack(void *callBack)`
	
	When connection confirmation is sent to the stack, a callback function can be registered so that at the next event, connection parameters and anchor point timing are provided through a registered callback function. The callback function is automatically unregistered once it is called and the application needs to register for any other connection it might want.
	
	The same approach can be used to get notified when a channel map indication is applied in the stack for a connection by registering an application callback function using the API function:
	
	`void BLE_Set_ChnlMapIndCallBack(void *callBack)`
	
	Once application callback functions are called, `API BLE_Extract_ConParam()` can be called in the application to extract the parameters. It has the format:
	
	`uint8_t BLE_Extract_ConParam(uint8_t conidx, struct ble_con_evt_param_t *param)`
	
	An application can call `BLE_Extract_ConParam()` at any time to get the next event timing and parameters of any desired link.

Adding an API function to read connection event counter	Through the API function `BLE_Get_conEvtCount(uint8_t conidx)`, an application can read the current connection event counter for a desired connection index. It returns zero if the connection index is not valid or the link is disconnected.
Adding an API function for choosing parameters to send in connection parameter request	`BLE_ConParam_SlaveRspOption(uint8_t repeatMasterParams)` allows the application to choose offset0 to 5, preferred periodicity, and reference connection event count are sent back exactly from ones that are received in central connection parameters request or the ones are preferred and calculated by the stack based on the ongoing scheduled events. **Note:** it can only be used for the peripheral role. The default value for `repeatMasterParams` is zero. Zero means that the function will respond with what is calculated by the stack; one means it will repeats the control parameters. It can be called anytime after the `BLE_Initialize()` function call. This function meets the need of some smart devices that reject parameter update responses sent by the peripheral role. By choosing to repeat the central device parameters, the rejection will be resolved or mitigated.
Bluetooth Low Energy sample application containing both FOTA and sleep	New *ble_peripheral_server_hrp_forta_sleep* sample with Bluetooth Low Energy abstraction, Firmware-Over-The-Air (FOTA) and sleep mode.
Secure boot loader sample	A *secure_bootloader* sample has been added.

6 issues

Documentation Changes

Summary	Release Notes
Getting Started Guide: improve PRINT macro instructions	Added explicit instructions for each supported IDE about how to add the OUTPUT_INTERFACE configuration to the preprocessor settings.
Getting Started Guide: alternate way of opening CMSIS perspective	The latest ON Semiconductor IDE features an easy access button to display the CMSIS-Pack Manager perspective.
Updated RivieraWaves CGM profile document	The RivieraWaves BLE Continuous Glucose Monitoring Profile Interface Specification has been updated as of 2021-01-08. It is in *documentation/ceva/RW-BLE-PRF-CGMP-IS.pdf* file.
Hardware Reference: Flash_WriteBuffer() has a maximum length of 32 words	Made changes to Section 7.2.2 "Flash Memory" in the Hardware Reference to clarify that the `Flash_WriteBuffer()` function has a limit of 32 words.
Getting Started Guide: correction to debugging applications that do not start at the base address of flash	Added an instruction to the Run/Restart commands that sets the program counter (pc).
Hardware Reference: VDDM flag description	The description for the `VDDM_RESET_FLAG` bit in the `ACS_RESET_STATUS` register now correctly identifies what triggers the flag.
Firmware Reference: Flash_WriteBuffer command order	Corrected the command order in the Flash library chapter of the Firmware Reference.
Firmware Reference: clarify entering and waking from sleep	Rewrote part of Section 5.3.3 "Boot and Wakeup Initialization" to clarify how enter and exit sleep mode.

8 issues

Bug Fixes

Summary	Release Notes

Channel assessment timeout is not configured	In *rsl10_protocol.c*, `chnlAsses_timer_cnt` was set instead of `chnlAsses_timer_duration` in case PARAM_ID_BLE_CA_TIMER_DUR. This has been resolved. In addition, if `chnlAsses_timer_duration` is set to zero, then updating the channel maps in the central role will be disabled. (Channel map update is initiated in the central role when the channel assessment detects that some channels do not have good quality.)
Wrong enum for cfm_pairing_lvl	Corrected enumerations assigned to variable `cfm_pairing_lvl`.
Encryption request not always receiving response	In the peripheral role, when a link layer procedure was ongoing and an encryption request is received, the device (peripheral role) did not always respond to it. The issue is now fixed and it waits to complete the ongoing procedure before responding to the encryption request.

3 issues

Known Issues

The following issues are known to be outstanding in release 3.5:

> **Acknowledge inappropriate terminology**
>
> ON Semiconductor plans to lead in replacing the terms "white list", "master" and "slave" as noted in this product release. We have a plan to work with other companies to identify an industry wide solution that can eradicate non-inclusive terminology but maintains the technical relationship of the original wording. Once new terminologies are agreed upon, we will update all documentation live on the website and in all future released documents.

Summary	Release Notes
Mesh Package not compatible with latest RSL10 CMSIS-Pack and ON Semiconductor IDE	The RSL10 Mesh Package ONSemiconductor.RSL10Mesh.1.6.79.pack is not compatible with the RSL10 CMSIS-Pack 3.5 and ON Semiconductor IDE 3.4.0.48 because of Bluetooth Low Energy stack changes and the Arm toolchain update to version 10. To build the RSL10 Mesh Package with the RSL10 CMSIS-Pack 3.5 and ON Semiconductor IDE 3.4.0.48, follow these steps: • RSL10 CMSIS-Pack - Open the ble_mesh/code/ble_std.c file, go to line 1080 and replace "*cfm->auth = GAP_AUTH_REQ_NO_MITM_BOND;*" with "*cfm->pairing_lvl = GAP_PAIRING_BOND_UNAUTH;*" • IDE - Right click on the ble_mesh project and go to Properties->C/C++ Build->Settings->Tool Settings-> Cross ARM C Linker->Miscellaneous and check the checkbox "Do not use syscalls (--specs=nosys.specs). • IDE - Go to Properties->C/C++ Build->Settings->Optimization->Other Optimization flags and add '**-fcommon**' and click "Apply and Close".
Keil: Intermittent instability with sleep and standby sample applications	Bluetooth low energy sleep and standby sample applications in Keil will sometimes not send Bluetooth Low Energy advertising (or connection) messages as expected after waking up.
Printing APP_DIS_MANUFACTURER_NAME_LEN prints extra characters	In the *app_diss.h* file of the *ble_peripheral_server_cgm* sample application, on line 25 `#define APP_DIS_MANUFACTURER_NAME_LEN (20)` is incorrect. Corrective Action: change the line to read `#define APP_DIS_MANUFACTURER_NAME_LEN (14)` to avoid the few extra characters printed out after the intended string.
FOTA instability at 48MHz	In the heart rate profile (HRP) Firmware-Over-The-Air (FOTA) sample, `Sys_Initialize()` can crash when SYCLOCK is 48MHz. To work around this, use SYCLOCK at 8MHz.

JTAG can become unlocked when exiting sleep mode	If using the JTAG locking mechanism, with the current implementation of `Sys_PowerModes_Sleep_Init` and `Sys_PowerModes_Sleep_Init_2Mbps` in *rsl10_sys_power_modes.c* the device will become unlocked when returning from sleep mode.
	
	To fix this, the `Sys_PowerModes_Sleep_Init and Sys_PowerModes_Sleep_Init_2Mbps` routines in *rsl10_sys_power_modes.c* must have the following code added to them:
	
	`/* Write lock keys and wake-up restore address */`
	`if (SYSCTRL->DBG_LOCK == DBG_ACCESS_UNLOCKED)`
	`{`
	`    *((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x00)) = 0;`
	`}`
	`else`
	`{`
	`    *((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x00)) = DBG_ACCESS_LOCK;`
	`}`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x04)) = SYSCTRL->DBG_LOCK_KEY[0];`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x08)) = SYSCTRL->DBG_LOCK_KEY[1];`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x0C)) = SYSCTRL->DBG_LOCK_KEY[2];`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x10)) = SYSCTRL->DBG_LOCK_KEY[3];`
	`*((volatile uint32_t *) (sleep_mode_env->wakeup_addr + 0x14)) = sleep_mode_env->app_addr;`

Keil: SEGGER RTT files causing warning	Some applications in Keil return a SEGGER_RTT_ASM_ARMv7M.S related warning. This issue is fixed by SEGGER, but the version of Keil ARM compiler used is incompatible with the new SEGGER RTT file updates.
Wrong signed definition for channel assessment parameters	The fields of `chnlAsses_min_thr`, `chnlAsses_max_thr`, and `chnlAsses_noise_thr` are defined as unsigned byte variables (`uint8_t`), when they should be signed byte variables (`int8_t`) to match with the channel assessment parameters used in the Bluetooth Low Energy stack. To avoid the mismatch, users need to define the values in a way that can be aligned with the Bluetooth Low Energy stack. For example, to have a value -10, in a parameter passed in the function `Device_Param_Read()`, when `chnlAsses_param_src_type` is `APP_PROVIDED`, a value 0xF6 can be used. (Values should be between -128 (0x80) and 127 (0x7F).)
UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it must account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.
Only one DIO interrupt debounce filter can be enabled at a time	Only a single DIO interrupt debounce filter can be enabled at a time.
I2C CMSIS Driver xfer_pending feature 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.
The BootROM CRC is limited to 88 vector table entries	The CRC function of the BootROM application is not valid if more than 88 vectors are in use.
Controller privacy cannot be used with some smartphones	Smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used, due to a limitation of RSL10. 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.

