RSL10 EDK Release Notes - 1.4

December 21, 2017

RSL10 is an ultra-low-power, multi-protocol 2.4 GHz radio, designed for use in wireless devices that demand low power consumption and a restricted size. The RSL10 Evaluation and Development Kit (EDK) includes the hardware, 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

Bluetooth Security and Addressing Enhancements

The Bluetoooth-based sample applications have been updated with some software enhancements to better support Bluetooth secure connections and alternate addressing modes for privacy. This includes loading of pre-calculated ECDH private-public key pairs that are needed to bond in this Bluetooth security mode, which if calculated at run time by the application can add several seconds to the application initialization.

To support these enhancements the following firmware items have been added:

- The device info record (NVR3) now has locations reserved for an identity resolving key (IRK), a connection signature resolving key (CSRK), and the ECDH keys
- A Bluetooth configuration support function (Device_Param_Read()) has been added to the system library, that uses an application-defined support function (Device_Param_Prepare()) to efficiently and correctly configure connection parameters, addresses and keys for the Bluetooth stack.
- Enhancements to the system initialization applications (*default_MANU_INFO_INIT, supplemental_calibrate*) to create entries for the IRK, CSRK, and ECDH keys if those values are unavailable, and to save and restore those entries if they are available.

For more information on integrating these enhancements into your user application, see Migrating from RSL10 EDK 1.3 to 1.4.

Notable Enhancements

Bluetooth Stack Enhancements

The following features were added to the Bluetooth low energy stack:

- You can establish a connection with the Transmit window offset equal to zero.
- The light stack supports secure connections.
- An application can put RSL10 in Direct Test Mode (DTM), and it can switch between Bluetooth low energy mode and DTM without sending a reset or doing any initialization.
- The address identity type is set accordingly in the pairing procedure (it was set to public in the previous release).
- The advertisement inter frame space can be set from an application. Using lower values helps to decrease power consumption, especially for sleep and standby apps.

Set RF Transmit Power

All the Bluetooth applications now set their RF transmit power using the `Sys_RFFE_SetTXPower()` function in app_init.c (after RF initialization). The set power function only requires the target power (in dBm) as its parameter and it can be used to set the power in the range of -20dBm to +6dBm with a tolerance of +/- 1dBm.

LED Behavior

All Bluetooth applications now have a LED_Timer function in *app_process.c* to allow easy access to LED functionality for each application. The default LED behaviour in most applications is to blink while connecting and become solid when connected. To show multi-link status, the LED blinks once (periodically) for each connection maintained until all expected devices are connected. The bonding applications are slightly different and have varied blinking rates for when the application is connecting or bonding.

Bonding Applications

All bonding applications such as *peripheral_server_bond* and *peripheral_server_hrp* are able to resolve resolvable addresses now. They have also been updated with refactored bondlist functions which are now consistent with all bonding applications in the sample code provided.

Queued Writes in the Central Client Application

The *central_client* application now optionally supports queued writes. The example queued write operation sends two prepared writes requests and one execute write request. The prepared write requests queue up data to be written to attributes on the slave device, which are then only applied to the attributes once the execute write command is received.

Hardware Resolution of Private Addresses in the Peripheral-Server Heart Rate Profile Application

A resolving list is now supported in the *peripheral_server_hrp* application. The resolvable addresses of peer devices can be resolved by a hardware accelerated resolving list, which makes address resolution simpler and can reduce power consumption.

Improvements to the Low-Power Mode Applications

The known issues with the RC oscillator measurement implementation have been fixed, and the implementation has been optimized to use less power and memory. A small number of measurements are taken at periodic intervals, and filtered using a simple low-pass filter implementation. This ensures that the RC oscillator's frequency is measured accurately enough to maintain a Bluetooth connection.

Improvements to the Remote Microphone Applications

The code for all *remote_mic* applications has been refactored and cleaned up. A DMIC implementation was added to the *remote_mic_tx_raw* appl ication.

Other Enhancements

Release Notes
Audio output from the Audio Stream Broadcast (Remote Microphone) custom protocol can now be routed to the output driver integrated into RSL10.
The *Evaluation and Development Board Manual* has been removed from the EDK documentation set and is available directly from the ON Semiconductor website: www.onsemi.com.
New sample applications to demonstrate remote microphone functionality have been added: Mono Raw Audio Stream Broadcast Receiver Custom Protocol sample application (*remote_mic_rx_raw*), Stereo Raw Audio Stream Broadcast Transmitter Custom Protocol sample application (*remote_mic_tx_raw*), and Coded Audio Stream Broadcast Transmitter/ Receiver Custom Protocol sample application (*remote_mic_trx_coded*). The *remote_mic_tx_raw* sample application is structured differently from the others because it uses the LPDSP32 framework. See the readmes that accompany each sample for more information.
Both the Bluetooth stack initialization function and the Sys_RFFE_SetTXPower function configure the power provided to the RF front-end's power amplifier (PA). This conflict results in the Bluetooth stack initialization corrupting the settings selected using the Sys_RFFE_SetTXPower function when attempting to configure the device for a PA power level between -20 dBm and -1 dBm. This is only an issue when the Bluetooth stack is initialized after the Sys_RFFE_SetTXPower function is called. When configuring the output PA power to -20 dBm or -19 dBm, the expected error might be beyond the +/- 1 dBm expected for all other PA power configurations.
A new sample application called Private Resolvable Address with Peripheral Device (peripheral_server_PRA) is available. It demonstrates the use of either a public or private address. If the private address type is selected, a randomly generated resolvable private address is used. For more information, see the readme that accompanies the sample application.
GAP (RW-BLE-GAP-IS_2mbps) and GATT (RW-BLE-GATT-IS) interface specification documentation is updated. However, the DTM API that is described in the GAP document is not available in this release.
The *central_client* application and derived applications have been updated to merge in support for multiple connections, previously provided by the *central_client_multi* sample code. All updated applications now support forming connections (both secure and unsecure) with up to four peer devices simultaneously.
The *peripheral_server* application and derived applications have been updated to add support for multiple connections. All updated applications now support forming connections (both secure and unsecure) with up to four peer devices simultaneously.
In the *Hardware Reference*, clarified how to use the user redundancy sectors to patch up to two sectors of the flash memory.
The Audio Stream Broadcast custom protocol sample applications include a PCM input which receives the audio stream in 32 kHz sampling frequency, and needs to down-sample to 16 kHz to feed the G.722 codec and the broadcasting protocol. This effect used to be achieved with a simple filtering scheme that dropped every second sample. Now, filtering is provided by using the asynchronous sample rate conversion (ASRC) block.

