S130 nRF51822

Bluetooth® Low Energy Central and Peripheral SoftDevice

SoftDevice Specification 0.5

Key Features:

- *Bluetooth* 4.1 compliant low energy single-mode protocol stack
  - Link layer
  - L2CAP, ATT and SM protocols
  - GATT and GAP APIs
  - Concurrent Central, Observer, Peripheral and Broadcaster roles
    - 3 central connections
    - 1 peripheral connection
    - 1 observer
    - 1 broadcaster
  - GATT Client and Server
  - Full SMP support including MITM and OOB pairing
- Complementary nRF518 SDK including *Bluetooth*® profiles and example applications
- Master Boot Record for over-the-air device firmware update
- Memory isolation from Application for protocol implementation robustness and security
- Thread-safe supervisor-call-based API
- Asynchronous event-driven behavior
- No RTOS dependency
  - A RTOS of your choice can be used
- No link-time dependencies
  - Standard ARM™ Cortex M0 project configuration for application development
- Support for multiprotocol operation concurrent with *Bluetooth low energy* connections and non-concurrently
  - Concurrent multiprotocol timeslot API
  - Alternate protocol stack running in application space

Applications:

- A4WP Wireless charging
- Sports & Fitness devices
  - Sports watch
  - Bike computers
- Computer peripherals and I/O devices
  - Mouse
  - Keyboard
  - Multi-touch trackpad
- Interactive entertainment devices
  - Remote control
  - Gaming controller
- Personal Area Networks
  - Health and fitness sensor and monitor devices
  - Medical devices
  - Key fobs and wrist watches
- Remote control toys
- Home automation

Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

Life support applications

Nordic Semiconductor's products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

Contact details

For your nearest dealer, please see www.nordicsemi.com

Information regarding product updates, downloads, and technical support can be accessed through your My Page account on our homepage.

Main office:
Otto Nielsens veg 12
7052 Trondheim
Norway
Phone: +47 72 89 89 00
Fax: +47 72 89 89 89

Mailing address:
Nordic Semiconductor
P.O. Box 2336
7004 Trondheim
Norway

Document Status

Status	Description
V0.5	This specification contains target specifications for product development.
V0.7	This specification contains preliminary data; supplementary data may be published from Nordic Semiconductor ASA later
V1.0	This specification contains final product specifications. Nordic Semiconductor ASA reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

Revision History

Date	Version	Description
July 2014	V0.5	Preliminary release with S130 v0.5 alpha

TABLE OF CONTENTS

1 Introduction

S130 is a *Bluetooth*® low energy (BLE) Central and Peripheral protocol stack solution supporting up to 3 central and 1 peripheral simultaneous connections and concurrent observer and broadcaster. It integrates a low energy controller and host, and provides a full and flexible API for building *Bluetooth* low energy System on Chip (SoC) solutions.

This document contains information about the SoftDevice features and performance.

Note: The SoftDevice features and performance are subject to change between revisions of this document. See Section *Notification of SoftDevice revision updates on page 25* for more information. To find information on any limitations or omissions, please refer to the SoftDevice release notes, which contain a detailed summary of the release status.

1.1. Documentation

Below is a list of the core documentation for the SoftDevice.

Document	Description
nRF51 Series Reference Manual	"Appendix A: SoftDevice architecture" in the nRF51 Series Reference Manual is essential reading for understanding the resource usage and performance related chapters of this document.
nRF51822 Product Specification (PS)	Contains a description of the hardware, modules, and electrical specifications specific to the nRF51822 chip.
nRF51822 Product Anomaly Notification (PAN)	Contains information on anomalies related to the nRF51822 chip.
Bluetooth Core Specification	The Bluetooth Core Specification version 4.1, Volumes 1, 3, 4, and 6 describes Bluetooth terminology which is used throughout the SoftDevice Specification.

2 Product overview

This section provides an overview of the SoftDevice.

2.1. SoftDevice

The SoftDevice is pre-compiled and linked binary software that integrates a *Bluetooth* 4.1 low energy (BLE) protocol stack.

