



nRF51422 Evaluation Kit

nRF51422

User Guide v1.1

Contents

1	Introduction.....	3
1.1	Minimum requirements	3
1.2	External resources	3
1.3	Writing conventions	3
1.4	Evaluation kit release notes	3
2	Kit content.....	4
2.1	nRF51422 Evaluation Kit hardware content	4
2.2	Downloadable content	5
3	Quick start	6
3.1	Install the nRF51422 Evaluation board (PCA10003)	7
3.2	ANT Link demo	8
4	Evaluation kit configuration.....	10
4.1	Development environment.....	10
5	Hardware description	12
5.1	nRF51422 Evaluation Kit board (PCA10003).....	12
5.2	nRFready ANT USB dongle (nRF2779)	20
6	Flash programming and application development.....	22
6.1	Programming and erasing flash using nRFgo Studio	22
6.2	Application development	25
7	Debugging the nRF51422 chip	33
7.1	nRF51422 debug features and precautions	34
8	Software Development Kit	38
8.1	Installing the nRF51 SDK.....	38
9	Troubleshooting	39
	Appendix A: Installing drivers and configuring KEIL projects for the SEGGER debugger	40

1 Introduction

The nRF51422 ANT™ Evaluation Kit (EK) provides a complete solution for testing and evaluating the nRF51422 device. The nRF51422 is part of the nRF51 series which offers a range of ultra-low power, System on Chip (SoC) solutions for your 2.4 GHz wireless products.

1.1 Minimum requirements

- nRFgo Studio v1.14 or later
- Computer with a minimum of 2 USB ports
- Windows XP or Windows 7

1.2 External resources

- Keil MDK-ARM Lite v4.54 or later <https://www.keil.com/demo/eval/arm.htm>
- J-Link Software v4.52b or later <http://www.segger.com/jlink-software.html>

1.3 Writing conventions

This User Guide follows a set of typographic rules that makes the document consistent and easy to read. The following writing conventions are used:

- Commands are written in `Lucida Console`.
- Pin names are written in `Consolas`.
- File names and user interface components are written in **bold**.
- Internal cross references are italicized and written in ***semi-bold***.

1.4 Evaluation kit release notes

Date	Kit version	Description
January 2013	2.0	Fixed known issue in kit v1.0.
November 2012	1.0	Known issues PCA10003 v1.0: <ul style="list-style-type: none">• The antenna matching network and layout on these boards is suitable for applications using TX output power 0 dBm or less. These boards are not suitable for applications using +4 dBm TX output power.

2 Kit content

The nRF51422 Evaluation Kit consists of hardware and access to software components, documentation, and design files from www.nordicsemi.com. The S210 nRF51422 SoftDevice is pre-programmed on the chip and cannot be deleted or modified.

2.1 nRF51422 Evaluation Kit hardware content

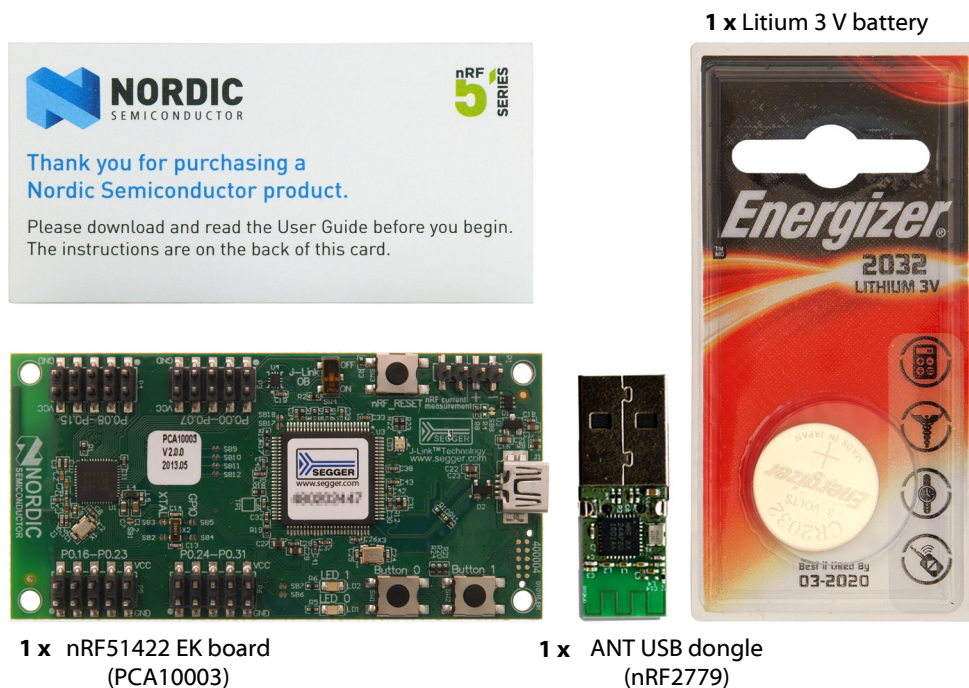


Figure 1 nRF51422 Evaluation Kit hardware content

2.2 Downloadable content

The nRF51422 Evaluation Kit includes firmware source code, documentation, hardware schematics, and layout files. To access this information, log in to your My Page account, enter your product key, and download the files. Instructions can be found in *chapter 3 on page 6*.

2.2.1 nRF51422 software

- nRFgo Studio
- nRF51 Software Development Kit (SDK)
 - Precompiled HEX files
 - Source code
 - Keil ARM project files
- ANTware II

2.2.2 nRF51422 documentation

- *nRF51422 Evaluation Kit User Guide*
- *nRF51 Series Reference Manual*
- *nRF51422 PS*
- *S210 nRF51422 SoftDevice Specification*
- *nRF51422 PAN*
- ANT API documentation

2.2.3 Schematics, Bill of Materials, PCB layout files, and production files

The ZIP file and its subdirectories contain the hardware design files for the Evaluation Kit.

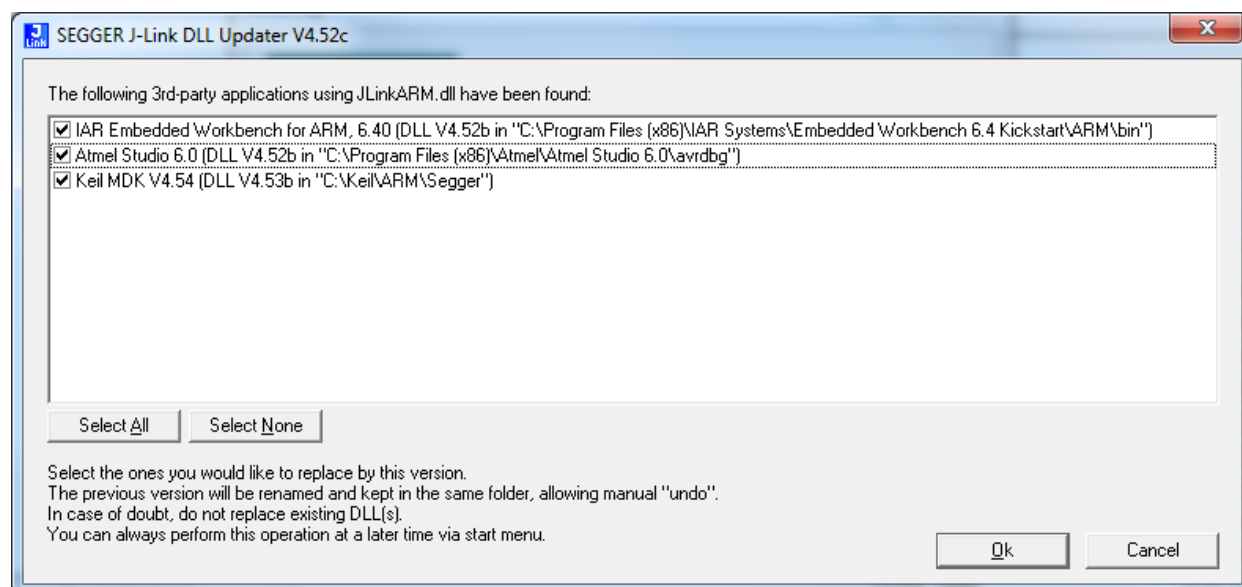
- Altium Designer files
- Schematics
- PCB layout files
- Production files
 - Assembly drawings
 - Drill files
 - Gerber files
 - Pick and Place files
 - Bill of Materials

3 Quick start

This section shows you how to set up the nRF51422 Evaluation Kit and provides example applications to help you start programming your device.

