# S210 nRF51422

ANT™

# SoftDevice Specification v1.0

## Key Features

- Embedded ANT stack
    - Simple to complex network topologies:
        - Peer-to-peer
        - Star
        - Tree
        - Star-to-star and more
    - Supports ANT+ device profile implementations enabling multi-vendor interoperability
    - Interoperable with nRF24AP1, nRF24AP2 variants, and Dynastream ANT chipset/module based products
    - Broadcast, acknowledged, and burst communication modes
    - Configurable channel period 5.2 ms - 2 s
    - Built-in device search and pairing
    - Built-in interference handling
    - Built-in timing and power management
    - Enhanced ANT features
        - Advanced burst transfer modes (up to 60 kbps)
        - Optional single channel encryption (AES-128)
        - Supports up to 8 public, private, and/or managed networks
        - Advanced power management features to optimize application power consumption including Event Filtering and Selective Data Updates
- Asynchronous event-driven behavior
- Complementary nRF514 SDK including ANT profiles and example applications
- Low link-time dependencies
- No RTOS dependency - you can choose which RTOS to use
- Safe application and protocol coexistence through memory isolation
- Standard ARM® Cortex™ -M0 project configuration for application development
- Thread-safe supervisor-call based API

## Applications

- Personal area networks
    - Sport and fitness sensors and monitoring devices
    - Healthcare and lifestyle sensors
    - Personal convenience devices, key fobs
- Environmental sensor networks/high density wireless networking and monitoring
- Home and industrial control and data acquisition, intelligent domestic appliances
- Logistics/goods tracking
- Smart RF tags

## 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.

## 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 http://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

## RoHS and REACH statement

Nordic Semiconductor's products meet the requirements of Directive 2002/95/EC of the European Parliament and of the Council on the Restriction of Hazardous Substances (RoHS) and the requirements of the REACH regulation (EC 1907/2006) on Registration, Evaluation, Authorization, and Restriction of Chemicals.

The SVHC (Substances of Very High Concern) candidate list is continually being updated. Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website www.nordicsemi.com.

# Revision History

| Date | Version | Description |
|---|---|---|
| December 2012 | 1.0 | Updated features and applications section; updated ***Figure 2***; added content to ***section 3.2 on page 8***; updated call stack Maximum usage to 400 bytes in ***section 6.1 on page 12***; updated ***Figure 4*** on ***page 16***; added ***Table 3 on page 17*** and ***Table 4 on page 17***. |
| October 2012 | 0.6.1 | Updated flash size of the S210 SoftDevice to 40 kb. |
| September 2012 | 0.6 | |

# Contents

# 1 Introduction

S210 SoftDevice is an ANT protocol stack solution that provides a full, flexible application programming interface (API) for building ANT System on Chip (SoC) solutions for Nordic Semiconductor nRF51 ICs. S210 SoftDevice reduces the need for a secondary application host MCU to ANT solutions.

This document contains information about the SoftDevice features and performance which are subject to change between revisions.

## 1.1 Required reading

The *nRF51422 Product Specification* and the *nRF51 Series Reference Manual* are essential developer resources for ANT solutions from Nordic Semiconductor. The Software Architecture section of the *nRF51 Series Reference Manual* is essential reading for understanding the resource usage and performance related chapters of this document.

Knowledge of the ANT protocol is required to use the S210 correctly and to understand the terminology used in this document.

## 1.2 Writing conventions

This Product Specification follows a set of typographic rules to ensure that the document is consistent and easy to read. The following writing conventions are used:

- Command and event names, bit state conditions, and register names are written in `Lucida Console`.
- Pin names and pin signal conditions are written in **`Consolas.`**
- File names and user interface components are written in **bold**.
- Internal cross references are italicized and written in ***semi-bold.***
- Placeholders for parameters are written in *italics*. For example, a syntax description of SetChannelPeriod will be written as: SetChannelPeriod(*ChannelNumber, MessagingPeriod*).
- Fixed parameters are written in regular font. For example, a syntax description of SetChannelPeriod will be written as: SetChannelPeriod (0, Period).

# 2 Product overview

S210 SoftDevice is precompiled, pre-linked, and pre-programmed software implementing an ANT protocol stack. The S210 is compatible with selected nRF514xx System on Chip (SoC) devices. The application programming interface (API) is implemented through supervisor calls resembling regular C-functions. This enables complete application 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 proprietary chip-vendor software frameworks. This means that any ARM Cortex™-M0 compatible toolchain can be used to develop ANT/ANT+ applications with the S210 SoftDevice.