The Application Programming Interface (API) is offered as a standard C language set of functions and data types that give the application complete compiler and linker independence from the SoftDevice implementation.

The SoftDevice enables the application programmer to develop their code as a standard ARM® Cortex M0 project without the need to integrate with proprietary chip-vendor software frameworks. This means that any ARM® Cortex M0-compatible tool chains can be used to develop *Bluetooth* low energy applications with the SoftDevice.

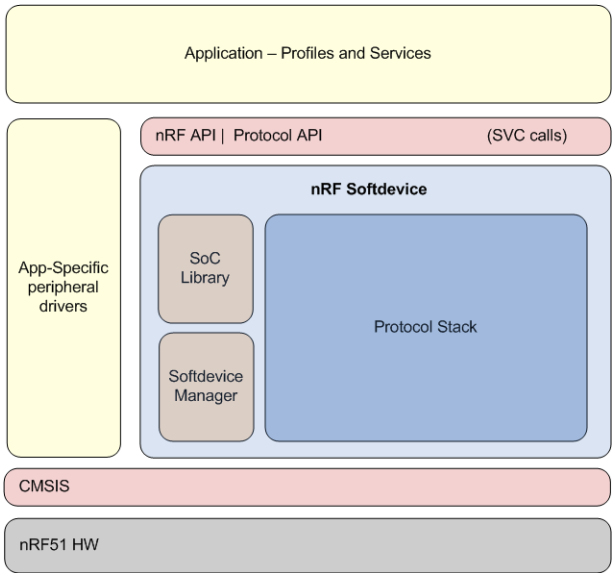

Figure 1 System on Chip application with the SoftDevice

The SoftDevice can be programmed onto compatible nRF51 Series chips during both development and production. This specification outlines the supported features of a production level SoftDevice. Alpha and Beta versions may not support all features.

2.2. Multi-protocol support

The SoftDevice supports both non-concurrent and fully concurrent multiprotocol implementations. For non-concurrent operation, a proprietary 2.4 GHz protocol can be implemented in the application program area and can access all hardware resources when the SoftDevice is disabled. For concurrent multiprotocol operation, with a proprietary protocol running concurrently with the SoftDevice protocol, see chapter *Concurrent Multiprotocol Timeslot API on page 17*.

3 *Bluetooth* low energy Protocol Stack

The Bluetooth 4.1 compliant low energy Host and Controller embedded in the SoftDevice are fully qualified with multi-role support (Peripheral and Broadcaster). The API is defined above the Generic Attribute Protocol (GATT), Generic Access Profile (GAP), and Logical Link Control and Adaptation Protocol (L2CAP). The SoftDevice allows applications to implement standard Bluetooth low energy profiles as well as proprietary use case implementations.

The nRF51 Software Development Kit (SDK) complements the BLE protocol stack with Service and Profile implementations. Single-mode System on Chip (SoC) applications are enabled by the full BLE protocol stack and nRF51xxx integrated circuit (IC).