Register, download, and install

1. If you have Keil MDK-ARM Lite already installed, go to step 2. Otherwise, download and install Keil MDK-ARM Lite from <https://www.keil.com/demo/eval/arm.htm> to your hard drive. Keil downloads to **c:\Keil** unless you change the location when installing.
2. Download and run the J-Link Software and documentation pack for Windows from <http://www.segger.com/jlink-software.html>. The serial number from your SEGGER J-Link hardware is needed to identify your device.
3. During installation you will be prompted to select the IDE that should be updated with the latest SEGGER DLLs. Check the box for **Keil MDK** and any other IDEs you want to use with SEGGER.

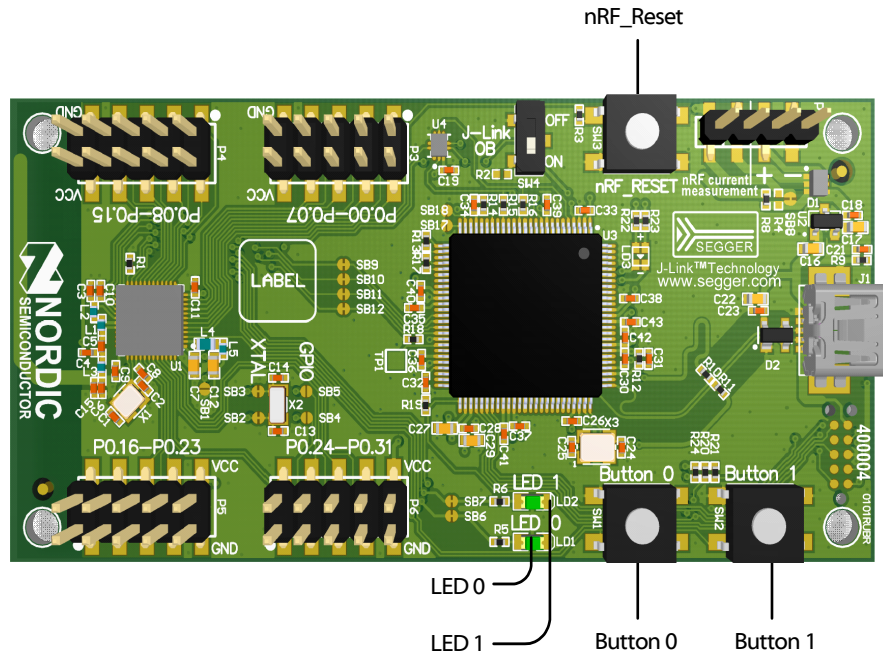


4. Go to www.nordicsemi.com and log in to your Nordic My Page account.
5. Select **My Products** from the left menu. This takes you to the My Products page.
6. Enter the product key (included with this kit) into the **Product Key** field and click **Add**.
7. From the **Add product** box, select the product name and click **Add**.
8. Click the **Downloads** link in the **Overview, My Products** table.
9. Download and run the nRF514 SDK installer. Make sure to choose the **Keil MDK-ARM** installer option.

3.1 Install the nRF51422 Evaluation board (PCA10003)

Connect the hardware

1. Connect a USB cable from the PCA10003 board to your computer.



3.2 ANT Link demo

Download and install ANTware II

1. Go to www.nordicsemi.com and log in to your Nordic My Page account.
2. Download and install ANTware II.

Plug in the ANT dongle

1. Plug the ANT dongle (nRF2779) into your computer with a USB cable. The driver installation will start automatically.

Start the ANT TX broadcast example

1. Go to the ANT Broadcast TX project found under <keilpath>
\ARM\Device\Nordic\nrf51422\Board\pca10003\ant\ant_broadcast\broadcast_tx\arm.
2. Open the project in Keil μ Vision by double clicking the **ant_broadcast_tx.uvproj** file.
3. Click the **Build** icon or press **F7** to compile the project.
4. Click the **Load** icon to download and run the Broadcast example firmware.

Note: **LED0** indicates ANT is transmitting and **LED1** indicates that the device is idle.

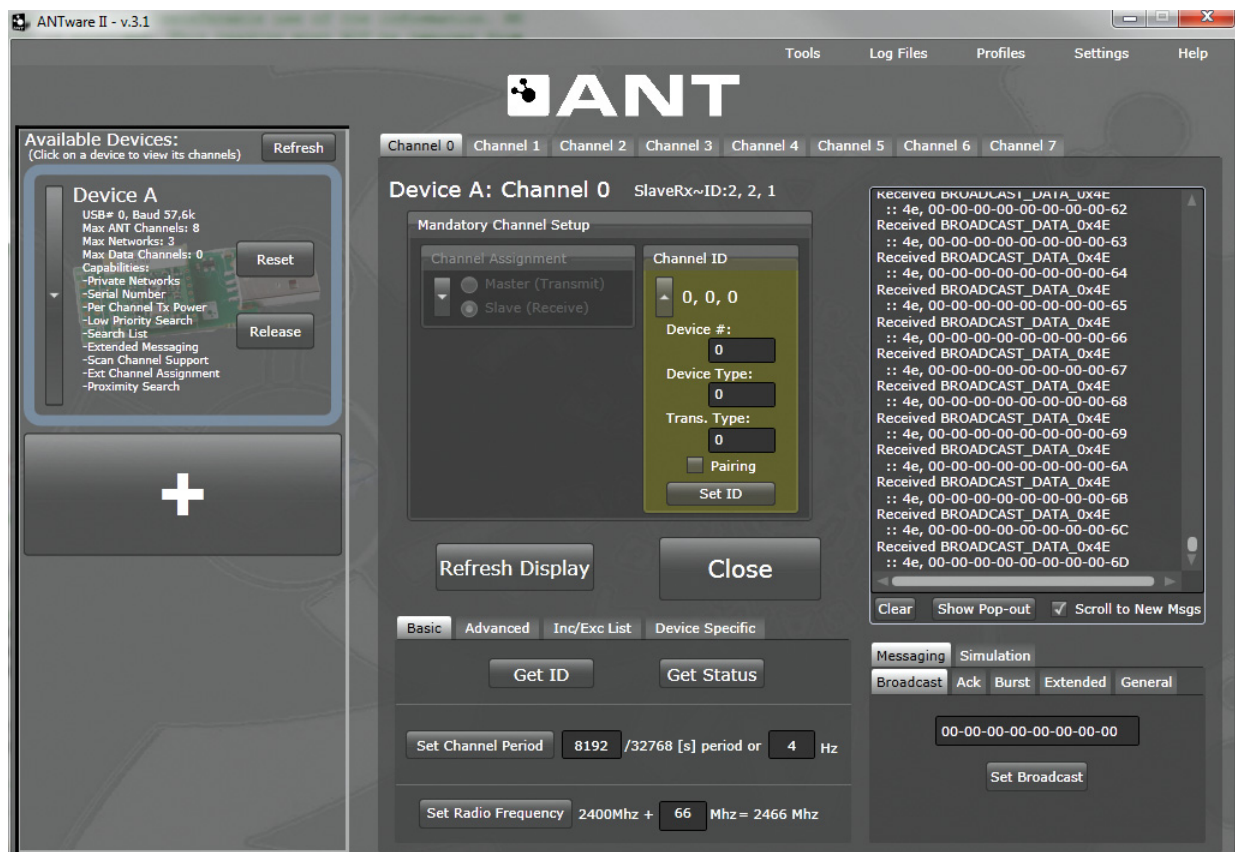
Start ANTware II and configure the ANT channel

Below is an example of how to set the Slave setting in ANTware II.