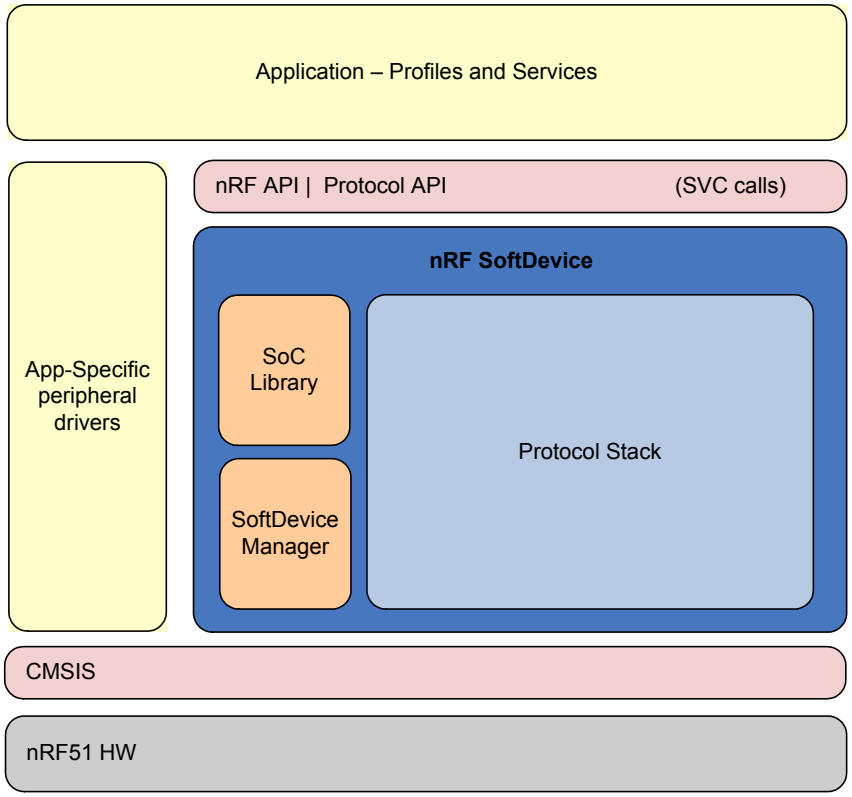



***Figure 1*** *System on Chip application with the SoftDevice*

This specification outlines features and support for the production version of the S210 SoftDevice. Pre-production versions may not have support for all features. To find information on any limitations or omissions, the S210 SoftDevice release notes will contain a detailed summary of the release status.

# 3 ANT Protocol Stack

S210 SoftDevice is a fully qualified ANT compliant stack that is pre-programmed on selected devices in the nRF514 family. S210 SoftDevice supports all ANT core functionality, including ANT+ profiles.
See http://thisisant.com for further information regarding ANT.

**Note:**
- ANT+ implementations must pass the ANT+ certification process to receive ANT+ Compliance Certificates.
- ANT+ network keys are needed to make ANT+ compliant products. The keys can be obtained by registering as an ANT+ adopter at http://thisisant.com.

## 3.1 Overview

The nRF514 Software Development Kit (SDK) supplements the ANT protocol stack with complete peripheral drivers, example applications, and ANT+ profile implementations.