15 issues

RSL10 CMSIS-Pack Release Notes – 3.4.2

September 8, 2020

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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- ON Semiconductor IDE
- Arm® Keil® uVision® IDE (tested with version V5.29.0.0)
- IAR™ Embedded Workbench® (tested with version 8.32.4)

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

Firmware Changes

Summary	Release Notes

No issues found

Documentation Changes

Summary	Release Notes

No issues found

Bug Fixes

Summary	Release Notes
Bluetooth Low Energy stack enumeration name needs correcting	Now using correct Bluetooth Low Energy stack enumeration from `gap_pairing_lvl` in connection confirmation for both secure and legacy connections.

1 issue

Known Issues

The following issues are known to be outstanding in release 3.4.2:

Summary	Release Notes
peripheral_server_standby in Keil sometimes doesn't advertise after wakeup	The peripheral server standby sample code is having an issue in the Keil IDE: from time to time, after waking up, it does not send Bluetooth Low Energy advertising (or connection) messages as expected.
Printing APP_DIS_MANUFACTURER_NAME_LEN prints extra characters	In the *app_diss.h* file of the *ble_peripheral_server_cgm* sample application, on line 25 `#define APP_DIS_MANUFACTURER_NAME_LEN (20)` is incorrect. Corrective Action: change the line to read `#define APP_DIS_MANUFACTURER_NAME_LEN (14)` to avoid the few extra characters printed out after the intended string.

JTAG can become unlocked when exiting sleep mode	If using the JTAG locking mechanism, with the current implementation of `Sys_PowerModes_Sleep_Init` and `Sys_PowerModes_Sleep_Init_2Mbps` in *rsl10_sys_power_modes.c* the device will become unlocked when returning from sleep mode.
	
	To fix this, the `Sys_PowerModes_Sleep_Init and Sys_PowerModes_Sleep_Init_2Mbps` routines in *rsl10_sys_power_modes.c* must have the following code added to them:
	
	`/* Write lock keys and wake-up restore address */`
	`if (SYSCTRL->DBG_LOCK == DBG_ACCESS_UNLOCKED)`
	`{`
	`*((volatile uint32_t *)`
	`(sleep_mode_env->wakeup_addr + 0x00)) = 0;`
	`}`
	`else`
	`{`
	`*((volatile uint32_t *)`
	`(sleep_mode_env->wakeup_addr + 0x00)) =`
	`DBG_ACCESS_LOCK;`
	`}`
	`*((volatile uint32_t *)`
	`(sleep_mode_env->wakeup_addr + 0x04)) =`
	`SYSCTRL->DBG_LOCK_KEY[0];`
	`*((volatile uint32_t *)`
	`(sleep_mode_env->wakeup_addr + 0x08)) =`
	`SYSCTRL->DBG_LOCK_KEY[1];`
	`*((volatile uint32_t *)`
	`(sleep_mode_env->wakeup_addr + 0x0C)) =`
	`SYSCTRL->DBG_LOCK_KEY[2];`
	`*((volatile uint32_t *)`
	`(sleep_mode_env->wakeup_addr + 0x10)) =`
	`SYSCTRL->DBG_LOCK_KEY[3];`
	`*((volatile uint32_t *)`
	`(sleep_mode_env->wakeup_addr + 0x14)) =`
	`sleep_mode_env->app_addr;`

Wrong signed definition for channel assessment parameters	The fields of `chnlAsses_min_thr`, `chnlAsses_max_thr`, and `chnlAsses_noise_thr` are defined as unsigned byte variables (`uint8_t`), when they should be signed byte variables (`int8_t`) to match with the channel assessment parameters used in the Bluetooth Low Energy stack. To avoid the mismatch, users need to define the values in a way that can be aligned with the Bluetooth Low Energy stack. For example, to have a value -10, in a parameter passed in the function `Device_Param_Read()`, when `chnlAsses_param_src_type` is `APP_PROVIDED`, a value 0xF6 can be used. (Values should be between -128 (0x80) and 127 (0x7F).)
UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it must account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.
Only one DIO interrupt debounce filter can be enabled at a time	Only a single DIO interrupt debounce filter can be enabled at a time.
I2C CMSIS Driver xfer_pending feature 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.
The BootROM CRC is limited to 88 vector table entries	The CRC function of the BootROM application is not valid if more than 88 vectors are in use.
Controller privacy cannot be used with	Smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used, due to a

some smartphones	limitation of RSL10. 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.

12 issues

RSL10 CMSIS-Pack Release Notes – 3.4

August 11, 2020

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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- ON Semiconductor IDE
- Arm® Keil® uVision® IDE (tested with version V5.29.0.0)
- IAR™ Embedded Workbench® (tested with version 8.32.4)

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

Firmware Changes

Summary	Release Notes
Sample application for long duration sleep use cases	The system library has been updated to support Sleep Mode with I/O wakeup for long duration sleep use cases. A new sample application called *peripheral_server_sleep_ext* has been added to demonstrate this feature.
CGM profile sample application	A new sample application called *ble_peripheral_server_cgm* demonstrates the usage of the Continuous Glucose Monitoring Bluetooth Low Energy Profile.

2 issues

Documentation Changes

Summary	Release Notes
Firmware Reference: MANU_INFO_OSC_32K calibration setting	The trimmed bit-field description was corrected for `MANU_INFO_OSC_32K` to clarify that all 7 bits of `FTRIM_32K` are included.
Firmware Reference: How to modify upper bytes of advertising packet	Documented how to use the `BLE_ADV_Flags_Set` function to allow the use of the upper three bytes of an advertising packet.

2 issues

Bug Fixes

Summary	Release Notes
IAR sample applications can't open readme files	The readme files in IAR project configurations were listed as `.txt` files instead of `.md` files; this has been fixed so that the readme files can now be found from the IAR IDE.
Standalone flashloader PC application - access violation	Reading the version information from the resource file was causing an access violation. It has been fixed.
ASHA readonlyproperty characteristic is not saved correctly in the service database	`ReadOnlyProperty` structure of ASHA service has different fields with different size types so that when the compiler allocates addresses to them, padding can be added. Thus `memcpy` from structure to attribute read response can result in some fields of this characteristics being reported incorrectly. The issue now is fixed.

Inconsistent RSSI value of advertisements	Fixed a case where in some situations, the RSSI value of advertisements to host/application could be inconsistent. This is now fixed in the Bluetooth Low Energy stack.
Param update response is not sent in some cases	When a param update response was rejected by a master device and the master device sent another param update request immediately, the slave device might not respond. This is now fixed in the Bluetooth Low Energy stack.
Maximum Rx octet size cannot be set for full Bluetooth Low Energy stack	API `BLE_DeviceParam_Set_MaxRxOcte()` was working only in HCI and MFi Bluetooth Low Energy stack variants. Now it is working for full and light Bluetooth Low Energy stack libraries as well.
Audio noise issue when RSL10 is the receiver of a transmitter based on Ezairo 7150	When an Ezairo 7150-based transmitter sends audio using the remote microphone protocol to an RSL10/Ezairo 7160-based receiver of the audio, the audio could be noisy on the right channel. This is now fixed.
License text missing	The missing license text in the bootloader project files was added to be consistent with other files.
Bluetooth Low Energy stack enumeration name updates	Enumeration name `auth` in connection confirmation and bond indication have been updated to `pairing level` for clarity. The updated enumerations found in *gapc_task.h* now must be used at the application level to avoid build errors.
Renew duration for PRA in host privacy limit is incorrect	The Bluetooth Low Energy stack has a minimum private resolvable address (PRA) renew duration of 1 second (was 150 seconds) in host privacy mode. The GAP API document has also been updated accordingly.

10 issues

Known Issues

The following issues are known to be outstanding in release 3.4:

Summary	Release Notes
peripheral_server_standby in Keil sometimes doesn't advertise after wakeup	The peripheral server standby sample code is having an issue in the Keil IDE: from time to time, after waking up, it does not send Bluetooth Low Energy advertising (or connection) messages as expected.
Printing APP_DIS_MANUFACTURER_NAME_LEN prints extra characters	In the *app_diss.h* file of the *ble_peripheral_server_cgm* sample application, on line 25 `#define APP_DIS_MANUFACTURER_NAME_LEN (20)` is incorrect. Corrective Action: change the line to read `#define APP_DIS_MANUFACTURER_NAME_LEN (14)` to avoid the few extra characters printed out after the intended string.
JTAG can become unlocked when exiting sleep mode	If using the JTAG locking mechanism, with the current implementation of `Sys_PowerModes_Sleep_Init` and `Sys_PowerModes_Sleep_Init_2Mbps` in *rsl10_sys_power_modes.c* the device will become unlocked when returning from sleep mode. To fix this, the `Sys_PowerModes_Sleep_Init and Sys_PowerModes_Sleep_Init_2Mbps` routines in *rsl10_sys_power_modes.c* must have the following code added to them:

```
/* Write lock keys and wake-up restore address */
if (SYSCTRL->DBG_LOCK == DBG_ACCESS_UNLOCKED)
{
    *((volatile uint32_t *)
(sleep_mode_env->wakeup_addr + 0x00)) = 0;
}
else
{
    *((volatile uint32_t *)
(sleep_mode_env->wakeup_addr + 0x00)) =
DBG_ACCESS_LOCK;
}
*((volatile uint32_t *)
(sleep_mode_env->wakeup_addr + 0x04)) =
SYSCTRL->DBG_LOCK_KEY[0];
*((volatile uint32_t *)
(sleep_mode_env->wakeup_addr + 0x08)) =
SYSCTRL->DBG_LOCK_KEY[1];
*((volatile uint32_t *)
(sleep_mode_env->wakeup_addr + 0x0C)) =
SYSCTRL->DBG_LOCK_KEY[2];
*((volatile uint32_t *)
(sleep_mode_env->wakeup_addr + 0x10)) =
SYSCTRL->DBG_LOCK_KEY[3];
*((volatile uint32_t *)
(sleep_mode_env->wakeup_addr + 0x14)) =
sleep_mode_env->app_addr;
```