1. Select **ANT dongle** (Device A).
2. Set Channel Assignment to **Slave**.
3. In the Channel ID window, set **Device #** to **0**, set **Device Type** to **0**, and set **Trans. Type** to **0**.
4. Click **Auto-Open**.

Monitor the traffic in the ANTware II

Received Broadcast Data is shown in the feedback window, showing packets coming from the ANT Broadcast TX. A counter is added on the two last digits and will increase by one for each packet received.



4 Evaluation kit configuration

This chapter shows where to download third party content and how the development environment is set up.

4.1 Development environment

ARM compiler/IDE (not included in this kit)

All the source code projects and examples can be compiled and used with the Keil Microcontroller Development Kit (MDK). For full use of the Development Kit source code projects, and to upgrade firmware, download and install the free KEIL MDK-ARM Lite from <https://www.keil.com/demo/eval/arm.htm>.

J-Link OB driver (not included in this kit)

For installing drivers for the integrated SEGGER chip, visit <http://www.segger.com/jlink-software.html>. You will be asked to enter your SEGGER serial number before the download will begin. You must correctly install the drivers for the device to use the J-Link debugger with Keil MDK. See **Appendix A: “Installing drivers and configuring KEIL projects for the SEGGER debugger” on page 40.**

4.1.1 Development environment setup

The nRF51422 device can be programmed from several environments. This section shows the development setup using Keil MDK-ARM.

The nRF51422 EK can be configured for standard ANT profile-based applications. For development or demonstration of standard ANT profile-based applications, you need:

- 1 x nRF51422 Evaluation board (PCA10003)
- 1 x nRF2779 (AP2) dongle
- ANTware II PC software

Figure 2 on page 11 show the relationship between the hardware and software components, and the Evaluation board.

Note: The Keil μ Vision IDE is not included in the kit content.

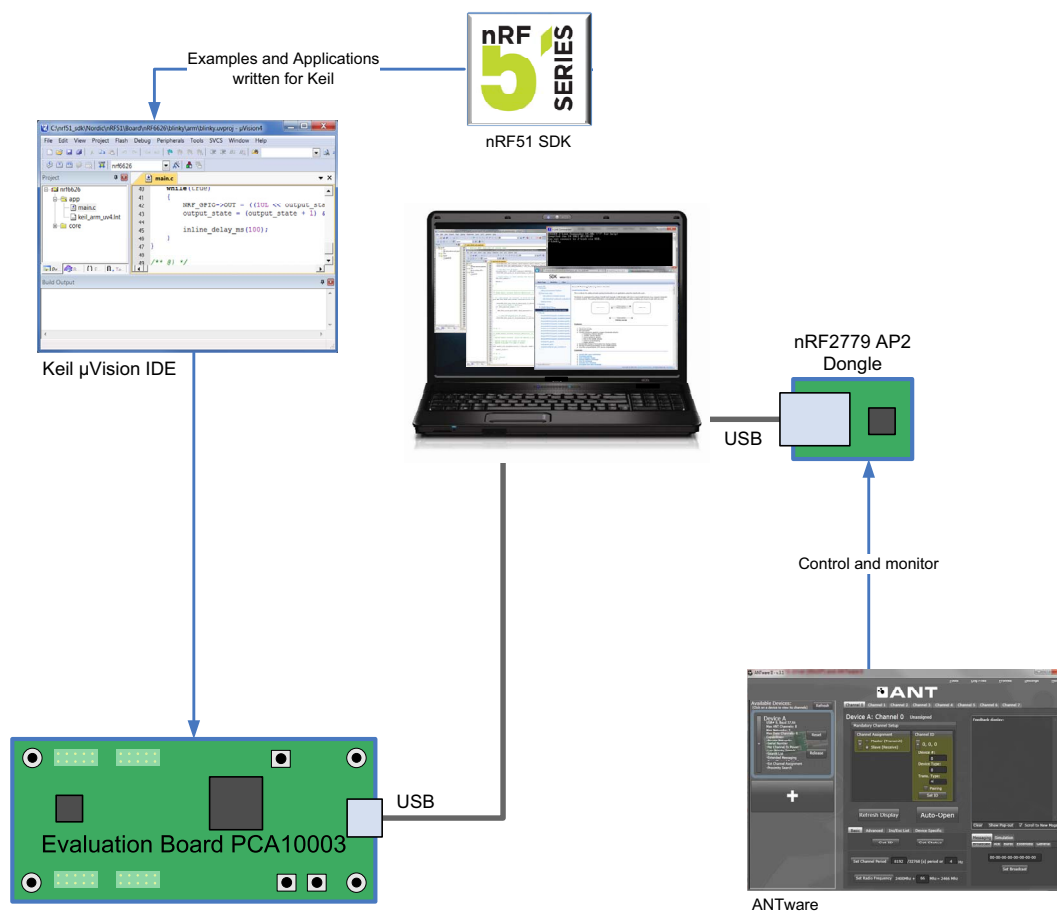


Figure 2 nRF51422 Evaluation Kit configuration

5 Hardware description

This chapter describes the nRF51422 Evaluation Kit hardware.

5.1 nRF51422 Evaluation Kit board (PCA10003)

The nRF51422 Evaluation Kit board (PCA10003) is a standalone nRF51422 ANT evaluation board with an integrated debugger from SEGGER.

Note: It is not possible to erase or upgrade the SoftDevice on this chip.

5.1.1 Key features

The nRF51422 EK board (PCA10003) has the following key features:

- nRF51422 flash based ANT™ SoC solution
- Buttons and LEDs for user interaction
- I/O interface for plug-in modules
- SEGGER J-Link OB programming and debugging capabilities
- USB to UART bridge