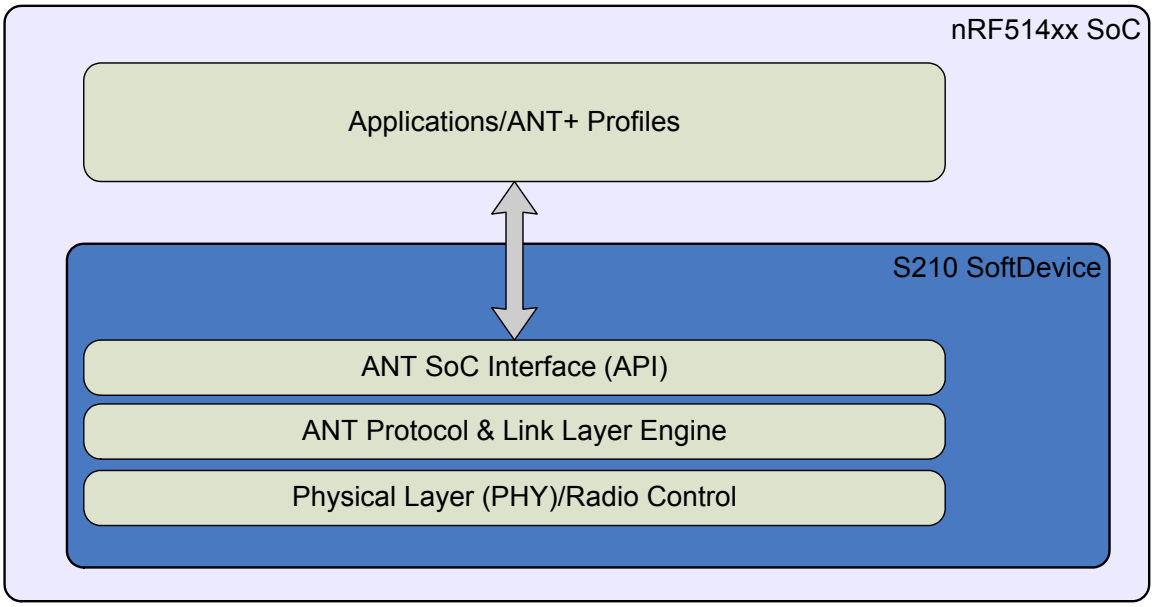



***Figure 2*** *ANT stack architecture*

## 3.2 ANT profile and feature support

All ANT nRF24AP1 and nRF24AP2 profiles and features are fully supported. Additional enhanced ANT features are available on the nRF51422 and are described below.

### 3.2.1 Advanced Burst Transfer

Advanced Burst Transfer is intended to facilitate and improve the ability of devices to transfer information. This is employed primarily in ANT-FS use cases, with an emphasis on longer file transfers. Advanced Burst Transfer improves the efficiency of transferring a file by increasing the transfer speed (up to 60 kbps) while providing greater immunity to RF interference. Burst stalling and adjustable retry mechanisms are introduced to allow greater flexibility of host data source and sink rates. Note that the peer device must also support this feature to achieve higher transfer rates. This feature is backward compatible with existing burst transfer methods.

### 3.2.2 Single channel encryption

The ANT channel encryption engine provides the ability for a single channel to transmit and receive data in a secure manner using AES-128. This feature simplifies applications that require data security such as medical and health devices. Broadcast messaging, acknowledged messaging, and burst transfers/advanced burst transfers are supported data communication modes for encryption.

ANT encryption mode allows broadcast data to be received by multiple receivers. Alternatively, point-to-point encrypted links can be created through the use of inclusion/exclusion encryption ID list.

### 3.2.3 Multiple network keys

For use cases that benefit from transmission of data on multiple networks, the nRF51422 will support use of up to 8 public, private, and/or managed (ANT+) network keys.

### 3.2.4 Event Filtering

Event Filtering is provided for the application to avoid or reduce processing of ANT events, in an effort to conserve power consumption and/or resources. The application can select which event messages to block from the ANT protocol layer to the application.

### 3.2.5 Selective Data Updates

Selective Data Updates is another feature intended to aid in managing power consumption of the application. The facility may be used when an application needs to process received messages only if the specified data has changed. This could apply to devices such as display units that update the display only when the displayed data has actually changed and are otherwise asleep. This feature can be enabled for Broadcast messages only or Broadcast and Acknowledged messages. This feature does not apply to Burst Transfers.

### 3.2.6 Asynchronous Transmission channel

Asynchronous Transmission mode introduces the ability for users to initiate transmissions that are not bounded by any specified periods or intervals. Asynchronous channels do not need to be “opened” and are simply initiated by sending data to the channel. Asynchronous channels may be used when user generated input on a device needs to cause data transmission to a receiver with minimal delay (eg. remote control applications). Receivers must employ scanning channels to effectively use this feature. Broadcast messaging, acknowledged messaging, and burst transfers are supported in this mode.

### 3.2.7 Fast Channel Initiation

Enabling the Fast Channel Initiation feature allows ANT synchronous transmission channels to start as soon as possible. The latency from the channel opening to transmission is reduced. This feature should only be used in scenarios where a device opens for a brief period of time and requires low latency.

# 4 SoC library

The SoC library ensures that application access to SoC hardware resources used by the SoftDevice happens in a controlled and secured manner. The following API's are available in S210.

| Feature | Notes |
|---|---|
| `nrf_mutex_` API | Atomic mutex API without the application needing to disable global interrupts. |
| `nrf_nvic_` API | Gives the application access to all NVIC features without corrupting SoftDevice configurations. |
| `nrf_rand_` API | Gives access to the true random number generator. |
| `nrf_power_` API | Safe power configuration interface. |
| `nrf_clock_` API | Safe clock configuration interface. |
| `nrf_app_event_wait` | Simple power management hook for the application. |
| `nrf_ppi_` API | Safe PPI access to dedicated application PPI channels. |

# 5 SoftDevice manager

The SoftDevice manager allows the application to perform top level management of the SoftDevice operation (such as one command enabling or disabling the SoftDevice operation).

| Feature | Notes |
|---|---|
| `nrf_softdevice_` API | Gives control of the SoftDevice state. |

# 6 S210 Resource requirements

The S210 SoftDevice resides in the lower part of the SoC code memory space. When enabled by the nRF_SoftDevice API it reserves RAM space for its operation. Furthermore, it will control access to hardware resources used by the SoftDevice that then only can be accessed by the application using the SoC library API's. This section gives an overview of the resource allocations made by the S210 SoftDevice.