Wrong signed definition for channel assessment parameters	The fields of `chnlAsses_min_thr`, `chnlAsses_max_thr`, and `chnlAsses_noise_thr` are defined as unsigned byte variables (`uint8_t`), when they should be signed byte variables (`int8_t`) to match with the channel assessment parameters used in the Bluetooth Low Energy stack. To avoid the mismatch, users need to define the values in a way that can be aligned with the Bluetooth Low Energy stack. For example, to have a value -10, in a parameter passed in the function `Device_Param_Read()`, when `chnlAsses_param_src_type` is `APP_PROVIDED`, a value 0xF6 can be used. (Values should be between -128 (0x80) and 127 (0x7F).)
UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it must account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.
Only one DIO interrupt debounce filter can be enabled at a time	Only a single DIO interrupt debounce filter can be enabled at a time.
I2C CMSIS Driver xfer_pending feature 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.
The BootROM CRC is limited to 88 vector table entries	The CRC function of the BootROM application is not valid if more than 88 vectors are in use.
Controller privacy cannot be used with some smartphones	Smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used, due to a limitation of RSL10. 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	To have good audio quality for PCM input, add a delay in the `ASRC_out` DMA interrupt. Use a delay instruction in the

sample application for PCM input	`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.

12 issues

RSL10 CMSIS-Pack Release Notes – 3.3

April 23, 2020

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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- ON Semiconductor IDE
- Arm® Keil® uVision® IDE (tested with version V5.29.0.0)
- IAR™ Embedded Workbench® (tested with version 8.32.4)

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

Firmware Changes

Summary	Release Notes
Bluetooth Low Energy event priorities	Any Bluetooth Low Energy event has a priority set by default that can be found in `enum rwip_prio_dft` in file *rwip_config.h*. If an event cannot be programmed because of conflict or overlap with another higher-priority event, its priority is incremented to the default priority increment per event as found in `enum rwip_incr_dft`. Now the API has a new capability: `uint8_t BLE_Set_EventPriority(uint8_t eventIndex, uint8_t priorityValue, uint8_t incrementValue)` Priority `priorityValue` and increment value `incrementValue` can be set for an event where the index to its event is specified through `eventIndex`. The meaning of the index can be found in `enum rwip_prio_idx` defined in file *rwip_config.h*.
CGM profile	Added Continuous Glucose Monitoring (CGM) Bluetooth Low Energy profile with server and client/collector roles. Certification pending.

2 issues

Documentation Changes

Summary	Release Notes
Hardware Reference: Trim Values for Operating >85C for Automotive NCV-RSL10	Information added to the Important Note about retention regulator trim settings in Section 5.4.5.1 "Wakeup from Retention Memory in Sleep Mode" to specify the trim values for NCV automotive version of RSL10.
Firmware Reference: Order of Flash Unlock and Enable Instructions	Corrected the order of instructions in the example for the `Flash_EraseSector` function.

2 issues

Bug Fixes

Summary	Release Notes

Fix for SweynTooth Vulnerability	Fix for the Security Bypass for Zero LTK Installation (CVE-2019-19194), invalid connection parameters (channel map) and DHCheck Skip.
Radio Receive Failure with Frequent Switching Between Advertising and Scanning	An issue was fixed where the receive function could stop working when quickly switching between advertising and scanning mode.
Using XTAL 32KHz does not allow Sleep mode duration to exceed 825 seconds	When the low power clock source is XTAL 32KHz, `BLE_Sleep_MaxDuration_Set()` sleep duration was limited to 825s. However, using RC OSC or external clocks was fine. Now the limitation has been removed for XTAL 32KHz as well.
Stable modulation index is set incorrectly	The stable modulation index setting in the supported features list has been corrected to not supported. This issue could cause some phone models to sometimes violate IFS when sending a connection indication to establish a connection. As a result, the connection establishment failure rate could increase.
Support 2 Mbps Bluetooth Low Energy link in RMP coexistence for Bluetooth modulation index	When the modulation index used in the remote microphone streaming protocol is the value required to support Ezairo 7150 SL based streamers, the Bluetooth Low Energy link at 2 Mbps can keep the link connected in coexistence to the remote microphone protocol.

5 issues

Known Issues

The following issues are known to be outstanding in release 3.3:

Summary	Release Notes
Renew duration for PRA in host privacy is not limited correctly	In host privacy mode, the renew duration for a private resolvable address can be set from 1 s to 11.5 hours; however, the GAP API document indicates that the lower limit is 150 s.
JTAG can become unlocked when exiting sleep mode	If using the JTAG locking mechanism, with the current implementation of `Sys_PowerModes_Sleep_Init` and `Sys_PowerModes_Sleep_Init_2Mbps` in *rsl10_sys_power_modes.c* the device will become unlocked when returning from sleep mode. To fix this, the `Sys_PowerModes_Sleep_Init and Sys_PowerModes_Sleep_Init_2Mbps` routines in *rsl10_sys_power_modes.c* must have the following code added to them:

```
/* Write lock keys and wake-up restore address */
if (SYSCTRL->DBG_LOCK == DBG_ACCESS_UNLOCKED)
{
    *((volatile uint32_t *) (sleep_mode_env->wakeup_addr +
0x00)) = 0;
}
else
{
    *((volatile uint32_t *) (sleep_mode_env->wakeup_addr +
0x00)) = DBG_ACCESS_LOCK;
}
*((volatile uint32_t *) (sleep_mode_env->wakeup_addr +
0x04)) = SYSCTRL->DBG_LOCK_KEY[0];
*((volatile uint32_t *) (sleep_mode_env->wakeup_addr +
0x08)) = SYSCTRL->DBG_LOCK_KEY[1];
*((volatile uint32_t *) (sleep_mode_env->wakeup_addr +
0x0C)) = SYSCTRL->DBG_LOCK_KEY[2];
*((volatile uint32_t *) (sleep_mode_env->wakeup_addr +
0x10)) = SYSCTRL->DBG_LOCK_KEY[3];
*((volatile uint32_t *) (sleep_mode_env->wakeup_addr +
0x14)) = sleep_mode_env->app_addr;
```