10 issues

Migrating from RSL10 EDK 1.3 to 1.4

Using the Bluetooth security and addressing enhancements:

To use the Bluetooth security and addressing enhancements described in Bluetooth security and addressing enhancements, user applications need to override the weakly defined `Device_Param_Prepare()` function. Example implementations of this function are provided with all Bluetooth applications that support encryption and bonding, and these can be added to a user application using the following steps:

1. Copy the function from the new application's *ble_std.c* file.
2. Update the `BLE_Initialize` function in *ble_std.c*. Everything but task state initialization (**/* Initialize task state for all peers */**) can be copied as it is. Remove the for loop and array indices for this section.
3. Add the rsl10_protocol include to *app.h* (**#include <rsl10_protocol.h>**).
4. Add all the new symbols from the *ble_std.h* and *ble_sec.h* files.

Required Changes to Sample Code between EDK 1.3 and 1.4

Code written for EDK 1.3 can be compiled and used under EDK 1.4, with the following changes applied.

Peripheral_server_sleep

The `BOOT_ON_RAM` symbol in the `sleep_mode_env` configuration has been renamed.

- For the `sleep_mode_env->wakeup_ctrl` configuration in the `Sleep_Mode_Configure` function, rename the `BOOT_ON_RAM` symbol to `BOOT_CUSTOM`.

Sleep_RAM_retention

The `BOOT_XTAL_ENABLE` symbol in the `sleep_mode_flash_env` configuration has been renamed.

- For the `sleep_mode_flash_env->wakeup_ctrl` configuration in the `Sleep_Mode_Configure` function, rename the `BOOT_XTAL_ENABLE` symbol to `BOOT_FLASH_XTAL_DEFAULT_TRIM`.

Custom_protocol_trx

The variable `stepSize` in structs `hop_status` and `cp_param_tag` has been removed.

- Remove `app_env.cp_param.stepSize = 0; /* for fast hopping */` in function `APP_CPInit(void)`.

Known Issues

The following issues are known to be outstanding in release 1.4

Summary	Release Notes
Multi-link simultaneous connection failure with central_client_bond Release configuration	The *central_client_bond* application fails to connect when multiple devices try to connect and bond to it at the same time. This issue is seen in the Release build configuration of the application, but there is a workaround. If only one device is connected and bonded to the application at a time, waiting about 5-10 seconds after the connection (to confirm that the bonding process is finished), the application can maintain multi-link connections and bonds.
Remote Microphone Transmit: the Release version does not finish the LPDSP32 initialization	The Release version of the *remote_mic_tx_raw* sample application is not functional. To make it work, replace the linker script file with: `"${workspace_loc:/${ProjName}/linker.lcf}"`

	in **Property> Cross ARM C Linker> General > Script file**.

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); // TODO`
	`#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.
Remote mic RX coex cannot change connection param and connect to some devices	When remote_mic_rx_coex is in audio stream mode listening and it has a Bluetooth connection, Bluetooth connection parameters update is not possible. Also if the Bluetooth link disconnects, it can connect to some master devices but not all of them while it is in audio listening mode. In this situation, it can connect to some third-party devices, but not to Android phones and sometimes not to iOS devices.
Remove MASS_ERASE call from standalone flash loader	The flash mass erase functionality used in the stand-alone flash loader is not updated for the latest recommendations in the Hardware Reference Manual. This will cause the erase to not succeed if flash redundancy sectors are in use on the device. The recommended workaround is to not use the “Erase all of flash before programming” option on the Program tab of the flash loader GUI. Instead, in the Tools tab of the GUI use the Region box to specify all of main flash (0x00100000-0x0015FFFF) and click the Erase button to erase only the main flash sector-by-sector.
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.
Direct Test Mode (DTM) script not calculating packet errors correctly	The Direct Test Mode Script is not calculating packet errors correctly for the 2 Mbps function or for data lengths larger than 37 bytes.

10 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_EDK.zip	Contains the Getting Started guide and installer for the standard RSL10 IDE (Kepler Eclipse).
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 which 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

Copyright © 2017 Semiconductor Components Industries, LLC (d/b/a ON Semiconductor).