5.1.2 Hardware pictures

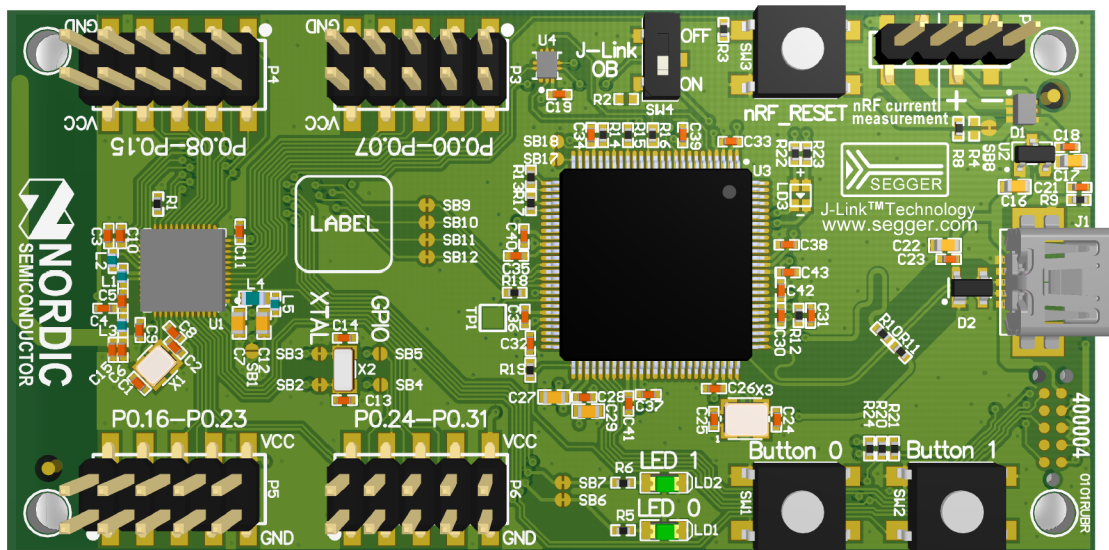


Figure 3 PCA10003 top

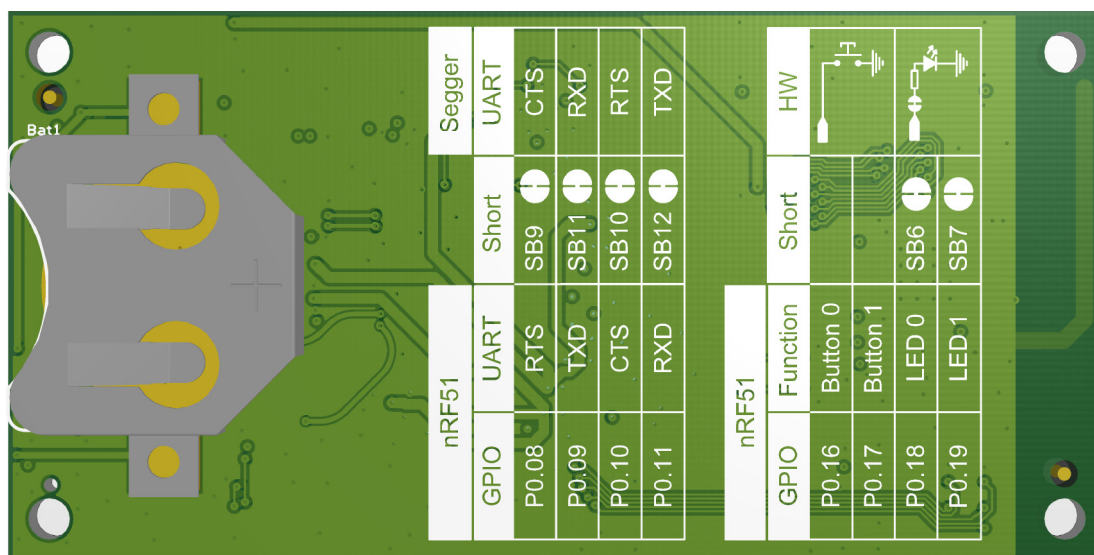


Figure 4 PCA10003 bottom

5.1.3 Block diagram

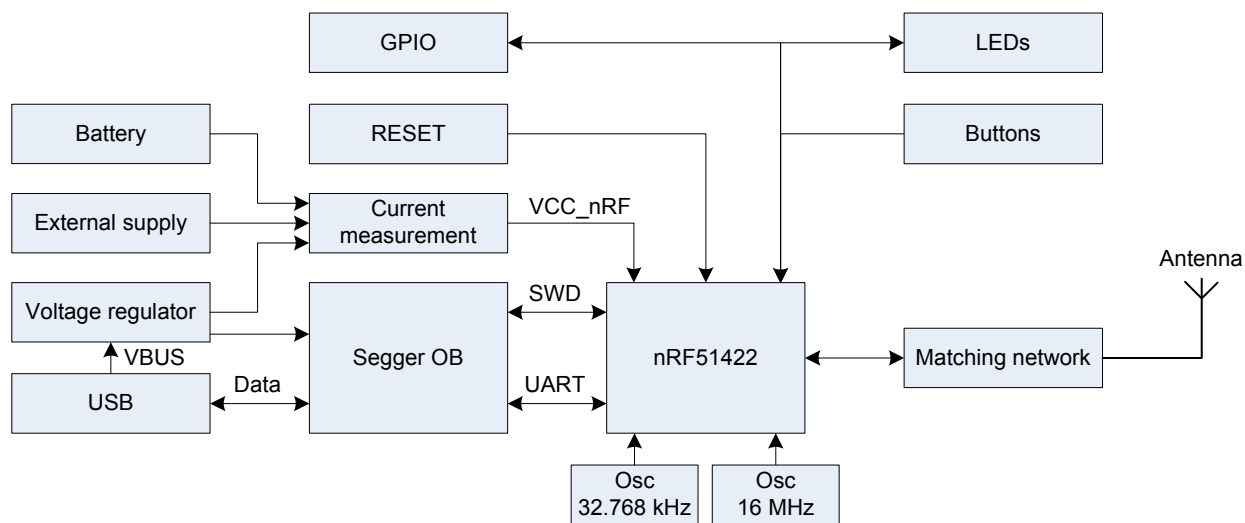


Figure 5 PCA10003 module block diagram

5.1.4 Reset button

The EK board (PCA10003) is equipped with a reset button (**SW3**) for the nRF51422. When debugging the nRF51422 using the J-Link OB, you should use the reset functionality built into the computer software.

5.1.5 Power supply

The EK board (PCA10003) has several power options:

- USB (see *Figure 6*)
- External power supply through **P1** (1.8 V to 3.6 V)
- CR2032 coin cell battery (see *Figure 7*)

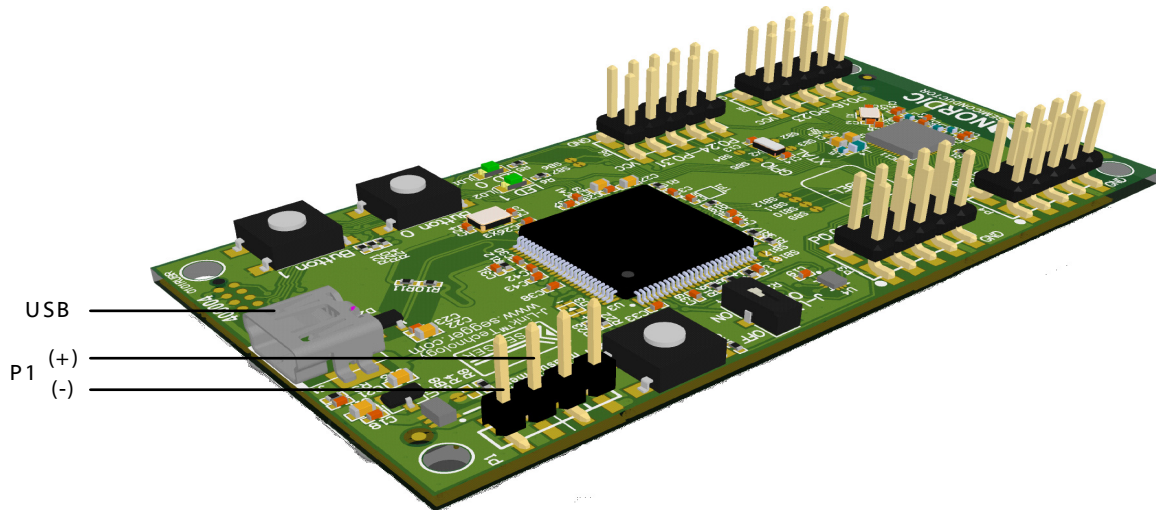


Figure 6 USB and external power supply

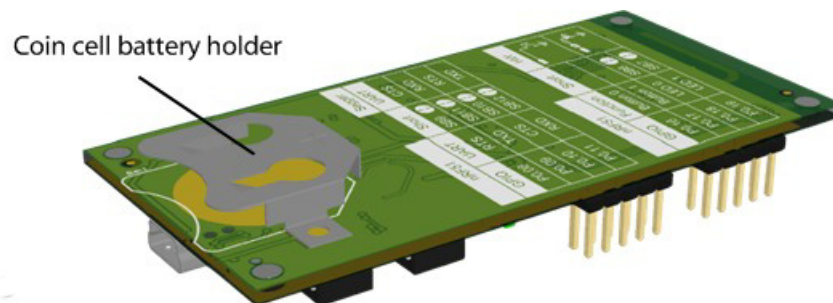


Figure 7 Coin cell battery supply

The 5 V from the USB is regulated down to 3.3 V through an on-board voltage regulator. The battery and external power supply is not regulated. The power sources are routed through a set of diodes (D1A, D1B, and D1C), where the circuit is supplied from the source with the highest voltage.