## 6.1 Memory resource map and usage

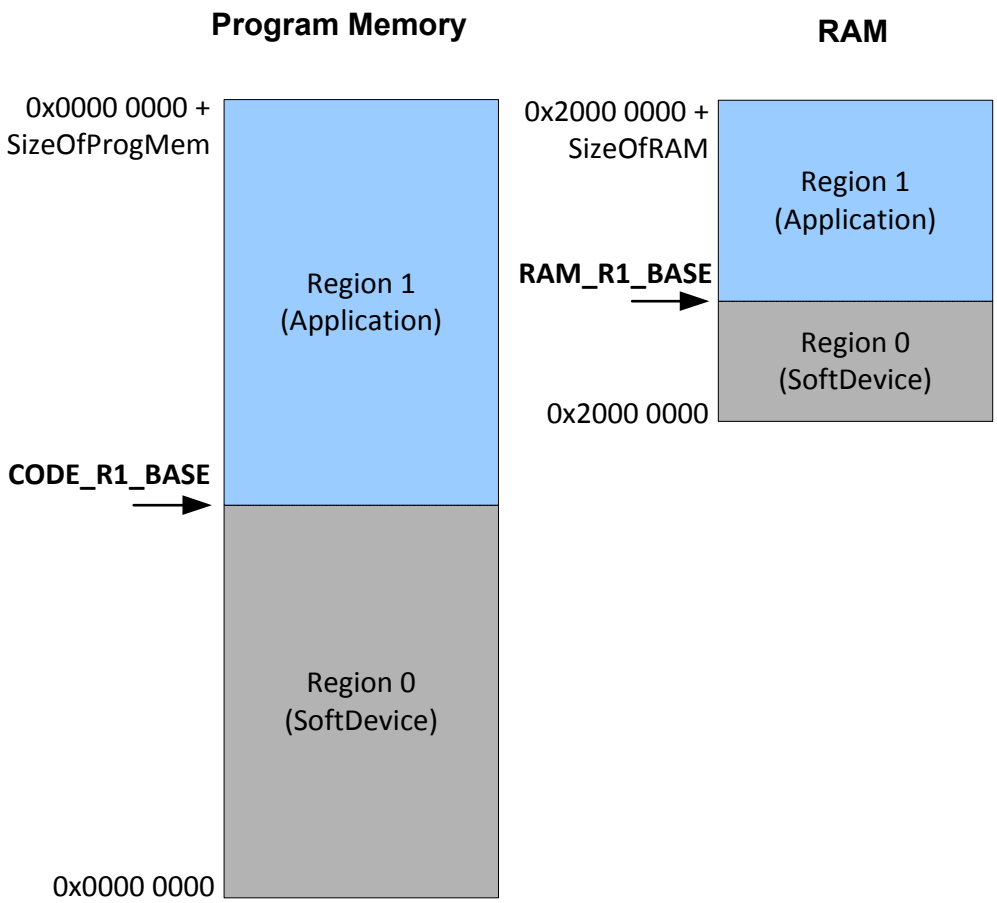



***Figure 3*** *Memory resource map*

*Table 1* shows the memory resources required by the SoftDevice for both enabled and disabled states, and the values of base addresses that must be used by the application program.

| Flash | S210 Enabled | S210 Disabled |
|---|---|---|
| Amount | 40 kB | 40 kB |
| CODE_R1_BASE | 0x0000 A000 | 0x0000 A000 |

| RAM | S210 Enabled | S210 Disabled |
|---|---|---|
| Amount | 2 kB | 0 kB |
| RAM_R1_BASE | 0x2000 0800 | 0x2000 0000 |

| Call stack | S210 Enabled | S210 Disabled |
|---|---|---|
| Maximum usage | 400 bytes | 0 bytes |

| Heap | S210 Enabled | S210 Disabled |
|---|---|---|
| Maximum allocated bytes | 0 bytes | 0 bytes |