Wrong signed definition for channel assessment parameters	The fields of `chnlAsses_min_thr`, `chnlAsses_max_thr`, and `chnlAsses_noise_thr` are defined as unsigned byte variables (`uint8_t`), when they should be signed byte variables (`int8_t`) to match with the channel assessment parameters used in the Bluetooth Low Energy stack. To avoid the mismatch, users need to define the values in a way that can be aligned with the Bluetooth Low Energy stack. For example, to have a value -10, in a parameter passed in the function `Device_Param_Read()`, when `chnlAsses_param_src_type` is `APP_PROVIDED`, a value 0xF6 can be used. (Values should be between -128 (0x80) and 127 (0x7F).)
UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it must account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.
Only one DIO interrupt debounce filter can be enabled at a time	Only a single DIO interrupt debounce filter can be enabled at a time.
I2C CMSIS Driver xfer_pending feature 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.
The BootROM CRC is limited to 88 vector table entries	The CRC function of the BootROM application is not valid if more than 88 vectors are in use.
Controller privacy cannot be used with some smartphones	Smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used, due to a limitation of RSL10. 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
```

10 issues

RSL10 CMSIS-Pack Release Notes – 3.2

December 10, 2019

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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- ON Semiconductor IDE
- Arm® Keil® uVision® IDE (tested with version V5.26.2.0)
- IAR™ Embedded Workbench® (tested with version 8.40.1)

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

Firmware Changes

Summary	Release Notes
CMSIS-Pack components for SEGGER RTT and PRINTF wrapper added	Added CMSIS-Pack components for SEGGER RTT and a PRINTF wrapper. The wrapper can redirect printf statements for both RTT and the UART CMSIS-Driver.
Ability to disable anchor point move request added	When a peer device starts a link layer control procedure when its timing and calculated bandwidth is not matched with the device timing, it sends a connection-parameters update with suggested anchor point movement values. Now the application has the option to disable this mechanism through the following API: `void BLE_Set_AnchorPointMoveReq(uint8_t anchorPoint_move)` If the input argument is zero, the feature will be disabled. If the input argument is set to one, the feature will be enabled. By default it is enabled. It can be called anytime dynamically, but after the `BLE_Initialize()` function.
Added updating of max Rx octet size dynamically	Using the `GAPC_SET_MAX_RX_SIZE_AND_TIME_CMD` command, an application can initiate a Bluetooth Low Energy update with a desired maximum Rx octet size and maximum Rx time. Earlier on, this was possible only for Tx, but now it can be used for Rx as well.
PWM driver and sample application added	Added a PWM Driver and a sample application called *pwm_driver*.
Timer driver and sample application	Added a Timer Driver and a sample application called *timer_driver*.

5 issues

Documentation Changes

Summary	Release Notes
RSL10 HW Ref Manual: VDDRF_SW and VDDSYN_SW Connection Recommendation	Extended recommendation of `VDDRF_SW` and `VDDSYN_SW` pads to now allow floating as well as connecting together.
RSL10 FW Ref Manual: Document Stack Support Functions	The *Firmware Reference* now contains documentation for the stack support functions called `BLE_Sleep_MaxDuration_Set`, `BLE_Sleep_ReductionTime_Set`, `BLE_Set_RxWinSize_Max`, `BLE_Set_RxWinSize_Disconnect`, `BLE_Set_AnchorPointMoveReg`, and `SecurityKeys_Read`. See the "Bluetooth Stack and Profiles" chapter for details.

2 issues

Bug Fixes

Summary	Release Notes
SPI configuration in RTE_Device.h is incorrectly parsed by Configuration Wizard	Disabling the SPI auto configuration setting using Config Wizard had been incorrectly disabling the whole SPI Configuration section; this has been fixed. Also, the auto configuration enable for each SPI peripheral has been separated in the SPI configuration and the settings of each peripheral have been grouped into an enable group.
GATTM_AddAttributeDatabase can get stuck in infinite loop	The `GATTM_AddAttributeDatabase` function had been getting stuck in an infinite loop when the custom attribute database contained more than 1 Bluetooth Low Energy service; this has been fixed.
Crash could occur in master role, during connection establishment	A memory crash (hard fault) sometimes occurred when establishing a Bluetooth Low Energy link at the reception of an advertisement in a master role. This has now been fixed.
stubprf.c uses incorrect symbol for the Proximity profile	Updated *stubprf.c* to use the correct symbol for the proximity profile.
RAM overlap in FOTA application	Fixed a RAM overlap issue in FOTA-related applications. This issue was capable of causing a FOTA update to fail, or a Bluetooth Low Energy application to advertise in a limited number of channels. This issue was introduced in SDK 3.1.
Typo in stubprf.c related to the Glucose Bluetooth profile	Updated the *stubprf.c* file to replace `CFG_PRF_GLPC` with `CFG_PRF_GLPS` so the profile can be used correctly.

6 issues

Known Issues

The following issues are known to be outstanding in release 3.2:

Summary	Release Notes
Wrong signed definition for channel assessment parameters	The fields of `chnlAsses_min_thr`, `chnlAsses_max_thr`, and `chnlAsses_noise_thr` are defined as unsigned byte variables (`uint8_t`), when they should be signed byte variables (`int8_t`) to match with the channel assessment parameters used in the Bluetooth Low Energy stack. To avoid the mismatch, users need to define the values in a way that can be aligned with the Bluetooth Low Energy stack. For example, to have a value -10, in a parameter passed in the function `Device_Param_Read()`, when `chnlAsses_param_src_type` is `APP_PROVIDED`, a value 0xF6 can be used. (Values should be between -128 (0x80) and 127 (0x7F).)
Support 2 Mbps BLE link in RMP coex for BT modulation index	When the modulation index used in the remote microphone streaming protocol is based on BT to support 7150 streamers, the Bluetooth Low Energy link at 2 Mbps can keep the link connected in coexistence to the remote microphone protocol.
UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it must account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.
Only one DIO interrupt debounce filter can be enabled at a time	Only a single DIO interrupt debounce filter can be enabled at a time.
I2C CMSIS Driver xfer_pending feature 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.
The BootROM CRC is limited to 88 vector table entries	The CRC function of the BootROM application is not valid if more than 88 vectors are in use.

Controller privacy cannot be used with some smartphones	Smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used, due to a limitation of RSL10. 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`

9 issues

RSL10 CMSIS-Pack Release Notes – 3.1

September 23, 2019

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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- ON Semiconductor IDE
- Arm® Keil® uVision® IDE (tested with version V5.26.2.0)
- IAR™ Embedded Workbench® (tested with version 8.32.3)

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

Firmware Changes

Summary	Release Notes
ASHA sample application updated for the official Android 10/Pixel 3 release	The ble_android_asha sample application has been updated with the latest changes in the specification made for the Android 10/Pixel 3 official release.
Removed unused functions and debug keys	Removed unused debug function and test key from stack.
Configuring a desired Rx window size can disconnect Bluetooth Low Energy links	Using the "BLE_Set_RxWinSizeDisconnect(uint32_t rx_win_size_disconnect)" API, applications can set a desired Rx window size in us, so that when the Rx window is widened up to a size equal to or greater than this value, the link is lost by the stack. By default this feature is disabled, and calling the API with a non-zero value causes the feature to be enabled. This API will be applied on all Bluetooth Low Energy links. This feature is checked in parallel to all Bluetooth Low Energy core conditions.
More descriptive error code returned when challenge response is read before it is calculated	Error code "insufficient authorization" was previously returned when an iOS device tried to read a challenge response attribute when the response was not yet calculated (not ready). The more accurately descriptive error code "Unlikely Error" is now returned.
Libraries containing source code now provided only as source code	libcmsis.a, libsyslib.a, libcalibratelib.a, libcustom_protocolLib.a, libflashlib.a, libmathlib.a, and libremote_micLib.a libraries have been removed from the CMSIS Pack and lib\Release folder. The libraries are now provided as source only.
High quality audio streaming with CELT codec	The remote_mic_rx_raw and remote_mic_tx_raw sample applications have been updated to support high quality audio steaming with the CELT codec.

6 issues

Documentation Changes

Summary	Release Notes
Updates to the documentation	The RSL10 Hardware and Firmware Reference Manuals have both been updated with minor clarifications. Minor updates have also been made to the RSL10 Firmware Over-the-Air User's Guide and the RSL10 LPDSP32 Support Manual.
RSL10 Getting Started Guide: ON Semiconductor IDE installation procedure	General documentation updates have been made to improve the ON Semiconductor IDE installation user experience.
RSL10 Getting Started Guide: Installing Arm CMSIS-Pack documented	While the example used in the RSL10 Getting Started Guide does not require any additional components or CMSIS-Packs, some of the other sample applications do. As a result, we have documented how to identify and resolve the missing components.

3 issues

Bug Fixes

Summary	Release Notes
SPI CMSIS-Driver: transfer complete event and busy flag behavior corrected	The "transfer complete" event was not being generated for one-way TX transfers, and the "busy" flag was incorrectly reported as a 2-bit value. This has been resolved.
Reduced the stack memory usage in the BondList_Add() function	The BondList_Add() function required a fixed 2 KB of stack memory for temporary use. This had the potential to cause a stack overflow. The function has now been optimized to allocate less memory and use the heap memory instead, returning false when not enough memory is available.
Default MTU size set correctly for all cases	Previously, when a secure connection was configured, by default the maximum MTU size was set to 65 bytes. However, in non-secure configurations the same value was also used, which was not correct. Now, using GAPM_SET_DEV_CONFIG_CMD, smaller values can be set. If a secure connection is used, for security management CID, the maximum expected MTU is set to 65; but for signalling CID, the default of 23 is now used. It is the application's responsibility to make sure that in a secure connection, a value equal to or greater than 65 is used for maximum MTU; otherwise, bonding fails.