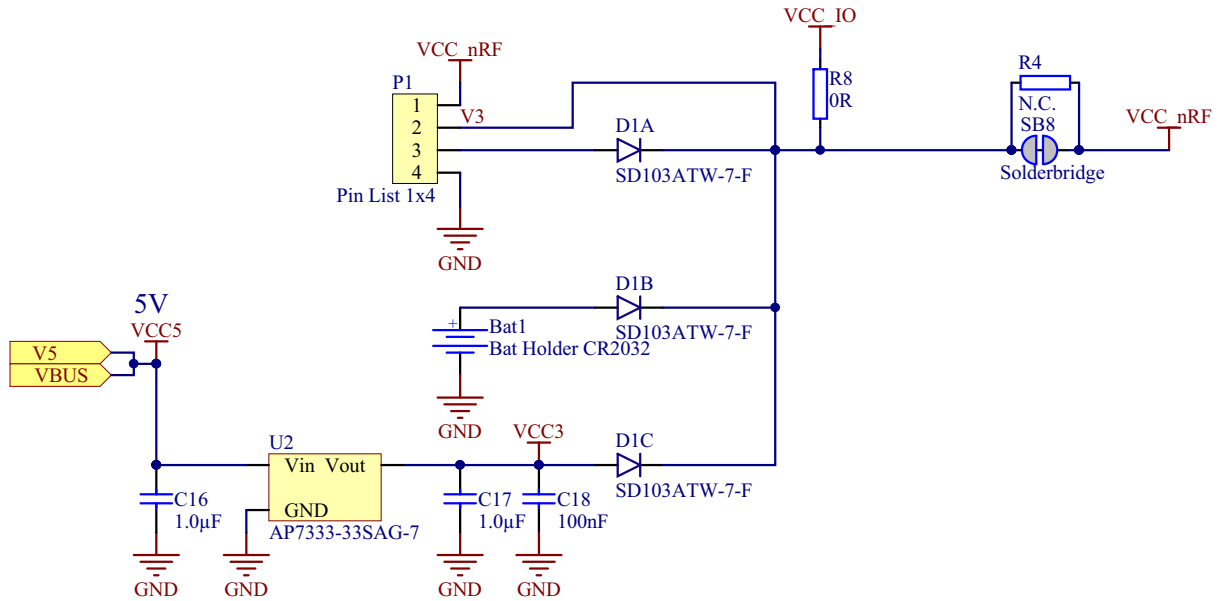


Figure 8 Power supply circuitry

5.1.6 SEGGER SWD disconnect

The SEGGER J-Link OB circuit on the EK board (PCA10003) only works when the board is powered through the USB connector. To ensure that the J-Link OB will not hold any of the SWD lines while powered down, the SWD lines will be disconnected automatically when the USB cable is unplugged. However, to disconnect the SWD lines manually, the board is equipped with a switch (**SW4**) that disconnects the lines even if the board is powered through the USB, see **Figure 9**.

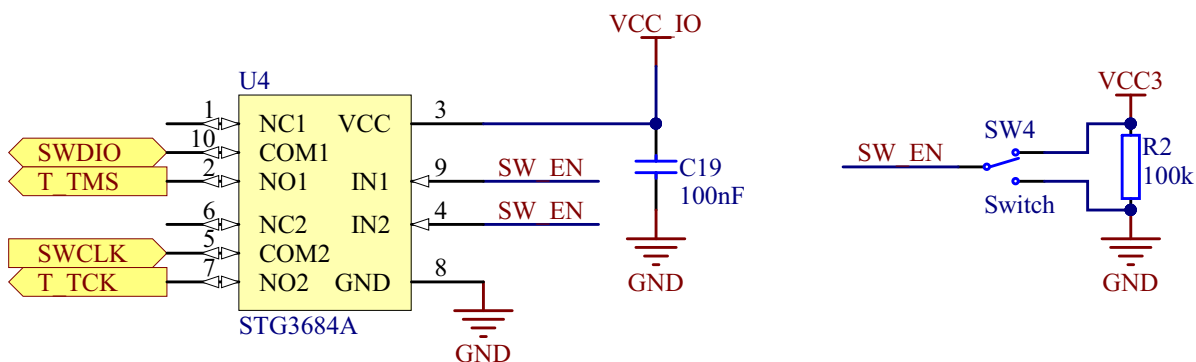


Figure 9 SWD disconnect logic

Note: To program or debug the nRF51422 using the SEGGER J-Link OB, the **SW4** must be switched to **ON**.

5.1.7 GPIO interface

Access to the nRF51422 GPIOs is available at connectors **P3**, **P4**, **P5**, and **P6** on the EK board PCA10003.

P3			P6				
P0.00	1	2	P0.01	VCC	10	9	GND
P0.02	3	4	P0.03		8	7	P0.30
P0.04	5	6	P0.05	P0.29	6	5	P0.28
P0.06	7	8	P0.07	P0.27	4	3	P0.26
GND	9	10	VCC	P0.25	2	1	P0.24

P4			P5				
P0.08	1	2	P0.09	VCC	10	9	GND
P0.10	3	4	P0.11	P0.23	8	7	P0.22
P0.12	5	6	P0.13	P0.21	6	5	P0.20
P0.14	7	8	P0.15	P0.19	4	3	P0.18
GND	9	10	VCC	P0.17	2	1	P0.16

Figure 10 PCA10003 GPIO pin headers

Note: Some pins have default settings.

- P0.26 and P0.27 are by default used for the 32 kHz crystal and are not available on the **P6** connector. Please see [section 5.1.9 on page 18](#) for more information.
- P0.16, P0.17, P0.18, and P0.19 are by default connected to the buttons and LED. Please see [section 5.1.8](#) for more information.
- P0.08, P0.09, P0.10, and P0.11 are by default used by the UART. Please see [section 5.1.10 on page 18](#) for more information.

5.1.8 Buttons and LEDs

The two buttons and two LEDs on EK board PCA10003 are connected to dedicated I/Os on the nRF51422 chip. The connections are shown in *Table 1*.

Part	GPIO	Short
Button 0	P0.16	
Button 1	P0.17	
LED 0	P0.18	SB6
LED 1	P0.19	SB7

Table 1 Button and LED connection

If GPIO P0.18 and P0.19 are needed elsewhere, the LEDs can be disconnected by cutting the short on **SB6** and **SB7**, see *Figure 11*.

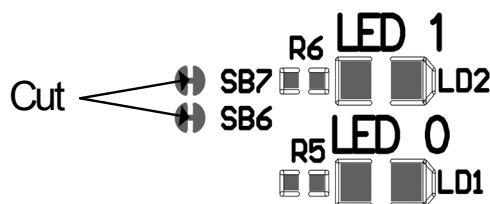


Figure 11 Disconnecting the LEDs

The buttons are active low meaning the input will be connected to ground when the button is activated. The buttons have no external pull-up resistor, so to use the buttons the P0.16 and P0.17 pins must be configured as an input with internal pull-up resistor.

The LEDs are active high, meaning that writing a logical one ('1') to the output pin will illuminate the LED.

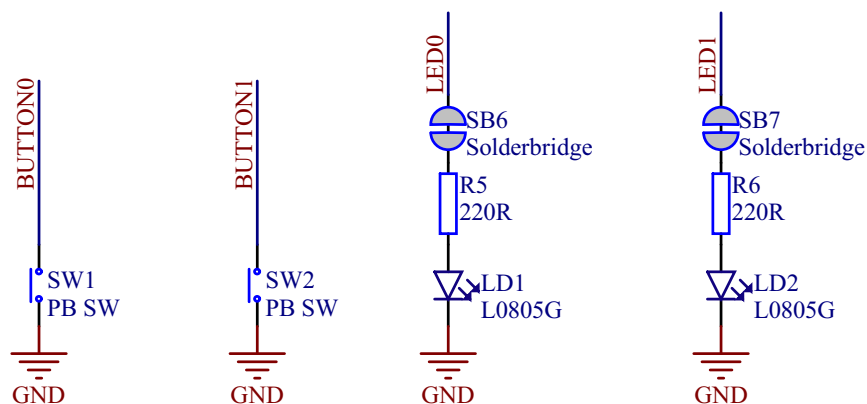


Figure 12 Button and LED configuration

5.1.9 32.768 kHz crystal

nRF51422 can use an optional 32.768 kHz crystal (X2) for higher accuracy and lower average power consumption. On the EK board PCA10003 module, P0.26 and P0.27 are by default used for the 32.768 kHz crystal and are not available as a GPIO on the **P6** connector.

If P0.26 and P0.27 are needed as normal I/Os the 32.768 kHz crystal can be disconnected and the GPIO routed to the **P6** connector. Cut the shorting track on **SB2** and **SB3**, and solder **SB4** and **SB5**. See **Figure 13** on page 18 for reference.