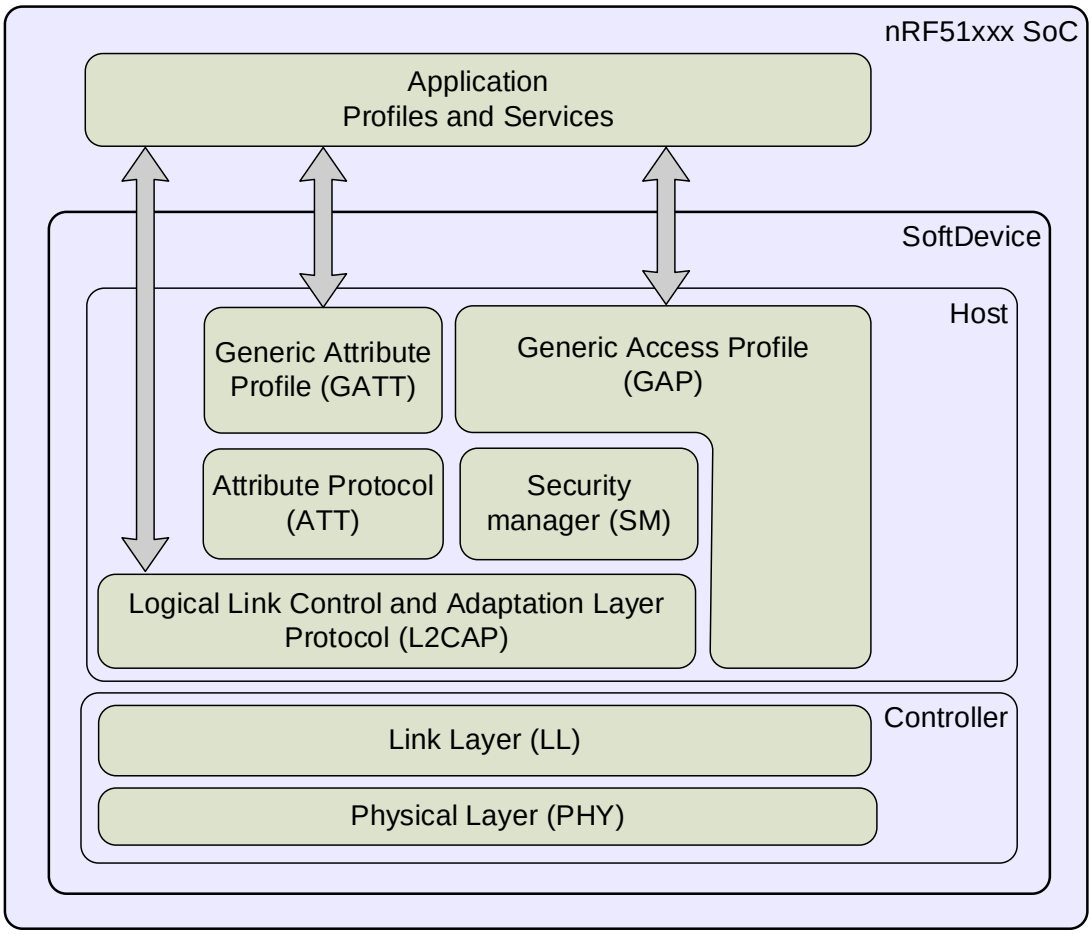

Figure 2 SoftDevice Stack Architecture

3.1. Profile and service support

The Profiles and corresponding Services supported by the SoftDevice are shown in the following table.

Adopted Profile	Adopted Services
HID over GATT	HID Battery Device Information
Heart Rate	Heart Rate Device Information
Proximity	Link Loss Immediate Alert Tx Power
Blood pressure	Blood pressure
Health Thermometer	Health Thermometer
Glucose	Glucose
Phone Alert Status	Phone Alert Status
Alert Notification	Alert Notification
Time	Current Time Next DST Change Reference Time Update
Find Me	Immediate Alert
Cycling speed & cadence	Cycling speed and cadence Device information
Running speed & cadence	Running speed and cadence Device information
Location and Navigation	Location and Navigation
Cycling Power	Cycling Power
Scan Parameters	Scan Parameters
	User Data

Table 1 Supported profiles and services

Note: Examples for selected profiles and services are available in the nRF51 SDK. See the SDK documentation for details.

3.2. *Bluetooth* low energy features

The BLE protocol stack in the SoftDevice has been designed to provide an abstract but flexible interface for application development for *Bluetooth* low energy devices. GAP, GATT, SM, and L2CAP are implemented in the SoftDevice and managed through the API. The SoftDevice implements GAP and GATT procedures and modes that are common to most profiles such as the handling of discovery, connection, pairing, and bonding.

The BLE API is consistent across *Bluetooth* role implementations where common features have the same interface. The following tables describe the features found in the BLE protocol stack.