Connection loss in peripheral_server_sleep resolved	The connection loss in the peripheral_server_sleep_ sample application with a connection interval of 25 ms or less has been resolved.
After connection parameter update, supervisor timeout is no longer double the expected value	After a connection parameter update, the supervision timeout was set to double the expected value; now the correct timeout is set.
Kernel timers now have critical section protection	`ke_timer_set` and `ke_timer_clear` are now re-entrant functions and can be called in an interrupt context without corrupting memory.
Rx window widening logic improved	Disconnect occurred when the Rx window was widened up to a quarter of a connection interval, but has now been corrected to disconnect at greater than half of the interval. As well, the Rx window is now checked for every connection event, and countervail is now applied in instant change procedures such as a connection parameter update. In addition, the Rx window can be widened up to 15.7 ms with 1 us resolution; after this window size, it is widened by a 625 us step size whenever the accumulated inaccuracy error reaches 625 us. Previously, it was widened up to 8 ms with resolution 1 us.
Corrected CRC in 2 Mbps when coexistence is used	When remote microphone protocol coexistence with Bluetooth Low Energy is used at 2 Mbps in Bluetooth Low Energy mode, the packets with sizes bigger than 27 bytes were being sent with the wrong CRC; this has now been corrected. To obtain this fix, use the updated remote microphone in remote_micLib.
I2C CMSIS driver now accepts multiple system clock frequencies	An issue was reintroduced in SDK 3.0 where the I2C CMSIS-Driver initialization had been hard coded to a specific system clock value. The initialization has been updated to accept multiple system clock frequencies to offer more flexibility.
All samples now compile with DBG_UART	When RSL10_DEBUG is set to DBG_UART or DBG_RTT in *app_trace.h*, some applications had been generating build errors; this has been fixed.
FOTA library and ble_peripheral_server_hrp_fota application now build when CMSIS and System components are set to "source" variant	When Device-Startup or Device-Libraries-System are set to the "source" instead of the "release" variant in *.rteconfig*, the FOTA library and *ble_peripheral_hrp_fota* application had generated some build errors because of multiple definitions. This is resolved, and only the "source" variant exists for system and startup CMSIS-Pack components now.
Bootloader application now builds when Startup and System components are set to "source" variant	When Device-Startup is set to the "source" instead of the "release" variant in *.rteconfig*, the Bootloader application was generating some build errors because of multiple definitions since this application uses a specific startup code. This is resolved, and only the "source" variant exists for the Startup CMSIS-Pack component.
Pops/clicks removed from remoteMic audio streaming	While listening to retransmitted packets in the remote microphone audio streaming protocol, a random click sound had been heard in the audio; this has been fixed.
PA back off bias now correctly applied	The sample applications have been updated to configure the TX power (using Sys_RFFE_SetTXPower() function) only after the radio has been initialized by the protocol in use (either Bluetooth Low Energy, Remote MIC, or Custom Protocol).
Removed duplicate linker script from sample apps	The linker script *sections.ld* is now added to the startup component in the CMSIS-Pack instead of duplicating it locally for each application. This helps you to pick up the linker script from *.rteconfig* and create sample applications from scratch.
IAR: calibration sample application correctly executes initialization function	The IAR sample applications *default_MANU_INFO_INIT* and *supplemental_calibrate* are now correctly executing the manufacturing initialization function (*Sys_Initialize*) from NVR3.

Fixed I2C bus error in I2C CMSIS-Driver with DMA Mode	An I2C Bus error event used to be returned in the I2C CMSIS-Driver sample application when DMA Mode was enabled. This has been corrected.
Correct BATMON IRQ Handler triggered in FOTA library	The FOTA library startup code has been updated to use the ADC_BATMON_IRQHandler, instead of the LSAD_BATMON_IRQHandler.
RTC count read is now atomic	The Sys_RTC_Value() function has been improved to account for cases when the RTC_CLK is clocked in the middle of a read. Reading is now atomic.
Fixed instability in peripheral_server_standby app when using pin wakeups	The instability in peripheral_server_standby sample application with DIO pin wake-up has been resolved.
Spurious emissions for the remote mic protocol: resolved	The PA back bias configuration has been corrected, so the remote microphone protocol implementation on the transmitter side no longer exhibits spurious emission violation non-conformance with regulatory tests.

21 issues

Known Issues

The following issues are known to be outstanding in release 3.1:

Summary	Release Notes
UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it must account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.
Only one DIO interrupt debounce filter can be enabled at a time	Only a single DIO interrupt debounce filter can be enabled at a time.
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.
The BootROM CRC is limited to 88 vector table entries	The CRC function of the BootROM application is not valid if more than 88 vectors are in use.
Controller privacy cannot be used with some smartphones	Smartphones Samsung S7 and S8 do not show RSL10 advertisements when controller privacy is used, due to a limitation of RSL10. 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`

7 issues

RSL10 CMSIS-Pack Release Notes – 3.0

July 4, 2019

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. The SDK is distributed through a CMSIS-Pack, which supports the following integrated development environments:

- ON Semiconductor IDE (Eclipse)
- Arm® Keil® Vision® IDE (tested with version V5.26.2.0)
- IAR™ Embedded Workbench® (tested with version 8.32.3)

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
CMSIS-Pack components for CMSIS-Drivers created	The CMSIS-Drivers have been converted into components, so the source code is no longer replicated in every sample application. In order to build applications that use the CMSIS-Drivers, the user needs to install the ARM.CMSIS pack.
CMSIS-CORE updated to version 5.5.1	In order to support FreeRTOS, the CMSIS-Core (cmsis-lib) header files have been updated to version 5.5.1.
Bluetooth Low Energy abstraction applied to peripheral_server_PRA sample project	The *peripheral_server_PRA* sample application has been refactored to use the Bluetooth Low Energy abstraction. The application is now named *ble_peripheral_server_PRA*.
Delay can now be applied to slave latency	By using the API BLE_DeviceParam_Set_SlaveLatencyDelay(uint8_t latencyDelay) after Bluetooth Low Energy initialization, an application can set a desired delay to the instant that slave latency is applied, after the time instant that is indicated in the connection indication or the connection parameter indication coming from a master device. In addition, when a non-empty packet is received or a non-empty packet is supposed to be sent, slave latency will be delayed for multiple connection intervals specified by latencyDelay. By default, slave latency delay is zero. Please note, this parameter is applied per device, not per link.
LPDSP32 applications updated to be CMSIS-Pack compatible	The RSL10/LPDSP32 sample projects (*Fibonacci* and *AudioCodecs*) have been updated to build using the RSL10 CMSIS-Pack.
Replaced the weak_prf library	The *weak_prf* library has been replaced with the *stubprf.c* file added to the Bluetooth Low Energy API component.
peripheral_server_sleep application now uses the non-light version of the stack by default	The sample application *peripheral_server_sleep* has been updated to use the non-light variant of the Bluetooth Low Energy stack by default (Debug and Release build configurations). Instructions on how to modify the project to use the light variant of the Bluetooth Low Energy stack have been added to the readme file. The light variant is no longer available through the build configurations because the *.rteconfig* file in the CMSIS-Pack is not synchronized with the build configuration.
FreeRTOS integrated into RSL10 CMSIS-Pack	Added FreeRTOS support with 3 new sample applications: *freertos_blinky*, *freertos_ble_peripheral_server_bond*, and *freertos_ble_central_client_bond*. In order to build these sample applications, the user needs to install the ARM.FreeRTOS CMSIS-Pack.
Bluetooth Low Energy abstraction applied to central_peripheral sample project	The *central_peripheral* sample application has been refactored to use the Bluetooth Low Energy abstraction. The application is now named *ble_central_peripheral*.
DMA Driver and sample application added to RSL10 CMSIS-Pack	A new DMA Driver peripheral abstraction and sample application have been added. This driver is re-used in all other CMSIS-Drivers to provide DMA support and avoid code duplication. The driver can be easily configured using the *RTE_Device.h* header file through the CMSIS Configuration Wizard.
DMA Support for I2C CMSIS-Driver added	DMA Support for the I2C CMSIS-Driver has been added. The *i2c_cmsis_driver* sample application has been updated to demonstrate the new feature. The DMA usage can be easily enabled or disabled in the sample application using the *RTE_Device.h* header file through the CMSIS Configuration Wizard.

DMA Support for SPI CMSIS-Driver added	DMA Support has been added for the SPI CMSIS-Driver. The *spi_cmsis_driver* sample application has been updated to demonstrate the new feature. The DMA usage can be easily enabled or disabled in the sample application using the *RTE_Device.h* header file through the CMSIS Configuration Wizard.
GPIO Driver now integrated into RSL10 CMSIS-Pack	A GPIO Driver peripheral abstraction and sample application have been added. The driver can be configured using the *RTE_Device.h* header file through the CMSIS Configuration Wizard GUI. See the *gpio_driver* sample application for more details.
Tx output power now consistent with datasheet	The target transmission power in the Sys_RFFE_SetTXPower function has been updated to be in the range [-17,+6] dBm, as specified by the datasheet.
DMA Support for UART CMSIS-Driver added	DMA Support has been added for the UART CMSIS-Driver. The *uart_cmsis_driver* sample application has been updated to demonstrate the new feature. The DMA usage can be easily enabled or disabled in the sample application using the *RTE_Device.h* header file through the CMSIS Configuration Wizard.
RSL10 Getting Started Guide: Migration to CMSIS-Pack added	RSL10 Getting Started Guide has been updated with instructions on how to migrate existing projects to CMSIS-Packs and vice-versa.
RSL10 HW Ref Manual: Improvements to ASRC Section	• Table 34: corrections for MODE1 (changed 0.4 to 0.84), and data added for MODE2. • added explanation about the different ASRC modes added before Table 34 • Figure 38: names for "_CNT" registers corrected by removing counters • paragraph beginning with "The audio sink clock being measured is sourced from" on page 386 put into an IMPORTANT table to stand out • register names DMA_CTRL0_SRC_ADDR_STEP_MODE and DMA_CTRL0_SRC_ADDR_STEP_SIZE in the Peripherals chapter corrected to SRC_ADDR_STEP_SIZE and SRC_ADDR_STEP_MODE, respectively