***Table 1*** *Memory resource requirements*

## 6.2 Hardware blocks and interrupt vectors

| Access type | Definition |
|---|---|
| Restricted | Used by the SoftDevice and outside the application sandbox. Application has restricted access via the SoftDevice API. |
| Blocked | Used by the SoftDevice and outside the application sandbox. Completely inaccessible to the application. |
| Open | Unused by the SoftDevice and in the application sandbox. |

| Peripheral protection and usage by SoftDevice | | | | |
|---|---|---|---|---|
| **ID** | **Base address** | **Instance** | **Access (S210 enabled)** | **Access (S210 disabled)** |
| 0 | 0x40000000 | POWER | Restricted | Open |
| 0 | 0x40000000 | CLOCK | Restricted | Open |
| 1 | 0x40001000 | RADIO | Blocked | Open |
| 2 | 0x40002000 | UART0 | Open | Open |
| 3 | 0x40003000 | SPIM0/ TWI0 | Open | Open |
| 4 | 0x40004000 | SPI1/ TWI1 | Open | Open |
| .. | | | - | - |
| 6 | 0x40006000 | GPIOTE | Open | Open |
| 7 | 0x40007000 | ADC | Open | Open |
| 8 | 0x40008000 | TIMER0 | Open | 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 | Blocked | Open |
| .. | | | | |
| 16 | 0x40010000 | WDT | Open | Open |
| 17 | 0x40011000 | RTC1 | Open | Open |
| 18 | 0x40012000 | QDEC | Open | Open |
| .. | | | - | - |
| 20 | 0x40014000 | Software interrupt | Open | Open |
| 21 | 0x40015000 | Software interrupt | Open | Open |
| 22 | 0x40016000 | Software interrupt | Open | Open |
| 23 | 0x40017000 | Software interrupt | Restricted[1] | 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 | Open | Open |
| NA | 0x???? | NVIC | Restricted[2] | Open |

1. May be used by the application, but is also used for protocol to application events.
2. Not Sandbox protected. For robust system function, application must comply with the restriction.

***Table 2*** *Peripherals used by the SoftDevice*

### 6.2.1 PPI channels

| Range | User | Notes |
|---|---|---|
| 0 - 7 | Application | Available for the application |
| 8 - 15 | SoftDevice | Used by the SoftDevice |

### 6.2.2 PPI groups

| Range | User | Notes |
|---|---|---|
| 0 - 1 | Application | Application configurable through the PPI API |
| 2 - 3 | SoftDevice | Used by the SoftDevice |

### 6.2.3 SVC number ranges

| Range | User | Notes |
|---|---|---|
| 0x00 - 0x0F | Application | Always forwarded to the application |
| 0x10 - 0xFF | SoftDevice | Not forwarded to the application |

# 7 Performance

## 7.1 Interrupt latency

Latency, additional to ARM Cortex™-M0 hardware architecture latency, is introduced by SoftDevice logic to manage interrupt events.

| Interrupt | Additional latency in CPU cycles |
|---|---|
| Open peripheral interrupt | 50 |
| Blocked or restricted peripheral interrupt (only forwarded when SoftDevice disabled) | 63 |
| Application SVC interrupt | 14 |

## 7.2 Processor availability

***Figure 4*** illustrates the parameter values in ***Table 3*** on page 17. The SoftDevice architecture chapter of the *nRF51 Reference Manual* describes exception (interrupt) management in SoftDevices and is required knowledge for this section.

The parameters of interest are defined around LowerStack and UpperStack exceptions. These exceptions process real time protocol events and API calls (or deferred internal SoftDevice tasks) respectively.

