s113_nrf52 release notes

Introduction to the s113_nrf52 release notes

About the document

These release notes describe the changes in the s113_nrf52 from version to version.

The release notes are intended to list all relevant changes in a given version. They are kept brief to make it easy to get an overview of the changes. More details regarding changes and new features may be found in the s113_nrf52 migration document (normally available for major releases only).

This document may be updated for an already released version of SoftDevice. The changes will be tagged with "**Update X**", where X is a number incremented each time the document has been revised.

Issue numbers in parentheses are for internal use and should be disregarded by the customer.

s113_nrf52_7.0.1

This is a production release that contains minor but important changes to the 7.0.0 release.

For some combinations of SoftDevice and nRF52 IC, only Bluetooth Core Specification v5.1 qualified listings are available with corresponding QDIDs from v7.0.1.

Updating to this version from v7.0.0 is recommended.

Notes:

- The release notes list changes since s113_nrf52_7.0.0.
- This SoftDevice is binary compatible to the s113_nrf52_7.0.0 and memory requirements have not changed. Applications are therefore not required to be recompiled.

SoftDevice properties

- **Update 1:** This SoftDevice variant is compatible with nRF52810, nRF52811, nRF52832, nRF52833, and nRF52840.
- This SoftDevice contains the Master Boot Record (MBR) version 2.4.1 (DRGN-10680).
  - This MBR version is compatible with previous MBR versions.
- The combined MBR and SoftDevice memory requirements for this version are as follows:
  - Flash: **112.0 kB (**0x1C000 bytes)
  - RAM: **4.4 kB** (0x1198 bytes). This is the minimum required memory. The actual requirements depend on the configuration chosen at `sd_ble_enable()` time.
  - Call stack: The SoftDevice uses a call stack combined with the application. The worst-case stack usage for the SoftDevice is **1.5 kB** (0x600 bytes). Application writers should ensure that enough stack space is reserved to cover the worst-case SoftDevice call stack usage combined with the worst-case application call stack usage.
- The Firmware ID of this SoftDevice is 0x00CC.

Changes

- SoftDevice
  - Bluetooth Core Specification v 5.1 compatibility (DRGN-12400).
  - The VersNr field in the LL_VERSION_IND packet now contains the value 0x0A to indicate Bluetooth Core Specification v 5.1 compatibility (DRGN-12466).
- LL
  - Bluetooth Core Specification Erratum #10818 is incorporated, allow HCI ACL data packets with 0-length payload, but do not transmit anything until receiving the next non-zero continuation fragment (DRGN-11430).

Bug fixes

- SoftDevice
  - Fixed an issue where USB support was not working for nRF52840 (DRGN-12470).
  - Fixed an issue where the time scheduled for a flash write or flash page erase using `sd_flash_write` or `sd_flash_page_erase` APIs on nRF52811 will be longer than required and the same as for nRF52832 (DRGN-12539).

Limitations

- SoftDevice
  - If Radio Notifications are enabled, flash write and flash erase operations initiated through the SoftDevice API will be notified to the application as Radio Events (DRGN-5197/FORT-809).
  - Synthesized low frequency clock source is not tested or intended for use with the BLE stack.
  - Applications must not modify the `SEVONPEND` flag in the `SCR` register when running in priority levels higher than 6 (priority level numerical values lower than 6) as this can lead to undefined behavior.
- GATT
  - To conform to the Bluetooth Core Specification v 5.0, there shall be no secondary service that is not referenced somehow by a primary service. The SoftDevice does not enforce this (DRGN-906).