API Features	Description
Interface to: GATT / GAP	Consistency between APIs including shared data formats
Attribute table population and access	Full flexibility to populate the attribute table at runtime, attribute removal is not supported
Asynchronous and event driven	Thread-safe function and event model enforced by the architecture
Vendor-specific (128-bit) UUIDs for proprietary profiles	Compact, fast and memory efficient management of 128-bit UUIDs
Packet flow control	Full application control over data buffers to ensure maximum throughput

Table 2 API features in the BLE stack

GAP Features	Description
Multi-role: Central, Peripheral, Observer & Broadcaster	Broadcaster and observer can run concurrently with peripheral and central connections. It is not possible to start a discoverable or connectable advertiser and a broadcaster concurrently. It is not possible to start a scanner and an observer concurrently.
Multiple bond support	Security keys and peer information stored in application space No limitations in stack implementation.
Security Mode 1: Level 1, 2 & 3	Support for all levels of SM 1.
User-defined Advertising data	Full control over advertising and scan response data for the application.

Table 3 GAP features in the BLE stack

GATT Features	Description
Full GATT Server	Including Service Changed Support Support for 4 concurrent ATT server sessions
Support for authorization	Enables control points Enables freshest data Enables GAP authorization
Full GATT Client	Flexible data management options for packet transmission with either fine control or abstract management
Implemented GATT Sub-procedures	Discover all Primary Services Discover Primary Service by Service UUID Find included Services Discover All Characteristics of a Service Discover Characteristics by UUID Discover All Characteristic Descriptors Read Characteristic Value Read using Characteristic UUID Read Long Characteristic Values Write Without Response Write Characteristic Value Notifications Indications Read Characteristic Descriptors Read Long Characteristic Descriptors Write Characteristic Descriptors Write Long Characteristic Value Write Long Characteristic Descriptors Reliable Writes

Table 4 GATT features in the BLE stack

Security Manager Features	Description
Lightweight key storage for reduced NV memory requirements	Efficient usage of key generation algorithms to minimize memory overheads.
Authenticated MITM (Man in the middle) protection	Protects the bonding procedure against malicious attackers. Allows for per-link elevation of the encryption security level.
Pairing methods: Just works, Passkey Entry and Out of Band	Full control over the pairing algorithm for strict security requirements.

Table 5 Security Manager (SM) features in the BLE stack

Attribute Protocol Features	Description
Server protocol	
Client protocol	
Max MTU Size 23 bytes	

Table 6 Attribute Protocol (ATT) features in the BLE stack

L2CAP Features	Description
27 byte MTU size	

Table 7 Logical Link Controller and Adaptation Layer Protocol (L2CAP) features in the BLE stack

Controller, Link Layer Features	Description
Master role	
Slave role	
Slave connection parameter update	
Encryption	

Table 8 Controller, Link Layer (LL) features in the BLE stack

Proprietary Features	Description
Tx Power control	Access for the application to change TX power settings anytime
Channel Map configuration	Setup of channel map for all connections from the application
Full Privacy 1.1 support	Synchronous and low power solution for BLE enhanced privacy with hardware-accelerated address resolution for whitelisting
Master Boot Record (MBR) for Device Firmware Update (DFU)	Enables over-the-air SoftDevice replacement, giving full SoftDevice update capability

Table 9Proprietary features in the BLE stack

1.1.1. Limitations on procedure concurrency

When there are multiple connections in the Central role and or Peripheral role, the concurrency of protocol procedures will have some limitations. The Host instantiates both GATT and GAP instances for each connection, while the SM Initiator is only instantiated once for all connections. The Link Layer also has concurrent procedure limitations that are handled inside the SoftDevice without requiring management from the application.

The limitations are outlined in *Table 10 Procedure concurrency* below.