Fixed value for advDelay access is allowed	The advDelay parameter can be set to zero instead of a random value using the BLE_DeviceParam_Set_AdvDelay(0) function, allowing for an advertisement event to happen at a fixed interval. Note this is for applications that are not Bluetooth Low Energy compliant.
ON Semiconductor IDE: Sample project folder automatically refreshed after build	All the sample projects have been updated to have the refresh policy configured. Manually refreshing the projects after build is no longer necessary in the ON Semiconductor IDE (Eclipse).
Capability to increase slave latency in preferred slave latency feature added	The application now can send a request to the Bluetooth Low Energy stack to decrease and now also increase slave latency. Note that the application is responsible for saving the peer master device's slave latency to avoid asking for an increase that is more than the one set by the application.
RSL10 HW Reference Manual: added radio timing and PA power-up specifications	Documentation details regarding radio timing and PA power-up have been added to the RSL10 Hardware Reference, including calculation of timing from idle to Tx on/Rx on, delay values for PA ramp-up and ramp-down, and ramp-up steepness and duration.
CMSIS-Pack Manager plugin exception fixed	Fixed nullPointerException in the CMSIS-Pack manager plugin for the ON Semiconductor IDE (Eclipse). Now handling the exception when the list of local installed packs is empty.
Support for up to 8 simultaneous links in Bluetooth Low Energy stack	The number of possible simultaneous links has been increased to 8 for all Bluetooth Low Energy stack variants (expect the light stack which still supports only one connection). Note that applications using the non-light variant of the Bluetooth Low Energy stack need to use pre-processor CFG_CON=8.
Bluetooth Low Energy stack abstraction layers as a CMSIS-Pack component	The Bluetooth Low Energy stack abstraction layers have been converted into a CMSIS-Pack component making the code reusable across projects and no longer needs to be duplicated locally for each application.
RSL10 Getting Started Guide includes IAR instructions	RSL10 Getting Started Guide includes instructions to get started with the IAR development environment and the RLS10 IAR sample projects in the RSL10 CMSIS-Pack
Sample applications consolidated	The following sample applications have equivalent enhanced versions and have thus been removed from the RSL10 CMSIS-Pack: *I2C_master* (replaced by *i2c_cmsis_driver*) *I2C_slave* (replaced by *i2c_cmsis_driver*) *SPI_master* (replaced by *spi_cmsis_driver*) *SPI_slave* (replaced by *spi_cmsis_driver*) *simple_UART_DMA* (replaced by *uart_cmsis_driver*) *simple_UART* (replaced by *uart_cmsis_driver*) *PCM_master* (replaced by *sai_cmsis_driver*) *PCM_slave* (replaced by *sai_cmsis_driver*) *central_client* (replaced by *ble_central_client_bond*) *central_peripheral* (replaced by *ble_central_peripheral*) *peripheral_server_bond* (replaced by *ble_peripheral_server_bond*) *central_client_bond* (replaced by *ble_central_client_bond*) *peripheral_server_hrp* (replaced by *ble_peripheral_server_hrp*) *peripheral_server_pra* (replaced by *ble_peripheral_server_PRA*)

SAI Driver and sample application now integrated into RSL10 CMSIS-Pack	A new SAI (Serial Audio Interface) CMSIS-Driver and sample application been added to the CMSIS-Pack. The driver implements both PCM and I2S modes with optional DMA support.
Firmware and documentation unbundled from IDE installer	The RSL10 firmware is no longer bundled with the IDE installer. Firmware is now delivered exclusively as the CMSIS-Pack that can be imported into multiple supported IDEs.
RSL10 Getting Started Guide: Debug In Sleep Mode section added	RSL10 Getting Started Guide has been updated with instructions on how to effectively use the debugger with sleep mode.
ON Semiconductor IDE: Java runtime environment bundled	A compatible version of the Java runtime environment is bundled with the ON Semiconductor IDE, so Java no longer needs to be installed as a prerequisite.
Sys_RFFE_SetTXPower band edge emissions improvements	The power amplifier and bias configuration in the Sys_RFFE_SetTXPower function have been updated to improve band-edge emissions performance for target power values from 3 to 6 dBm.
ON Semiconductor IDE: CMSIS-Pack plugins customized	The Eclipse CMSIS-Packs plugins have been customized to: 1. Stop triggering automatic CMSIS-Pack index updates through the web 2. Have the pack root folder set by default to: C:\Users\<user_id>\ON_Semiconductor\PACK
RMP protocol works with Bluetooth Low Energy 2Mbps	Remote Mic Protocol was designed to work in coexistence with Bluetooth Low Energy at 1 Mbps but now also supports Bluetooth Low Energy link at 2 Mbps. The *remote_micLib* has been updated.
ON Semiconductor IDE: gnu-mcu-eclipse plugins updated	Upgraded the gnu-mcu-eclipse plugins in the ON Semiconductor IDE (Eclipse) to the latest version (4.5.1). This version includes multiple improvements and a new Peripheral Registers View. The *rsl10.svd* file is now configured in the debug session setup. The old *embsysreg* plugin has been removed. New instructions on how to use the new peripheral registers view are available in the *RSL10 Getting Started Guide*.
RSL10 FOTA User Guide: now includes signature validation	The *RSL10 Firmware Over the Air User's Guide* has been updated with more details regarding the signature validation procedure.
ON Semiconductor IDE: rebranded	The IDE has been re-branded as "ON Semiconductor IDE", as opposed to "RSL10 Development Tools".
Synchronization of non-connected stack activity	The application can configure the stack to receive an event called GAPM_ACT_START_IND when the controller indicates the commencement of a start advertising command, start scan command, or start connection command. To enable this event, set the `att_cfg` field in GAPM_SET_DEV_CONFIG_CMD. Note that the field `att_cfg` in struct `gapm_set_dev_config_cmd` has been renamed to `att_and_ext_cfg`.
Library linkage order issues and circular dependencies resolved	The linker expert settings in all sample projects for the ON Semiconductor IDE have been updated to make the GCC linker resolve circular dependencies. Therefore, specific library linkage order is no longer required.
sleep_RAM_retention sample application: more flexibility in selecting RTC alarm interval	The RTC configuration in the sleep_RAM_retention sample application has been updated to allow arbitrary alarm intervals.
Keil: added peripheral_server_sleep sample application	Added the peripheral_server_sleep sample application for the Keil environment.

40 issues

Bug Fixes

Summary	Release Notes
The stack answers the ATT protocol while the link is connected/disconnected	During link disconnection and reconnection, the number of available buffers between the controller and host was not communicated correctly, and this caused no buffers to be available for the L2CAP protocol. Now, at link disconnection, the controller informs the host that buffers are added, so the issue is resolved.
Incorrect RSSI value for connected mode resolved	It was reported that during scan mode, RSSI was in the expected range, while in connected mode for empty packets it was lower than expected. This issue has now been resolved.

GAPM_CANCEL_CMD description clarified	The description of the returned value for the "operation" field of GAPM_CMP_EVT has been clarified in the GAP interface specification.
Remote microphone protocol can no longer detect other transmitter packets	When the access word of a receiver in RMP had only one or two bits difference with a transmitter access word, it was possible for the receiver to detect the packets coming from this non-desired transmitter. This issue has been fixed.
Channel map is accepted	Resolved an issue which caused the channel map to be rejected in some situations while no link layer procedure was ongoing.
Behavior fixed when Bluetooth Low Energy link switches between connected and not-connected mode quickly	If the Bluetooth Low Energy link was connected and lost continuously and quickly (for example, by toggling Bluetooth on iPhones (enabling and disabling), or when the device was at the boundary of the RF link budget), the Bluetooth Low Energy stack sometimes went into a state that could not respond to the attribute protocol. This issue has been resolved.
sbrk function no longer increases heap memory by four times	A bug in the startup code, which casued the heap memory allocated to be increased by four times due to an incorrect typecasting, has been fixed. The bug was introduced in SDK 2.3.
RW-BLE-GATT-IS.pdf document updated to avoid out of order packet transmission	The specification for GATTC_CMP_EVT in the RivieraWaves GATT Interface Specification has been clarified, to prevent users from sending GATT messages (from the application to the stack) that might cause out of order packets over the air. A recommendation has been added to the document.
peripheral_server_standby: the LED now blinks to show BLE ADV/CONN is active	The LED connected to DIO6 on the Evaluation and Development Board are on and off during the active mode and standby mode, respectively. Therefore the LED now flashes at a slower or faster rate when the advertising interval (or connection interval) is decreased or increased (respectively).
Pairing with a Bluetooth Low Energy 5 device as second connection is now possible	When RSL10 was connected to a master device and then tried to connect to another central Bluetooth Low Energy 5 device and pair (smart devices such as Pixel and Samsung), the link was lost. This has been fixed.
Clarified SetTXPower function error for 0dBm and VCC_TARGET = 1.10	The call to Sys_RFFE_SetTXPower in the *peripheral_server_sleep* and *peripheral_server_standby* applications returns an error indicating that the configured VCC_TARGET may not be enough to provide the required TX power. This is expected, because in these applications we are demonstrating low power consumption by running with VCC configured to 1.10 V. Comments have been added to the applications to clarify the behavior, and the function error codes have been added to the *RSL10 Firmware Reference* manual.
ON Semiconductor IDE: Eclipse plugins check for updates has been disabled	The IDE has been updated to disable Eclipse plugins update requests by default.
Non XTAL-32 KHz-based low power clock sources can support any advertising interval/sleep internal	If a non XTAL-32KHz-based low power clock is used as the low power clock source, sleep can now be any duration. Wakeup was irregular because of an issue in the Bluetooth Low Energy stack. Now, applications can use the kernel timer to manage their timing. When there are no Bluetooth Low Energy activities, the stack goes to sleep following a kernel timer programmed by the applications. Alternatively, an application can use the `BLE_Sleep_MaxDuration_Set()` API function to set the sleep duration time before calling the `BLE_Power_Mode_Enter()` function if no BLE activity or Kernel timer is ongoing..