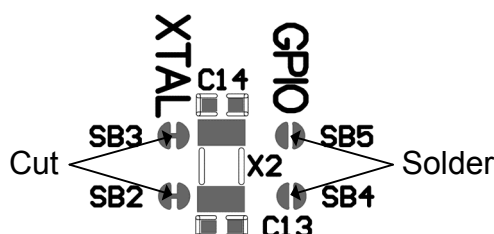


Figure 13 Disconnecting 32.768 kHz crystal and connecting P0.26 and P0.27 to P6

5.1.10 UART configuration

Table 2 shows an overview of the UART connections on nRF51422 and the SEGGER IC.

nRF51422		SEGGER IC	
Default GPIO	UART	Short	UART
P0.08	RTS	SB9	CTS
P0.09	TXD	SB11	RXD
P0.10	CTS	SB10	RTS
P0.11	RXD	SB12	TXD

Table 2 Relationship of UART connections on nRF51422 and SEGGER

The UART signals by default are routed directly to the SEGGER chip. To use pins P0.08 to P0.11 for other purposes than UART to the SEGGER chip, the shorting of the solder bridges should be removed. See **Figure 14**.

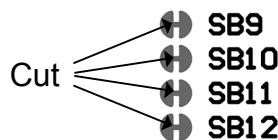


Figure 14 Disconnecting UART lines

Note: In order to use the USB to UART bridge, the software on the nRF51422 must enable flow control. For details on how to set up the UART with flow control see the *nRF51 Series Reference Manual*.

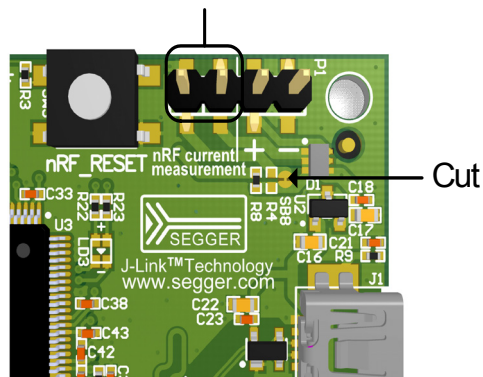
5.1.11 Measuring current

The current drawn by the nRF51422 device can be monitored on the PCA10003. To measure the current, you must first prepare the board by cutting the shorting of solder bridge **SB8**.

There are two ways of measuring the current consumption:

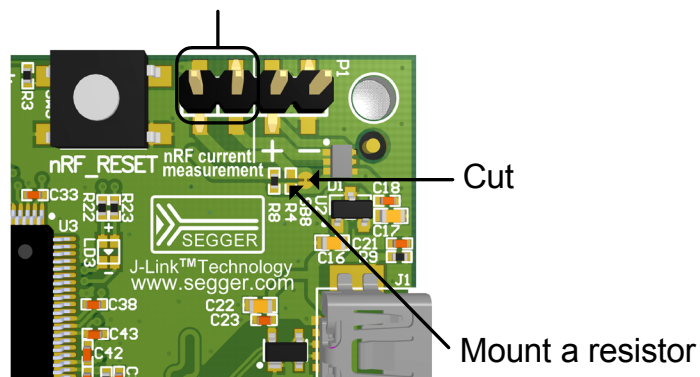
1. Connect an ampere-meter between pin 1 and pin 2 of connector **P1**. This will monitor the current directly.

Connect an ampere-meter



2. Mount a resistor on the footprint for **R4**. The resistor should not be larger than 10 Ω . Connect an oscilloscope or similar with two probes on pin 1 and pin 2 on the **P1** connector and measure the voltage drop. The voltage drop will be proportional with the current consumption. For example, if a 1 Ω resistor is chosen, 1 mV equals 1 mA.

Connect an oscilloscope



5.2 nRFready ANT USB dongle (nRF2779)

The nRFready ANT USB dongle (nRF2779) enables you to send data to nRF51422 as well as monitor the data sent from nRF51422. Software and drivers for the ANT USB dongle install automatically, but are also available for download from the nRFready ANT USB dongle product page at www.nordicsemi.com. The ANT USB dongle combined with ANTware II gives you a peer device for nRF51422 that you can use to test the wireless connection.

5.2.1 Key features

The nRF2779 has the following key features:

- nRF24AP2 IC
- ANT™ compatible

5.2.2 Hardware pictures

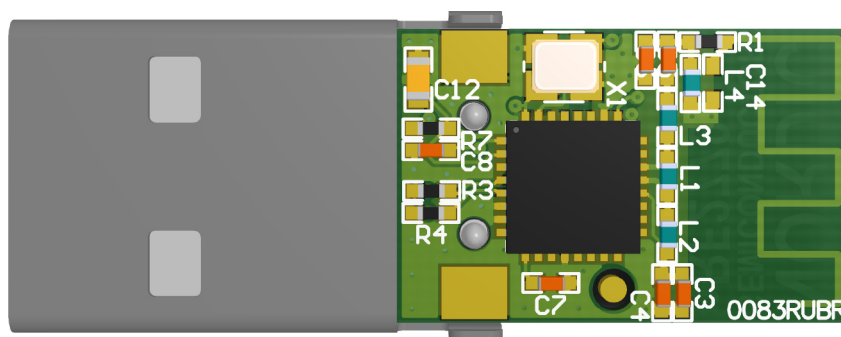


Figure 15 nRF2779 top side

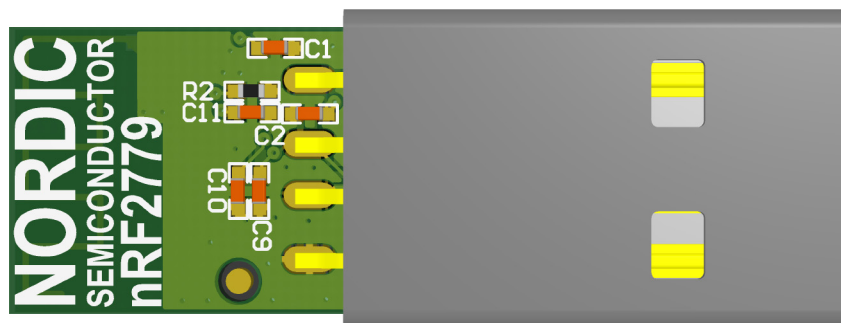


Figure 16 nRF2779 bottom side

5.2.3 Block diagram

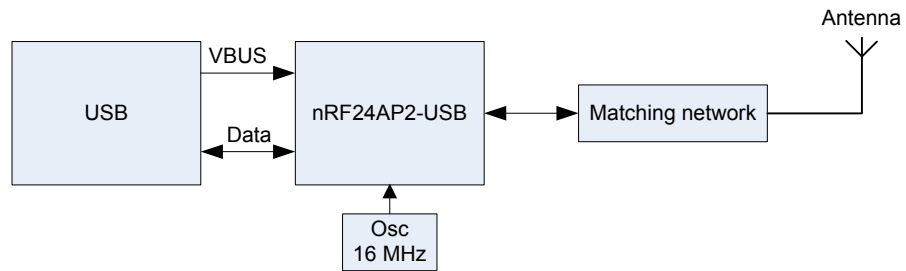


Figure 17 nRF2779 block diagram

6 Flash programming and application development

The nRF51422 chip is shipped with the SoftDevice pre-programmed. It is not possible to upgrade or erase the SoftDevice on nRF51422.

6.1 Programming and erasing flash using nRFgo Studio

Use nRFgo Studio to program or erase an application HEX file onto the nRF51422 chip.

Note: For details on memory organization and protection see the *nRF51 Series Reference Manual*.