Protocol procedures	Limitation with multiple connections active
GATT	None. All procedures can be executed in parallel.
GAP	None. All procedures can be executed in parallel. Note that some GAP procedures require LL procedures (connection parameter update and encryption). In this case, the GAP module will queue the LL procedures and execute them in sequence.
SM	SM procedures cannot be executed in parallel, that is, each SM procedure must run to completion before the next procedure begins across all connections. For example sd_ble_gap_authenticate().
LL	LL Disconnect procedure has no limitations and can be executed on any, or all, links simultaneously. All peer initiated control procedures will be accepted according to specification. All local initiated procedures (e.g. connection parameter update and encryption) can only execute one at a time for a given role. A procedure can be initiated on a master link and slave link in parallel.

Table 10 Procedure concurrency

4 SoC library

The following features are in place to ensure the Application and SoftDevice coexist with safe sharing of common SoC resources.

Feature	Description
Mutex	The SoftDevice implements atomic mutex acquire and release operations that are safe for the application to use. Use this mutex to avoid disabling global interrupts in the application, because disabling global interrupts will interfere with the SoftDevice and may lead to dropped packets or lost connections.
NVIC	Gives the application access to all NVIC features without corrupting SoftDevice configurations.
Rand	Provides random numbers from the hardware random number generator.
Power	Access to POWER block configuration while the SoftDevice is enabled: - Access to RESETREAS register - Set power modes - Configure power fail comparator - Control RAM block power - Use general purpose retention register - Configure DC/DC converter state o OFF o ON o AUTOMATIC - The SoftDevice will manage the DC/DC converter state by switching it on for all Radio Events and off all other times.
Clock	Access to CLOCK block configuration while the SoftDevice is enabled. Allows the HFCLK Crystal Oscillator source to be requested by the application.
Wait for event	Simple power management call for the application to use to enter a sleep or idle state and wait for an event.
PPI	Configuration interface for PPI channels and groups reserved for an application.
Concurrent Multiprotocol Timeslot API	Schedule other radio protocol activity, see chapter *Concurrent Multiprotocol Timeslot API on page 17*.
Radio Notification	Configure Radio Notification signals on ACTIVE and/or nACTIVE. See chapter *Radio Notification on page 17*.
Block Encrypt (ECB)	Safe use of 128 bit AES encrypt HW accelerator.
Event API	Fetch asynchronous events generated by the SoC library.
Flash memory API	Application access to flash write, erase, and protect. Can be safely used during all protocol stack states.
Temperature	Application access to the temperature sensor.

Master Boot Record (MBR)	The MBR provides support for Bootloader implementation and Firmware update functions.

5 SoftDevice Manager

The following features enable the Application to manage the SoftDevice on a top level.

Feature	Description
SoftDevice control API	Control of SoftDevice state through enable and disable. On enable, the low frequency clock source selects between the following options: • RC oscillator • Crystal oscillator

6 Flash Memory API

TBD

The implementation of the Flash memory API is the same as is the S110 v7.0.0 and S120 v1.0.0 SoftDevices however the behavior is not characterized for the S130 SoftDevice at this time.

7 Radio Notification

TBD

The implementation of the Radio Notification is the same as is the S110 v7.0.0 and S120 v1.0.0 SoftDevices however the behavior is not characterized for the S130 SoftDevice at this time.

8 Concurrent Multiprotocol Timeslot API

TBD

The Concurrent Multiprotocol Timeslot API is not implemented in S130 v0.5 alpha but will be implemented for the v1.0.0 release.

9 Bootloader

TBD

The Bootloader implementation is the same as is the S120 v1.0.0 SoftDevice.

10 SoftDevice Resource Requirements

After the SoftDevice is installed on a System on Chip (SoC) it is located in the lower part of the code memory space. When enabled, the SoftDevice controls and uses resources from the chip, including reserving RAM space for its operation and access to hardware peripherals. This chapter describes how the SoftDevice – when both enabled and disabled - uses memory and hardware resources.

10.1. Memory resource map and usage

The memory map for program memory and RAM at run time with the SoftDevice enabled is illustrated in *Figure 3* below. Memory resource requirements, both when the SoftDevice is enabled and disabled, are shown in *Table 11*.

