

# **s132\_nrf52\_5.0.0-1.alpha release notes**

## **Introduction to the s132\_nrf52 release notes**

### **About the document**

The release notes describe the changes in the s132\_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 the overview. More details regarding changes and new features may be found in the s132\_nrf52 migration document (normally available for major releases only).

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

Copyright (c) Nordic Semiconductor ASA. All rights reserved.

# s132\_nrf52\_5.0.0-1.alpha

The s132 is a SoftDevice for the nRF52832 chip.

These release notes list the changes and differences from **s132\_nrf52\_3.0.0**.

Notes:

- This is a major release which has changed the Application Programming Interface (API), requiring applications to be recompiled.
- The memory requirements of the SoftDevice have changed.

## SoftDevice properties

- The combined MBR and SoftDevice memory requirements for this version are as follows:
  - Flash: **128 kB** (0x20000 bytes).
  - RAM: **6.43 kB** (0x19C0 bytes) (minimum required memory - actual requirements are dependent upon 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 **1.54 kb** (0x624 bytes) (s132\_nrf52\_3.0.0 has 0x600 bytes of worst case stack usage). Application writers should ensure that enough stack space is reserved to cover both worst case SoftDevice call stack usage combined with worst case application call stack usage.

## New functionality

- LL
  - Support for transmitting and receiving on the 2 Mbps PHY has been added (DRGN-7552).

## Using 2 Mbps

The SoftDevice provides a new GAP option `BLE_GAP_OPT_PREFERRED_PHYS_SET`, a new SV call `sd_ble_gap_phy_request()`, and a new event, `BLE_GAP_EVT_PHY_UPDATE` to support the new PHY. Please read the API documentation for more details about these.

This alpha version of the SoftDevice supports connection establishment using the 1 Mbps PHY and then changing either the transmitting PHY or the receiving PHY (asymmetric link configuration), or both (symmetric link configuration) to use the 2 Mbps PHY. The PHYs can be changed using the above mentioned SV call.

Link Layer encryption and long data packet payload (up to 251 octets) are supported on both 1Mbps and 2Mbps PHYs.

## Bug fixes

There are no bug fixes in this release.

## 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 level 1 as this can lead to undefined behavior.
  - If the application uses `TIMER0` inside a timeslot (scheduled with the Radio Timeslot API), `INTENSET` for `TIMER0` must be cleared before the timeslot ends (DRGN-7776).
- LL
  - The peripheral role has priority over the central role when it comes to keeping the links alive.
  - For 2 Mbps, see the section "Using 2 Mbps" above.
- GAP
  - A broadcaster **and** a scanner cannot both be active if there are 8 connections established (DRGN-6543).
- GATTS

- To conform to the Bluetooth specification there shall not be a secondary service that is not referenced somehow by a primary service. The SoftDevice does not enforce this (DRGN-906, DRGN-2260).

## Known Issues

- If `sd_softdevice_enable()` is called with `fault_handler` set to `NULL`, an invalid function pointer, or a pointer to a returning function, the behavior will be undefined (DRGN-7122).
- If Connection Event Length Extension is enabled, the Radio Notification may be suppressed between connection events (DRGN-7687).
- When `sd_ble_gap_connect()` returns an error code, the scanner may be stopped (DRGN-7679). To ensure the scanner is in a known state, `sd_ble_gap_scan_stop()` should be used to stop the scanner when `sd_ble_gap_connect()` returns an error code.