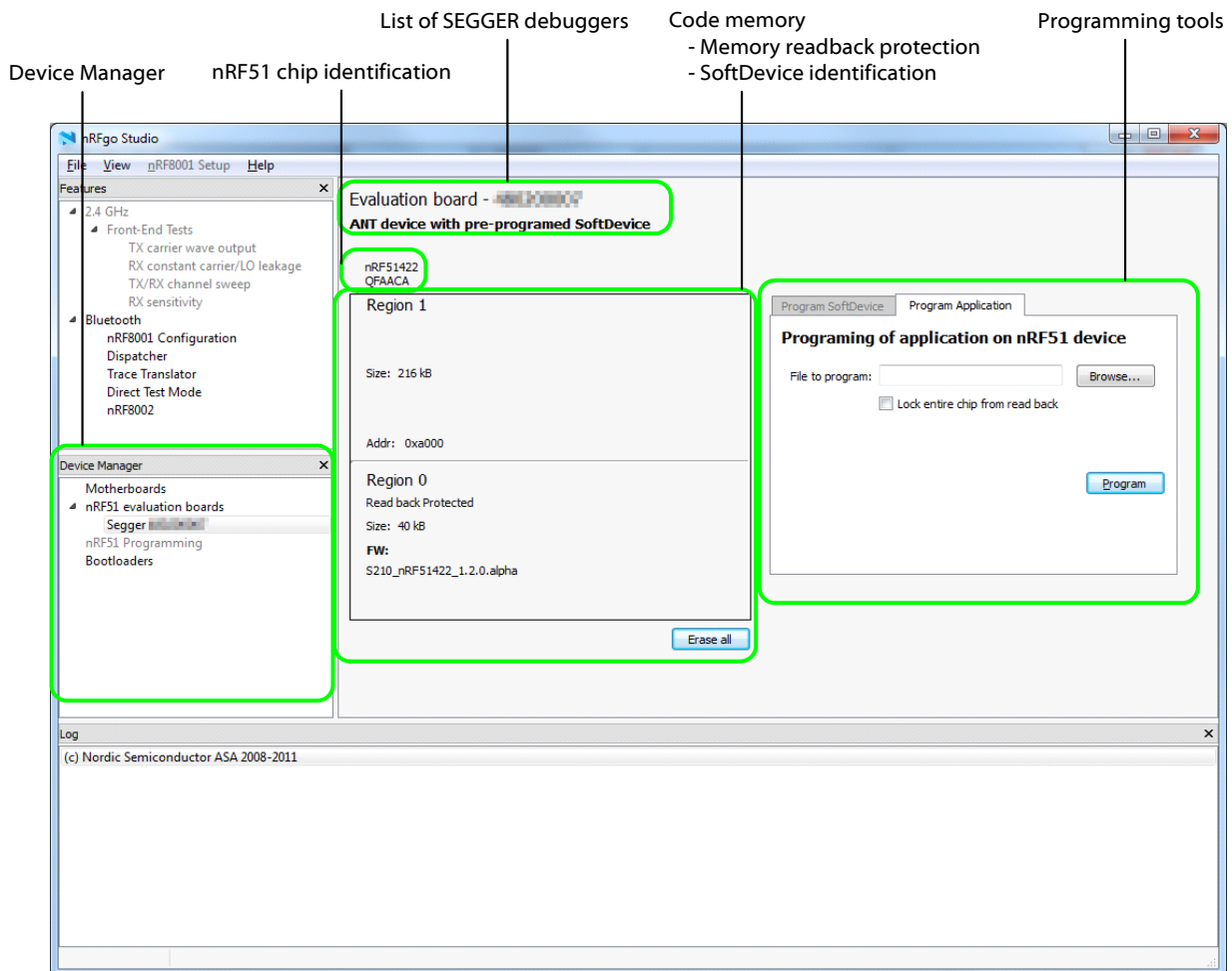


Figure 18 nRFgo Studio dashboard

6.1.1 Selecting a board to program

1. Open nRFgo Studio.
2. In the Device Manager pane select which board to program or erase.
3. The nRF51422 Evaluation Board (PCA10003) will show up under **nRF51 evaluation boards**. The SEGGER serial number identifies the device.
4. Select the board directly by clicking on the SEGGER module listed. The selected board is identified with board type, SEGGER serial number, and the nRF51 chip.

6.1.2 Identifying the nRF51422 chip and chip content

When you select a board, nRFgo Studio identifies the nRF51422 chip and how its memory is organized. The following chip and memory information is displayed:

- **nRF51 chip identification** - Identifies the chip by name and build code (for example, nRF51422 QFAACA). If the debugger is not connected to the chip, or the debugger has a problem communicating with the chip, it will show the following message "No device detected. Ensure that you have the SEGGER connected correctly to the board and that the board is powered and configured for debugging."
- **Code memory** - Shows how the code memory is divided, whether into one or two regions and the size of each region. nRF51422 will always show two regions. For devices containing a SoftDevice, the code memory is divided in two regions, with the SoftDevice in Region 0. The tool shows you how much memory is used by the SoftDevice and how much is left for the application.
 - Memory readback protection - Shows how the readback protection is set. The two possible options are readback protection on Region 0 or readback protection of the whole code memory.
 - SoftDevice identification - nRFgo Studio tries to identify the firmware located in the chip at Region 0. Recognized firmware is displayed as its ID; unrecognized firmware is displayed as its FWID number.

6.1.3 Erase all

You should use the Erase all function in the following situation:

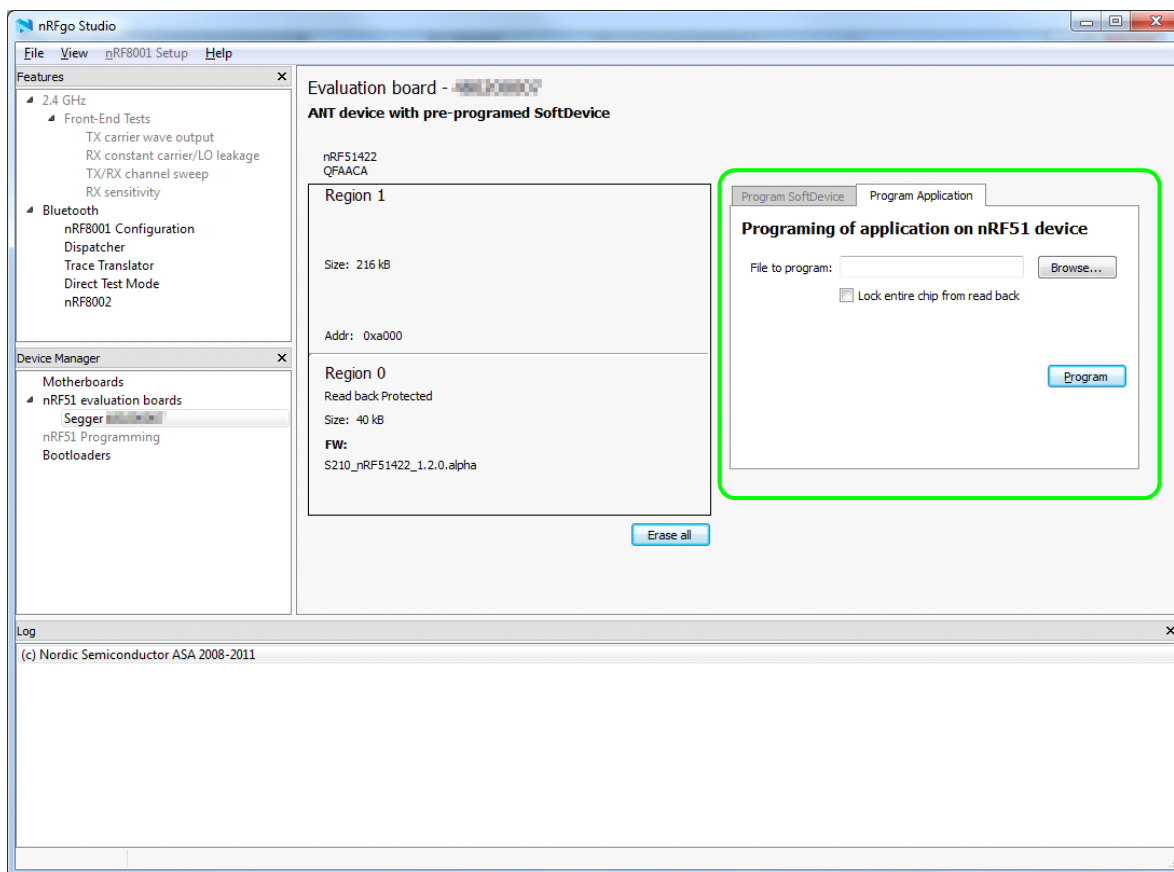
- You have programmed an application on top of the S210 nRF51422 SoftDevice and selected **Lock entire chip from readback**. Once you have performed Erase All, the application will be erased from the chip. The S210 nRF51422 SoftDevice will not be erased.