Known Issues

- MBR
  - When copying the Bootloader on the nRF52811 using the `SD_MBR_COMMAND_COPY_BL` MBR command, the MBR will not write-protect itself. This does not change the behavior of the MBR or DFU process as the MBR cannot be configured to overwrite itself (DRGN-11287).
- SoftDevice
  - **Update 1:** When running on nRF52810, nRF52811 or nRF52832, using `sd_power_usb*` APIs can lead to undefined behaviour. Note that this issue was also present in the earlier s113_nrf52_7.0.0 SoftDevice (DRGN-12720).
  - **Update 1:** When running on nRF52833 or nRF52840, using `sd_flash_protect` or `sd_protected_register_write` APIs can lead to undefined behaviour (DRGN-12447).
  - The BLE_GAP_EVT_SEC_INFO_REQUEST event will not report the identity address of the peer to the application. This issue was also present in previous releases. A workaround is to do a mapping of the connection handle to the peer's identity address (DRGN-10340).
  - `sd_ble_gap_device_name_set()` may return `NRF_ERROR_INTERNAL` instead of `NRF_ERROR_NO_MEM` if the allocated space for the device name is too small. A workaround is to allocate enough space for the device name before calling `sd_ble_gap_device_name_set()` (DRGN-10195).
  - The SoftDevice will generate a resolvable address for the TargetA field in directed advertisements if the target device address is in the device identity list with a non-zero IRK, even if privacy is not enabled and the local device address is set to a public address. This issue was present also in previous releases. A workaround is to set the IRK to zero or to remove the device address from the device identity list (DRGN-10659).
- GATTC
  - The `ble_gattc_service_t::uuid` field is incorrectly populated in the `BLE_GATTC_EVT_PRIM_SRVC_DISC_RSP` event if the `sd_ble_gattc_primary_services_discover()` or `sd_ble_gattc_read()` is called when a Primary Service Discovery by Service UUID is already ongoing (DRGN-11300). When the application has called `sd_ble_gattc_primary_services_discover()`, it should wait for the `BLE_GATTC_EVT_PRIM_SRVC_DISC_RSP` event before calling `sd_ble_gattc_primary_services_discover()` or `sd_ble_gattc_read()`.
- LL
  - If the application adds an all zeroes IRK with the `sd_ble_gap_device_identities_set()`, it will be treated as a valid entry in the device identity list. An all zeroes IRK is invalid and must not be added (DRGN-9083).

s113_nrf52_7.0.0 (Deprecated)

The s113_nrf52_7.0.0 is the first production release of this SoftDevice variant. The S113 contains all the features present in the s112_nrf52_7.0.0 SoftDevice. In addition, it includes the LE Data Packet Length Extensions, Connection-Oriented Channels in LE Credit Based Flow Control Mode, and the ability to trigger a task, for example a GPIOTE task, at the start of a connection event. The S113 API is a compatible superset of the S112 SoftDevice API and a compatible subset of the S132 and S140 SoftDevice APIs. For features that are common to S112, S113, S132, and S140, the Application Programming Interface (API) is the same. To show the API compatibility, the S113 follows the same version numbering as S112, s132, and s140. For features that are available in the S113 compared to the S112 v 6.1.1, see section "New functionality" below.

Notes:

- The release notes list changes since s112_nrf52_6.1.1.

SoftDevice properties

- This SoftDevice variant is compatible with nRF52810, nRF52811, nRF52832 and nRF52840.
- This SoftDevice contains the Master Boot Record (MBR) version 2.4.1 (DRGN-10680).
  - This MBR version is compatible with previous MBR versions.
- The combined MBR and SoftDevice memory requirements for this version are as follows:
  - Flash: **112.0 kB (**0x1C000 bytes)
  - RAM: **4.4 kB** (0x1198 bytes). This is the minimum required memory. The actual requirements depend on the configuration chosen at `sd_ble_enable()` time.
  - Call stack: The SoftDevice uses a call stack combined with the application. The worst-case stack usage for the SoftDevice is **1.5 kB** (0x600 bytes). Application writers should ensure that enough stack space is reserved to cover the worst-case SoftDevice call stack usage combined with the worst-case application call stack usage.
- The Firmware ID of this SoftDevice is 0x00C3.

New functionality

- LL
  - Data length extension feature support (DRGN-7245)
- L2CAP
  - Connection-Oriented Channels in LE Credit Based Flow Control Mode (DRGN-8572).
- GAP
  - API to obtain the next connection event counter (DRGN-10913).
  - API for triggering a task when the SoftDevice is about to start a connection event (DRGN-10914).
  - API for inclusion configuration of the CAR and PPCP characteristics (DRGN-10874).