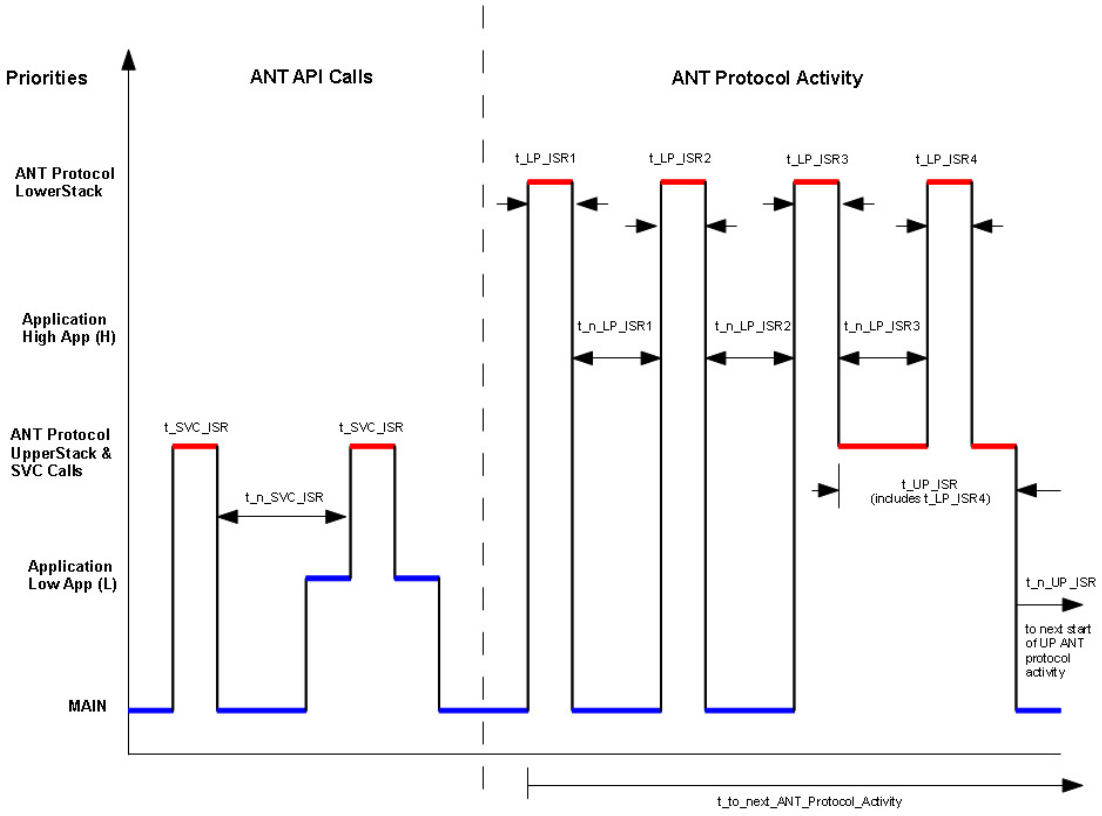



***Figure 4*** *Interrupt latencies due to SoftDevice processing*

### 7.2.1 LowerStack

| Parameter | Description | Min. | Typ. | Max. |
|---|---|---|---|---|
| t_LP_ISR1 | Maximum ANT protocol interrupt processing time | 65 µs | 1 | 110 µs |
| t_n_LP_ISR1 | Minimum time between ANT protocol interrupts | 73 µs | 1 | 118 µs |
| t_LP_ISR2 | Maximum ANT protocol interrupt processing time | 6 µs | 6 µs | 6 µs |
| t_n_LP_ISR2 | Minimum time between ANT protocol interrupts | 270 µs | 270 µs | 270 µs |
| t_LP_ISR3 | Maximum ANT protocol interrupt processing time | 55 µs | 1 | 140 µs |
| t_n_LP_ISR3 | Minimum time between ANT protocol interrupts | 155 µs | 1 | 280 µs |
| t_LP_ISR4 | Maximum ANT protocol interrupt processing time | 125 µs | 125 µs | 125 µs |

1. This data depends on ANT channel type and activity.

***Table 3*** *LowerStack interrupt latency numbers*

### 7.2.2 UpperStack

| Parameter | Description | Min. | Typ. | Max. |
|---|---|---|---|---|
| t_UP_ISR | Maximum ANT protocol interrupt processing time | 420 µs | 1 | 645 µs |
| t_n_UP_ISR | Minimum time between ANT protocol interrupts | 760 µs | 1 | Channel Period |
| t_SVC_ISR | Maximum ANT SVC call interrupt processing time | | 2 | |
| t_n_SVC_ISR | Minimum time between ANT SVC call interrupts | | 3 | |

1. This data depends on ANT channel type and activity.
2. This data depends on an SVC call.
3. This data depends on the Application.

***Table 4*** *UpperStack interrupt latency numbers*

Application Low, App(L), can be blocked for a maximum of:

$$App(L)_{latency_{\max}} = t_{\_UP\_ISR} + t_{\_LP\_ISR3}$$

Application High, App(H), can be blocked for a maximum of:

$$App(H)_{latency_{\max}} = \max(t_{\_LP\_ISRx})\ where\ x = 1,2,3,4$$

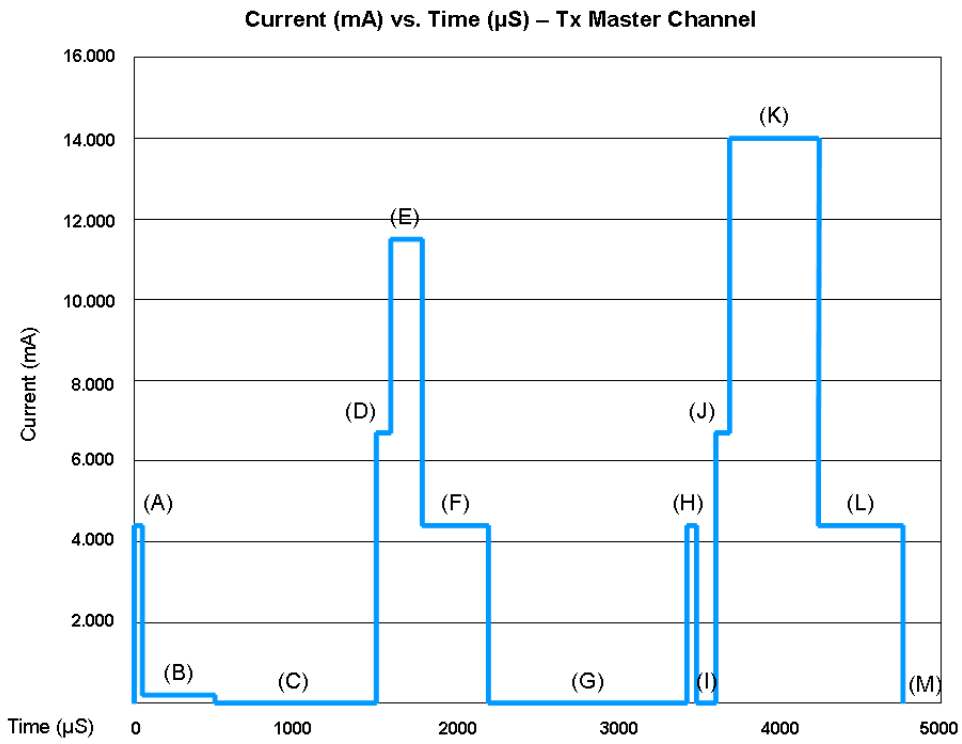
**Restrictions:**

- The ANT protocol on the nRF51 has been designed to allow time critical applications to be executed in application high (H) state in the presence of ANT protocol activity.
- Short application high (H) duration is required for minimal impact to certain ANT protocol modes (especially burst transfers).

# 8 Power profiles

These profiles give a detailed overview of the stages of a radio event, the approximate timing of stages within the event, and how to calculate the peak current at each stage.