Sys_ASRC_Enable function updated	The Sys_ASRC_Enable function has been updated to allow enabling and disabling the ASRC
Stack Memory Corruption	Prevented a possible race conditions that could lead to memory corruption resulting in a hard fault interrupt.
Keil: calibration sample applications updated to execute initialization function	The Keil sample applications default_MANU_INFO_INIT and supplemental_calibrate have been updated to properly execute the manufacturing initialization function in NVR3.

16 issues

Known Issues

The following issues are known to be outstanding in release 3.0:

Summary	Release Notes
IAR: calibration sample applications fail to run initialization function	The IAR sample applications default_MANU_INFO_INIT and supplemental_calibrate fail to execute the manufacturing initialization function (Sys_Initialize) from NVR3.
I2C Bus error event returned in I2C CMSIS-Driver DMA Mode	I2C Bus error event returned in the I2C CMSIS-Driver sample application only when DMA Mode is enabled.

UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it should account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.
Instability in peripheral_server_sleep and peripheral_server_standy app when using pin wakeups	The peripheral_server_sleep and peripheral_server_standby applications may become unstable if wakeup by DIO pins 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.
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 more than 88 vector table entries	The CRC function is not valid if more than 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`

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.
ON Semiconductor IDE: Installations can fails if path name has too many characters	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.

12 issues

RSL10 SDK Release Notes - 2.4.1

March 25, 2019

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.

> **NOTE**
> The CMSIS Pack has been updated to version 2.4.1 which includes the new and improved FOTA features. There is no need to download and run the SDK installer as it remains at version 2.3.

What's New

Summary	Release Notes
New sample application for Heart Rate Peripheral FOTA	This new application (*ble_peripheral_server_hrp_fota*) is similar to the Heart Rate Bluetooth Low Energy application (*ble_peripheral_server_hrp*) but with added features to support FOTA updates.
Add new FOTA command line tool into RSL10 SDK	Replaced the previous Firmware Over-The-Air (FOTA) PC application with a new tool, named *FOTA.Console.exe*. This new tool is available in the *utility_apps.zip* package and provides support for FOTA updates using the RSL10 Dongle.
Getting Started Guide: Add Printf Debug Capabilities	An advanced Debugging section is added to the RSL10 Getting Started Guide covering PRINTF() debugging using UART and RTT.
Update Bluetooth Low Energy abstraction callback function return type to allow error propagation	Updated the return type of the Bluetooth Low Energy GATT abstraction callback function from void to uint8_t. This allows the user application to return an error during an attribute read or write. The error code is propagated to the Bluetooth Low Energy stack and peer device.
Update bootloader sample application for compatibility with FOTA	The bootloader sample application has been updated to be compatible with the Firmware Over-The-Air (FOTA) Device Firmware Update (DFU) component. Documentation is provided in the *RSL10 Bootloader Guide*.
RSL10 FOTA documentation	Added a new manual *RSL10 Firmware Over-The-Air User's Guide*. This document provides information on how to use the RSL10 tools, firmware and sample code to perform FOTA updates.

6 issues

Bug Fixes

Summary	Release Notes
Corrected the CMSIS Pack Bluetooth Low Energy applications' light/hci library variant	Bluetooth Low Energy applications in the CMSIS Pack had the incorrect library variant set by default in the *.rteconfig* file (release_light or release_hci). This bug was introduced in SDK 2.3 and has now been fixed.
Sample application ble_android_asha restored in the CMSIS Pack	The ble_android_asha sample application was released in the RSL10 Standalone SDK 2.3 but not included in the CMSIS Pack. This application is now also available in the CMSIS Pack.

2 issues

Known Issues

The following issues are known to be outstanding in release 2.4.1:

Summary	Release Notes

UART CMSIS-Driver clears busy status while last byte transfer is still ongoing	The UART CMSIS-Driver busy status flag is cleared once the hardware is ready to transmit or receive additional data. Therefore, the last byte transmission is still ongoing once the flag is cleared. If the application needs to know when the transfer has finished, it should account for additional time to complete the transmission of the last byte after the busy flag is cleared, based on the baud rate.
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 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.

15 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 SDK Getting Started Guide	Second document you should read after this one. For a full description, see Further Reading
RSL10 Software Development Kit (SDK) Installer	Installer for the standard RSL10 SDK (Oxygen Eclipse)
RSL10 SDK CMSIS Pack	Arm CMSIS Pack for Oxygen and Keil
RSL10 Software Documentation Package	All manuals in one folder that can be accessed without installing or starting an IDE
RSL10 Software Utility Apps	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 ■ Firmware Over-The-Air (FOTA) Console Tool
RSL10 SDK LPDSP32 Package	Startup files, Eclipse project files, a utility to integrate LPDSP32 output code with Arm Cortex®-M3 processor code, a test suite, and documentation
RSL10 Bluetooth Mesh Package	The tools and documentation that support Mesh networking
RSL10 Software Release Notes History	An archive file containing previous versions' release notes

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.

RSL10 SDK Release Notes - 2.2

October 2, 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
Integrate AES-128 ECB in Remote Microphone applications	AES-128 ECB as an end-to-end encryption, implemented in the Arm Cortex-M3 processor, is now available for the remote microphone (non-coexisting) application.
Setting maximum sleep duration	The default maximum sleep duration has increased to 30 s. In addition, a desired maximum duration time can be set after the `BLE_Initialization()` function by using the `BLE_Sleep_MaxDuration_Set()` function. To reduce the sleep duration, use the function `BLE_Sleep_ReductionTime_set()`. It is zero by default. Both sleep functions accept input arguments in units of 625 us. Do not set the maximum sleep duration to a value where its equivalent number of low power clocks overflows a 32-bit limit.
Adding API to limit Rx window size	Using the `BLE_Set_RxWinSize_Max(uint32_t max_rxWin, uint8_t instant_change_include)` API after `BLE_initialize`, you can set the maximum Rx window size to avoid consuming more power in the case of a poor radio link budget. The default follows the Bluetooth Low Energy standard, so calling this API is usually unnecessary. It is a dynamic API that can be called any time in an application, but it affects all Bluetooth Low Energy links connected to the device. Input arguments are: • max_rxWin: in micro seconds, a zero input means an invalid parameter and it follows default behavior. • instant_change_include: when set to zero, this mechanism is not applied during connection establishment or parameter update procedures; otherwise it is applied in all Bluetooth Low Energy link steps.
Apply Bluetooth Low Energy abstraction to peripheral_server_hrp application	New sample application: ble_peripheral_server_hrp demonstrates how to implement the Heart Rate Sensor portion of the Heart Rate Service, including advertising, pairing, bonding and whitelisting. It is a refactored version of the peripheral_server_hrp sample application.
ble_central_client_scan sample application	New sample application: ble_central_client_scan demonstrates a Bluetooth scanning procedure using the Bluetooth Low Energy abstraction and using the UART CMSIS-driver for input/output.

5 issues

Bug Fixes

Summary	Release Notes
Peripheral Sleep/Standby now sleeping when external clock is used	Fixed issue where the external clock, `RTC_CLK_SRC_DIO`, in *peripheral_server_sleep* and *peripheral_server_standby* was not working for SDK 2.1 or 2.2. Previously, the signal would not go to sleep at all, and would only advertise once. It would also not be possible to connect while the external clock was used.
Negative TX power in Supplemental Calib app	A button press is added in the supplemental_calib and default_manu_info applications to avoid over-writing into the NVR3 memory.
Revised the Bluetooth stack	Improvements: • 2MBPS feature mask issue • Slave MD bit management updated to speed the acknowledgement • Sync window management updated for all procedures (connection, update, connected) • Parameter request procedure updated if peer does not support it

	• Bonding management updated • Updated link loss detection because of slave latency • Test mode initialization and buffer management updated • Priority update function modified • Security manager updated • L2CAP buffer management and fragment updated • Prevent having a software AES request during the data RX which lead to a TX Crypt error • EDCH - Fixed Co-ordinate invalid Curve attack
Receiving Bluetooth Low Energy packets when remote microphone and Bluetooth Low Energy are used together	When a remote microphone protocol event is supposed to be programmed right before a Bluetooth Low Energy event, and the Bluetooth Low Energy event gets preemption, depending on its state and timing, the Bluetooth Low Energy packet might not be received. The required modification has been applied on the remote microphone protocol to address this issue.
Updating connection parameters with new iPhones	iPhones 8 and X that support LLCP parameter updates initiated by slave devices did not accept these parameters from RSL10. Now the stack sets the parameters of this message in a way that iPhone accepts. It is especially important when MFi is used.
Standby application now works at 2 Mbps	The standby application was supporting only 1 Mbps. Now the required changes have been applied on syslib to save and restore the 2 Mbps configuration, allowing the standby application to work at 2 Mbps.
Upgraded the Bluetooth Low Energy core version	Previous SDKs used Bluetooth Low Energy 4.2. Since RSL10 has been certified for Bluetooth Low Energy core specification 5.0, that specification has been applied to SDK 2.2.
Stack no longer crashes when receiving multiple characteristic handle/UUID pairs in a Read By Type Response	When Data Length Extension (DLE) was enabled and a central device discovered characteristics of a peripheral with a large attribute database (at least 25 attributes with a 16-byte UUID), the central device used to crash. This is fixed.
BLE Mesh flash issues	Fixed a BAD_ADDRESS error reported when using the Flash_EraseSector function in the ble_mesh application.
CMSIS-Driver clock restrictions removed	The I2C CMSIS-Driver has been updated to allow unprecise prescale clock values when it's impossible to achieve an exact ratio between the system clock and the desired I2C bus speed.