Changes

- SoftDevice
  - Removed macros defining PPI channels and groups available to the application (DRGN-10382).
- GAP
  - The API for configuring improved advertiser role scheduling is removed. The SoftDevice now uses the improved scheduling configuration by default (DRGN-10754).

Bug fixes

- SoftDevice
  - Fixed an issue where utilizing the MWU on nRF52832 would lead to undefined behavior (DRGN-10917).
  - Fixed an issue where the application would be blocked when requesting an earliest possible Radio Timeslot (DRGN-10402).
- LL
  - Fixed an issue where the slave might disconnect if many packets where lost and there was an ongoing Connection Parameter Update (DRGN-11147).

Limitations

- SoftDevice
  - If Radio Notifications are enabled, flash write and flash erase operations initiated through the SoftDevice API will be notified to the application as Radio Events (FORT-809).
  - Synthesized low frequency clock source is not tested or intended for use with the BLE stack.
  - Applications must not modify the `SEVONPEND` flag in the `SCR` register when running in priority levels higher than 6 (priority level numerical values lower than 6) as this can lead to undefined behavior.
- GATT
  - To conform to the Bluetooth Core Specification v 5.0, there shall be no secondary service that is not referenced somehow by a primary service. The SoftDevice does not enforce this (DRGN-906).

Known Issues

- MBR
  - When copying the Bootloader on the nRF52811 using the `SD_MBR_COMMAND_COPY_BL` MBR command, the MBR will not write-protect itself. This does not change the behavior of the MBR or DFU process as the MBR cannot be configured to overwrite itself (DRGN-11287).
- SoftDevice
  - The time scheduled for a flash write or flash page erase using `sd_flash_write` or `sd_flash_page_erase` APIs on nRF52811 will be longer than required and the same as for nRF52832.
  - When running on nRF52840, using `sd_flash_protect` or `sd_protected_register_write` APIs can lead to undefined behaviour.
  - The BLE_GAP_EVT_SEC_INFO_REQUEST event will not report the identity address of the peer to the application. This issue was also present in previous releases. A workaround is to do a mapping of the connection handle to the peer's identity address (DRGN-10340).
  - `sd_ble_gap_device_name_set()` may return `NRF_ERROR_INTERNAL` instead of `NRF_ERROR_NO_MEM` if the allocated space for the device name is too small. A workaround is to allocate enough space for the device name before calling `sd_ble_gap_device_name_set()` (DRGN-10195).

- A memory access fault (NRF_FAULT_ID_APP_MEMACC) can occur in sd_nvic_critical_region_exit() if a high priority SoftDevice interrupt occurs during a critical section, for example due to radio traffic (DRGN-10613). This issue was present also in previous releases. It can be fixed by editing __NRF_NVIC_SD_IRQS_1 in nrf_nvic.h so that it becomes:

```
#define __NRF_NVIC_SD_IRQS_1 ((uint32_t)(1U << (MWU_IRQn - 32)))
```

- The SoftDevice will generate a resolvable address for the TargetA field in directed advertisements if the target device address is in the device identity list with a non-zero IRK, even if privacy is not enabled and the local device address is set to a public address. This issue was present also in previous releases. A workaround is to set the IRK to zero or to remove the device address from the device identity list (DRGN-10659).

- GATTC
  - The ble_gattc_service_t::uuid field is incorrectly populated in the BLE_GATTC_EVT_PRIM_SRVC_DISC_RSP event if the sd_ble_gattc_primary_services_discover() or sd_ble_gattc_read() is called when a Primary Service Discovery by Service UUID is already ongoing (DRGN-11300). When the application has called sd_ble_gattc_primary_services_discover(), it should wait for the BLE_GATTC_EVT_PRIM_SRVC_DISC_RSP event before calling sd_ble_gattc_primary_services_discover() or sd_ble_gattc_read().