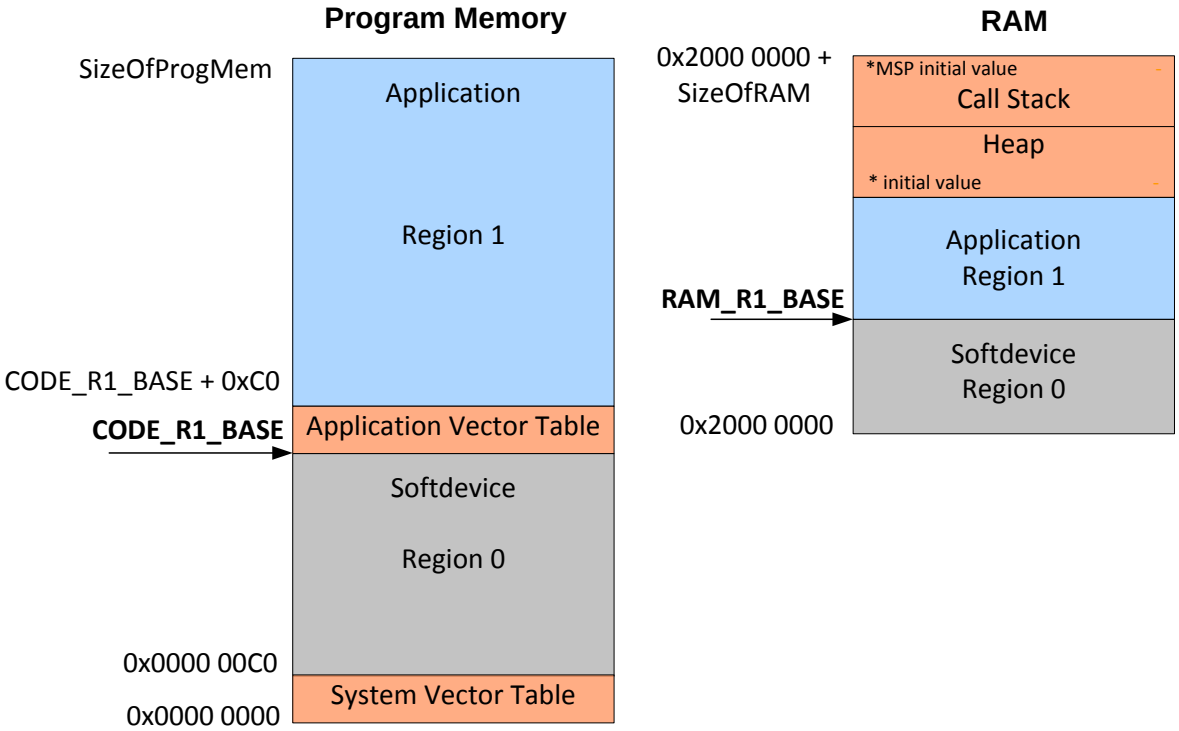

Figure 3 Memory resource map

Flash	S130 Enabled	S130 Disabled
Amount	128 kB[a]	128 kB
CODE_R1_BASE	0x00020000	0x00020000

RAM	S130 Enabled	S130 Disabled
Amount (minimum: 1 link)	10 kB	8 bytes
RAM_R1_BASE (1 link, excl. ATTDB size)	0x20002800	0x20000008

Call stack[b]	S130 Enabled	S130 Disabled
Maximum usage	1536 bytes (0x600)	0x00

Heap	S130 Enabled	S130 Disabled
Maximum allocated bytes	0 bytes (0x00)	0x00

Table 11 S130 Memory resource requirements

10.2. Hardware blocks and interrupt vectors

Table 12 defines access types used to indicate the availability of hardware blocks to the application. *Table 13* specifies the access the application has, per hardware block, both when the SoftDevice is enabled and disabled.