Further investigation into a crash issue on Mesh	The stack used to try to delete an event that was deleted already, causing a memory crash when Mesh was using advertising and scanning. It has been fixed.
Scrambled Bluetooth address using PRA is no longer the same value after Reset	Updated the *peripheral_server_PRA* sample application to generate a different random address once the device is power cycled.

Removed Sys_ASRC_Enable function	The `Sys_ASRC_Enable` function did 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;`

13 issues

Migrating from RSL10 SDK 2.1 to 2.2

If you use Release 2.1 code or projects in Release 2.2, remember the following:

- Compatibility of the Remote Microphone protocol is not guaranteed between SDK 2.1 and SDK 2.2. In other words, an application compiled as TX in SDK 2.1 will not necessarily work with an RX application compiled in SDK 2.2.
- Although the Release 2.1 sample code builds successfully in Release 2.2, backwards compatibility is not guaranteed.

Known Issues

The following issues are known to be outstanding in release 2.2:

Summary	Release Notes
Sleep App: Connection drop when using RC_OSC and an external clock	A bug exists with the *peripheral_server_sleep* app, in which the connection drops between any central server and an RSL10 board if an external clock or the `RTC_CLK_SRC_RC_OSC` is used with `LOW_POWER_CLK_UPDATE` disabled. As a temporary fix to avoid a dropped connection, enable `LOW_POWER_CLK_UPDATE` in *app.h*.
Sync_pulse DIO initial for remote_mic_coded app	Initialize `DIO_SYNC_PULSE` as a `GPIO_OUT` in the *remote_mic_trx_coded* app. `Sys_DIO_Config(DIO_SYNC_PULSE, DIO_MODE_GPIO_OUT_0);`
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 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.

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.

13 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.RSL10.2.2.0.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.

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.

RSL10 SDK Release Notes - 2.0

April 11, 2018

> **Only these Release Notes have changed**
> We added two known issues; see information about the Java version and CEVA GAP documentation below.

RSL10 is a Bluetooth® 5 certified multi-protocol radio 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

This SDK installation provides Oxygen Eclipse, a more recent IDE than the Kepler Eclipse previously provided. It includes an upgrade to version 6.3.1 of the GNU Arm® Embedded Toolchain. Besides providing the benefits of a newer Eclipse version, this installation has the added benefit of a seamless migration of code that was written for RSL10 SDK Release 1.4.

Furthermore, this release provides an Arm CMSIS-Pack for Oxygen that you can add to the SDK installation if you are accustomed to working in a CMSIS-Pack environment.

Summary	Release Notes
Replace the Evaluation and Development Board manual with a link to the one on the public website	The Evaluation and Development Board manual is available from www.onsemi.com. For your convenience, a PDF with a link to the manual has been placed in the *documentation* folder and the *Documentation.zip* file.
Changed "Evaluation and Development Kit" to "Software Development Kit"	The software product installed for RSL10 support has changed its name from "Evaluation and Development Kit (EDK)" to "Software Development Kit (SDK)" to be consistent with the industry practice of naming these types of software development tools.
C startup logic can be accelerated by using the Flash copier block & DMA Channel	*start.c* has been changed to use the Flash Copier and DMA to consume less energy to copy the data memory, and to reset the BSS memory section in the startup.
[CMSIS-Driver] Implementation of I2C Driver and sample application	Added an I2C CMSIS-Driver and sample application in the *source/samples/i2c_cmsis_driver* folder. See the *RSL10 Sample Code User's Guide* for more information.
[CMSIS-Driver] Implementation of SPI Driver and sample application	Added a CMSIS-Driver and sample application for the SPI interface in the *source/samples/spi_cmsis_driver* folder. See the *RSL10 Sample Code User's Guide* for more information.
[CMSIS-Driver] Implementation of UART Driver and sample application	Added a CMSIS-Driver and a sample application for the RSL10 UART in the *source/samples/uart_cmsis_driver* folde r. See the *RSL10 Sample Code User's Guide* for more information.
Improved ability for iPhones to reconnect	The stack accepts the termination reason of "CONNECTION LIMIT EXCEEDED" from an application.
Indicating when an LEA event is not programmed	When a secondary LEA event cannot be programmed in scatter net use cases, then an LEA audio IRQ is raised to inform the application, but not with valid audio. The IRQ happens earlier than the time of the expected secondary event but after the primary event.

8 issues

Bug Fixes

Summary	Release Notes
Fixes applied to the Bluetooth low energy applications	• In peripheral_server_bond, NVR2 is no longer being erased at beginning of the code. • TX power is now being set in hci_app. • In sleep and standby applications, the clock accuracy has been corrected from 20 ppm to 500 ppm
Fixes applied to the Bluetooth low	• The RX window in connection parameter updates was opening too early and now starts at the right time. It saves on

energy stack	power consumption. • During connection establishment, if the first anchor points were missed, connection establishment was not possible; now the connection is established. • The RX window now widens when a 20 ppm clock is used. • During numeric comparison pairing, the application was pairing successfully although it did not send verification confirmation from the user. Now it fails appropriately. • The scan duration is now consistent with the GAP documentation, changing from 7s to 10s. • In advertisement indications, RSSI is now reported as a signed value. • The RX window was widened too much during the connection establishment and connection parameter update. Now it conforms to the specification. • Using DTM in an application is stable, and commands can be received/sent from/to a tester through a communication interface such as I2C. • The heap memory of kernel messaging and the attribute database has been increased to allow adding more attributes in one service. • The handling of non-32768 Hz low power clock in the stack is modified to resolve the problem of randomly missing events. • The application can now replace the sum of clock accuracy of the slave and master devices of a BLE link with its own desired value. It helps to cope with variations of RC oscillator clock accuracy.
remote_mic_coex application audio and advertising are fixed	*remote_mic_rx_coex* used to be interrupted when audio was sent from *remote_mic_tx_coex*. Now the receiver continues to receive data. Also, when the device is advertising and listening to an audio stream, advertisement does not stop.
Signal-to-Noise Ratio improved for the ASRC	The SNR performance of the ASRC when the input signal is above 0.5 full scale has been improved by using mode-1 instead of mode-2.
Sys_RFFE_SetTXPower() now outputs the correct RF TX power	Modified the Sys_RFFE_SetTXPower function to drive more accurate TX power output if a calibrated RSL10 is used.
The SPI sample code can now successfully do the comparison operation	The system clock on the SPI slave side has been adjusted so that it can catch the data stream to do the comparison operation.
The mechanism to generate seed point of srand for Bluetooth low energy is truly random	The seed point for the *rand* function of the stack can be set by the application to be able to generate a truly random key.
Fixed the multi-link simultaneous connection failure with the central_client_bond Release configuration	The *central_client_bond* application no longer fails to connect to devices when multiple devices try to connect and bond to it at the same time. In previous releases, for the release build configuration, bonding had to complete before a second device could attempt to bond to the application.
Remote Microphone Transmit: fixed the Release version to finish the LPDSP32 initialization	The Release version of the *remote_mic_tx_raw* sample application is now functional. Replacing the linker script file is no longer necessary.
Connection parameters fixed for remote microphone RX coex	When *remote_mic_rx_coex* was listening in audio stream mode and it had a Bluetooth connection, Bluetooth connection parameter updates were not possible. Also, if the Bluetooth link disconnected, it could connect to some master devices but not all of them while it was in audio listening mode. Now, it can connect to both Android phones and iOS devices even while in audio listening mode.
MASS_ERASE call removed from standalone flash loader	The standalone and open flash loaders were updated to avoid inconsistent behavior when flash redundancy sectors are in use.
EraseAll/VerifyAll fixed for standalone flash loader	The VerifyAll operation of the standalone flash loader has been updated to match the EraseAll operation. From now on, both operations verify only the main area of flash.
Direct Test Mode (DTM) script now calculating packet errors correctly and is using the physical radio channel number	The DTM script has been changed to calculate PER correctly for 2 Mbps. Also the channels numbers that are entered in a DTM script are expected to be radio physical channel numbers, and the DTM script does not apply any conversion to the number when it sends a DTM command.

13 issues

Migrating from RSL10 SDK 1.4 to 2.0

You do not have to make any changes to your Release 1.4 code or projects to work with them in SDK 2.0.

Known Issues

The following issues are known to be outstanding in release 2.0

Summary	Release Notes

SDK does not launch with Java SE 10	If you have Java SE 10 installed, the Oxygen Eclipse-based IDE will not work. You must install Java version 9 or 8.
Adjust the Sys_RFFE_SetTXPower function for uncalibrated boards	Transmission power is not properly generated for 1dBm to 6 dBm for uncalibrated boards.
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.
Wrong identity address type in pairing procedure	When controller privacy is used, address identity type is not set correctly, causing pairing failure for iPhone and some versions of Android.
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 PCM port in the Stereo Raw Audio Stream Broadcast Transmitter Custom Protocol sample application (*remote_mic_tx_raw*) causes periodic noise on the right channel.
Investigate increase in latency for custom_protocol_trx	Configuring the RX output to SPI will cause 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.
CEVA GAP documentation updated incorrectly	When the GAP (RW-BLE-GAP-IS_2mbps) interface specification documentation was updated, it described a DTM API that is not available in this release.
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" in the *Documentation.zip* archive and the installed *documentation* folder 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

Release Item	Description
RSL10_SDK.zip	Contains the Getting Started guide, installer for the standard RSL10 IDE (Oxygen Eclipse), and the 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