## 8.1 TX master channel

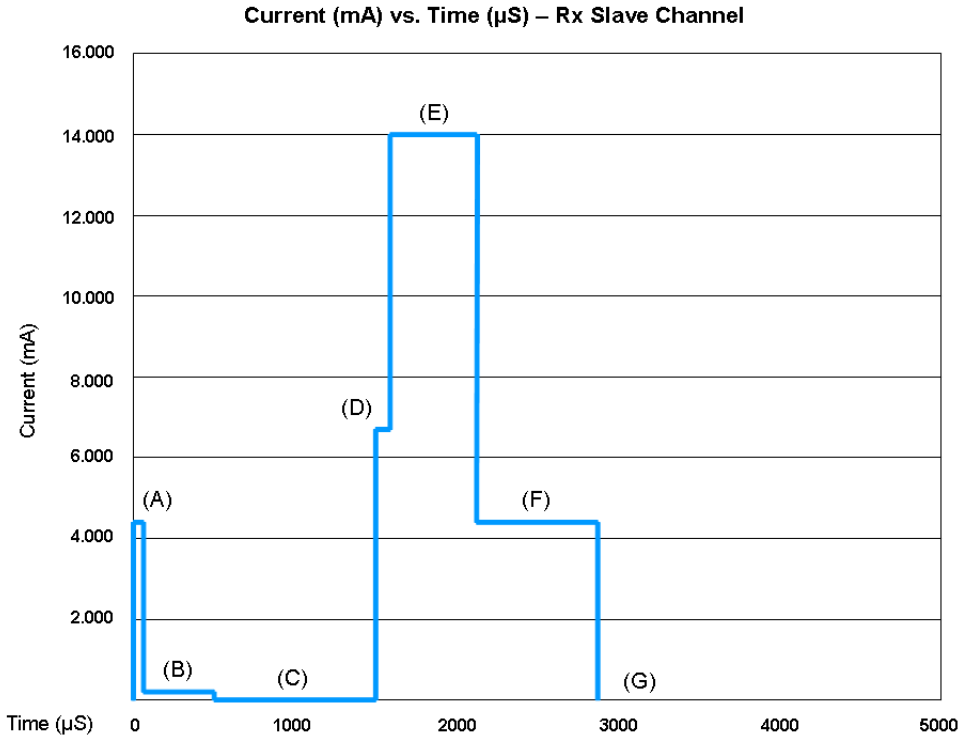



| Stage | Description | Current Calculations[1] |
|---|---|---|
| (A) and (H) | Preprocessing | $I_{RTC} + I_{X32k} + I_{CPU,Flash}$ |
| (B) | Standby + XO ramp | $I_{RTC} + I_{X32k} + I_{START,X16M}$ |
| (C), (G) and (I) | Standby | $I_{RTC} + I_{X32k} + I_{X16M}$ |
| (D) | Radio Start | $I_{RTC} + I_{X32k} + I_{X16M} + \int (I_{START,TX})$ |
| (E) | Radio TX | $I_{RTC} + I_{X32k} + I_{X16M} + I_{TX,0dBm}$ |
| (F) | Post-processing | $I_{RTC} + I_{X32k} + I_{X16M} + I_{CPU,Flash}$ |
| (J) | Radio turn-around/Start | $I_{RTC} + I_{X32k} + I_{X16M} + \int (I_{START,TX})$ |
| (K) | Radio RX | $I_{RTC} + I_{X32k} + I_{X16M} + I_{RX}$ |
| (L) | Post-processing | $I_{RTC} + I_{X32k} + I_{CPU,Flash}$ |
| (M) | Idle - TX channel | $I_{RTC} + I_{X32k}$ |

1. See the nRF51422 Product Specification for the symbol values.

**Note:** When using the 32k synthesized clock you should replace $I_{X32k}$ with $I_{SYNT32k}$.

## 8.2 RX channel




| Stage | Description | Current Calculation[1] |
|---|---|---|
| (A) | Pre-processing | $I_{RTC} + I_{X32k} + I_{CPU,Flash}$ |
| (B) | Standby + XO ramp | $I_{RTC} + I_{X32k} + I_{START,X16M}$ |
| (C) | Standby | $I_{RTC} + I_{X32k} + I_{X16M}$ |
| (D) | Radio start | $I_{RTC} + I_{X32k} + I_{X16M} + \int (I_{START,RX})$ |
| (E) | Radio RX | $I_{RTC} + I_{X32k} + I_{X16M} + I_{RX}$ |
| (F) | Post-processing | $I_{RTC} + I_{X32k} + I_{CPU,Flash}$ |
| (G) | Idle - RX channel | $I_{RTC} + I_{X32k}$ |

1. See the nRF51422 Product Specification for the symbol values.

**Note:** When using the 32k synthesized clock you should replace $I_{X32k}$ with $I_{SYNT32k}$.

# 9 SoftDevice compatibility and selection

## 9.1 S210 SoftDevice identification and version scheme

S210 SoftDevice will be identified by the S210 part code, a qualified device part code (e.g. nRF51422), and a version number consisting of major, minor, and revision numbers.

For example: S210_nRF51422_1.2.3

***Table 5*** outlines how version numbers will increment.

| Revision | Description |
|---|---|
| Major increments | Modifications to the API or the function or behavior of the implementation or part of it have changed.<br><br>Changes as per minor increment may have been made.<br><br>Application code will not be compatible without some modification. |
| Minor increments | Additional features and/or API calls are available.<br><br>Changes as per revision increment may have been made.<br><br>Application code may be modified to take advantage of new features. |
| Revision increments | Issues have been resolved or improvements to performance implemented.<br><br>Existing application code will not require any modification. |

***Table 5*** *Revision scheme*

Additionally, for revisions of the SoftDevice which are not production qualified, the qualification level, alpha or beta, and sequence number will be appended. The following are examples.

- An alpha qualified revision: S210 nRF51422 1.2.3alphaX
- The same SoftDevice with beta qualification: S210 nRF51422 1.2.3betaY
- The same SoftDevice with production qualification: S210 nRF51422 1.2.3

Test qualification levels are outlined in ***Table 6***.

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

***Table 6*** *Test qualification levels*

## 9.2 nRF514xx device compatibility and qualification

S210 SoftDevice is compatible with one or more nRF514xx devices and is qualified for mass production with some of these devices. The following is a list of S210 versions versus compatible devices and qualification status current at the time of publication of this document.

| SoftDevice | Compatible IC | Qualification |
|---|---|---|
| S210_nRF51422_2.0.0 | nRF51x22 | Production |
| S210_nRF51422_1.0.1.alpha | nRF51x22 | Alpha |