Access type	Definition
Restricted	Used by the SoftDevice and outside the application sandbox. Application has limited access through the SoftDevice API.
Blocked	Used by the SoftDevice and outside the application sandbox. Application has no access.
Open	Not used by the SoftDevice. Application has full access

Table 12 Hardware access type definitions

[a] 1kB = 1024 bytes

[b] This is only the callstack used by the SoftDevice at run time. The application call stack memory usage must be added for the total call stack size to be set in the user application.

ID	Base address	Instance	Access (SoftDevice enabled)	Access (SoftDevice disabled)
0	0x40000000	MPU	Restricted	Open
0	0x40000000	POWER	Restricted	Open
0	0x40000000	CLOCK	Restricted	Open
1	0x40001000	RADIO	Blocked	Open
2	0x40002000	UART0	Open	Open
3	0x40003000	SPIM0 / 2W0	Open	Open
4	0x40004000	SPIM1/2W1/SPIS1	Open	Open
		…		
6	0x40006000	GPIOTE	Open	Open
7	0x40007000	ADC	Open	Open
8	0x40008000	TIMER0	Blocked	Open
9	0x40009000	TIMER1	Open	Open
10	0x4000A000	TIMER2	Open	Open
11	0x4000B000	RTC0	Blocked	Open
12	0x4000C000	TEMP	Open	Open
13	0x4000D000	RNG	Restricted	Open
14	0x4000E000	ECB	Restricted	Open
15	0x4000F000	CCM	Blocked	Open
15	0x4000F000	AAR	Blocked	Open
16	0x40010000	WDT	Open	Open
17	0x40011000	RTC1	Open	Open
18	0x40012000	QDEC	Open	Open
19	0x40013000	LPCOMP	Open	Open
20	0x40014000	Software interrupt	Open	Open
21	0x40015000	Radio Notification	Restricted[a]	Open
22	0x40016000	SoC Events	Blocked	Open
23	0x40017000	Software interrupt	Blocked	Open
24	0x40018000	Software interrupt	Blocked	Open
25	0x40019000	Software interrupt	Blocked	Open
		…		
30	0x4001E000	NVMC	Open	Open
31	0x4001F000	PPI	Restricted	Open
NA	0x50000000	GPIO P0	Open	Open
NA	0xE000E100	NVIC	Restricted[b]	Open

Table 13 Peripherals used by the SoftDevice

[a] Blocked only when radio notification signal is enabled. See Table <<XX>> on page <<XX>> for software interrupt allocation.

[b] Not protected. For robust system function, the application program must comply with the restriction and use the NVIC API for configuration when the SoftDevice is enabled.

10.3. Application signals – software interrupts (SWI)

Software interrupts are used by the SoftDevice to signal a change in events. The table below shows the allocation of software interrupt vectors to SoftDevice signals.

SWI	Peripheral ID	SoftDevice Signal
0	20	Unused by the SoftDevice and available to the application.
1	21	Radio Notification - optionally configured through API.
2	22	SoftDevice Event Notification.
3	23	Reserved.
4	24	LowerStack processing - not user configurable.
5	25	UpperStack signaling - not user configurable.

Table 14 Software interrupt allocation

10.4. Programmable Peripheral Interconnect (PPI)

When the SoftDevice is enabled, the PPI is restricted with only some PPI channels and groups available to the application. The table below shows how channels and groups are assigned between the application and SoftDevice.

Note: All PPI channels are available to the application when the SoftDevice is disabled.

PPI Channel Allocation	SoftDevice enabled	SoftDevice disabled
Application	Channels 0-7	Channels 0-15
SoftDevice	Channels 8-15	-

Preprogrammed Channels	SoftDevice enabled	SoftDevice disabled
Application	-	Channels 20-31
SoftDevice	Channels 20-31	-

PPI group allocation	SoftDevice enabled	SoftDevice disabled
Application	Groups 0-1	Groups 0-3
SoftDevice	Groups 2-3	-

10.5. SVC number ranges

Table 15 shows which SVC numbers an application program can use and which numbers are used by the SoftDevice.

Note: The SVC number allocation does not change with the state of the SoftDevice (enabled or disabled).

SVC number allocation	SoftDevice enabled	SoftDevice disabled
Application	0x00-0x0F	0x00-0x0F
SoftDevice	0x10-0xFF	0x10-0xFF

Table 15 SVC number allocation

10.6. External requirements

For correct operation of the SoftDevice, it is a requirement that the 16 MHz crystal oscillator (16 MHz XOSC) startup time is less than 1.5 ms. The external clock crystal and other related components must be chosen accordingly. Data for the device XOSC input can be found in the product specification for the device.

11 Processor availability and interrupt latency

TBD

The processor availability and interrupt latency for peripheral and central connection events is as documented for the S110 and S120 SoftDevices respectively. Please refer the SoftDevice specifications.

12 BLE data throughput

TBD

The data throughput for Central connections is the same for the S130 as it is for the S120 v1.0.0 SoftDevice.

The data throughput for Peripheral connections for the S130 is 1/2 of the data throughput when receiving and 1/3 of the data throughput when transmitting compared to the S110 v7.0.0 SoftDevice.

The data throughput for concurrent central and peripheral connections is not yet characterized.

13 BLE power profiles

TBD

The power profile for Central connection events for the S130 is the same as for the S120 v1.0.0 SoftDevice.

The power profile for Peripheral connection events for the S130 is the same as for S110 v7.0.0 SoftDevice.

The power profiles for concurrent central and peripheral connection events is not yet characterized.

14 SoftDevice identification and revision scheme

The SoftDevices will be identified by the SoftDevice part code, a qualified IC partcode (for example, nRF51822), and a version string.

For revisions of the SoftDevice which are production qualified, the version string consists of major, minor, and revision numbers only, as described in Table 36.

For revisions of the SoftDevice which are not production qualified, a build number and a test qualification level (alpha/beta) are appended to the version string.

For example: s110_nrf51822_1.2.3-4.alpha, where major = 1, minor = 2, revision = 3, build number = 4 and test qualification level is alpha. Additional SoftDevice revision examples are given in Table 37

Revision	Description
Major increments	Modifications to the API or the function or behavior of the implementation or part of it have changed. Changes as per Minor Increment may have been made. Application code will not be compatible without some modification.
Minor increments	Additional features and/or API calls are available. Changes as per Revision Increment may have been made. Application code may have to be modified to take advantage of new features.
Revision increments	Issues have been resolved or improvements to performance implemented. Existing application code will not require any modification.
Build number increment (if present)	New build of non-production version.

Table 16 Revision scheme

Sequence number	Description
s110_nrf51822_1.2.3-1.alpha	Revision 1.2.3, first build, qualified at alpha level
s110_nrf51822_1.2.3-2.alpha	Revision 1.2.3, second build, qualified at alpha level
s110_nrf51822_1.2.3-5.beta	Revision 1.2.3, fifth build, qualified at beta level
s110_nrf51822_1.2.3	Revision 1.2.3, qualified at production level

Table 17 SoftDevice revision examples

Qualification	Description
Alpha	Development release suitable for prototype application development. Hardware integration testing is not complete. Known issues may not be fixed between alpha releases. Incomplete and subject to change.
Beta	Development release suitable for application development. In addition to alpha qualification: Hardware integration testing is complete but may not be feature complete and may contain known issues. Protocol implementations are tested for conformance and interoperability.
Production	Qualified release suitable for product integration. In addition to beta qualification: Hardware integration tested over supported range of operating conditions. Stable and complete with no known issues. Protocol implementations conform to standards.

Table 18 Test qualification levels

14.1. Notification of SoftDevice revision updates

When new versions of a SoftDevice become available or the qualification status of a given revision of a SoftDevice is changed, product update notifications will be automatically forwarded, by email, to all users who have a profile configured to receive notifications from the Nordic Semiconductor website.

The SoftDevice will be updated with additional features and/or fixed issues if needed. Supported production versions of the SoftDevice will remain available after updates, so products do not need requalification on release of updates if the previous version is sufficiently feature complete for your product.