6.1.4 Programming an application

This function lets you program an application onto the chip (on top of the S210 nRF51422 SoftDevice).

Before nRFgo Studio starts programming it verifies that the HEX file matches the actual memory configuration. If it matches, nRFgo Studio continues with the programming, if not it stops the programming and returns an error message.

Note: This programming will not set up any memory Regions.



1. Follow the steps in **section 6.1.1 on page 23** and then select the **Program Application** tab.
2. Click **Browse** and select the HEX file to program.
3. Select whether to enable or disable readback protection of the entire chip. If you enable readback protection, you will have to do an Erase All to reprogram the chip again.

A chip that is programmed with **Lock entire chip from read back** enabled will not work with a development toolchain. To make it work you must perform Erase all.

Lock entire chip from read back can be used to prevent an accidental overwrite of the chip content.

6.2 Application development

The user application is compiled, linked, and downloaded independently from the SoftDevice. This means that developing and debugging on a chip pre-programmed with a SoftDevice is similar to that of a blank chip. The main differences are memory layout and the call stack size.

6.2.1 Configuring memory layout

Specific SoftDevice versions and stacks can have different requirements. Please review these before proceeding.

The applications vector table must be set up differently depending on whether it will run on a chip that is blank or pre-programmed with a SoftDevice.

The SoftDevice program area starts at address 0x0 and has a predefined size. The application start vector must be placed right after the SoftDevice. The available size has to be set so that it uses the remaining memory for the application. Similarly, the SoftDevice data area starts at the lowest RAM address. The application data area must be placed after the SoftDevice data area.

Table 3 shows examples for setting up the start address and size depending on the code and data size used by the SoftDevice. The example is based on a chip with 256 kB of code memory and 16 kB of RAM.

Device configuration	SoftDevice		App. code start address	Available code memory	App. data start address	Available RAM
	Code memory usage	RAM usage				
Blank chip	0 kB	0 kB	0x0	0x40000	0x20000000	0x4000
SoftDevice A	64 kB	2 kB	0x10000	0x30000	0x20000800	0x3800
SoftDevice B	128 kB	8 kB	0x20000	0x20000	0x20002000	0x2000

Table 3 SoftDevice memory layout

Note: See the *nRF51422 Product Specification* for details on the total code memory and RAM available in the device. The amount of code memory and RAM used by the SoftDevice is described in the *S210 nRF51422 SoftDevice Specification*.

There are two ways to configure the memory layout:

- Using the Keil IDE
- Using a Scatter file

Note: The example code given by Nordic Semiconductor configures the memory layout in the Keil IDE. Scatter file loading is not available when using the evaluation version of the Keil IDE.

6.2.1.1 Memory layout configuration in Keil IDE

To access the Keil IDE memory layout:

1. Click the **Project** menu and select **Options for Target**.
2. Select the **Linker** tab.
3. Check **Use memory layout from Target Dialog**.

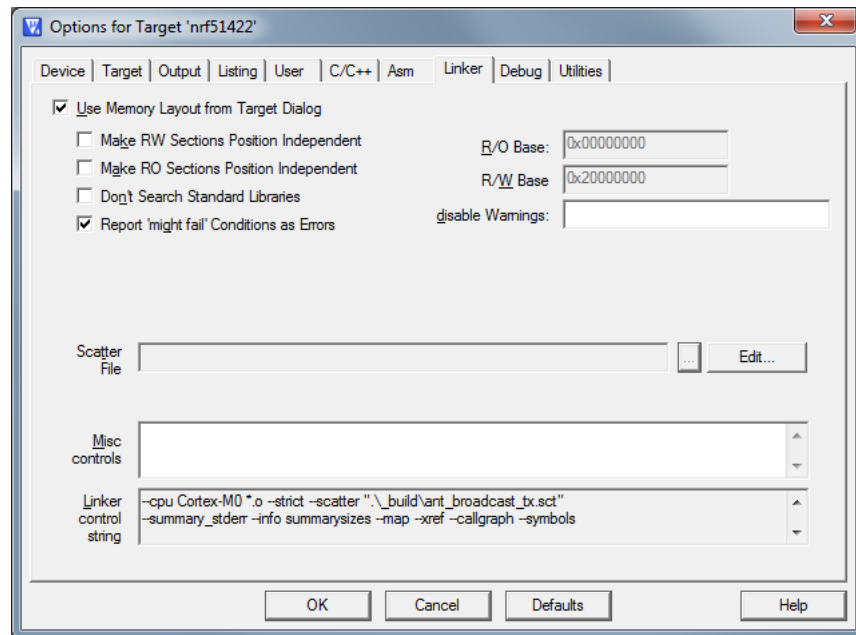


Figure 19 Keil linker settings

4. Select the **Target** tab.
5. In **Read/Only Memory Areas**, define values for **Start** and **Size**.
6. In **Read/Write Memory Areas**, define values for **Start** and **Size** as seen in *Figure 20 on page 27*.
7. Click **OK**.

Below is an example configuration for an application using a chip with 256 kB of code memory and 16 kB of RAM, and a SoftDevice using 128 kB of code memory and 8 kB of RAM (SoftDevice B described in *Table 3 on page 25*).

- Base code memory address 0x20000 and available code memory size is 0x20000 (128 kB).
- Base RAM memory address 0x20002000 and available RAM size is 0x2000 (8 kB).

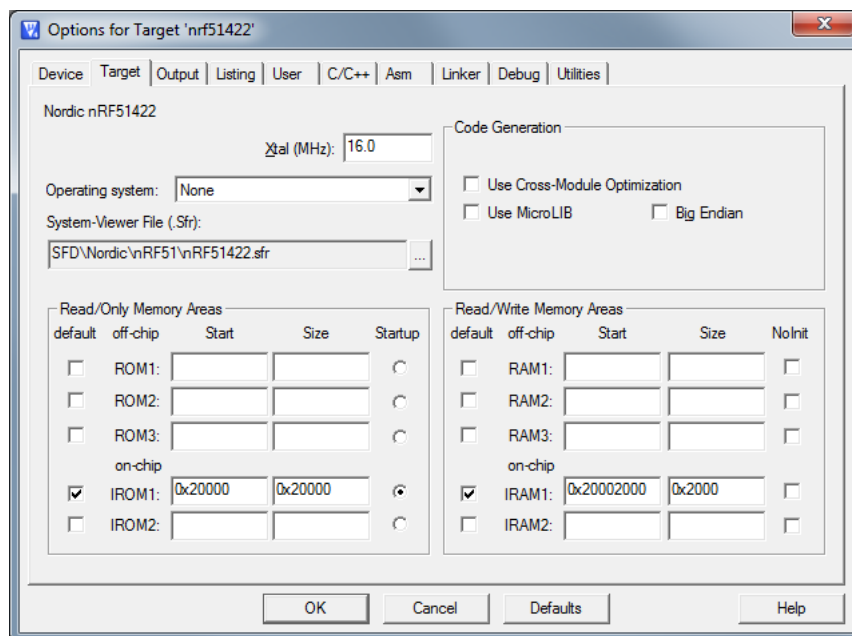


Figure 20 Memory layout with example SoftDevice

Memory		Description
IROM1	Start	Specify the start address for the application code
	Size	Specify available code memory size for the application code
IRAM1	Start	Specify start address for the application data
	Size	Specify available RAM size for the application data

Table 4 Memory layout

6.2.2 Shared call stack

The user application shares the call stack with the SoftDevice if the SoftDevice is loaded on the chip. The application must reserve enough memory for both itself and the SoftDevice in the call stack. The call stack size required by the SoftDevice varies between devices and protocol stack versions, and is supplied in the *S210 SoftDevice Specification*.

The user application sets its call stack size plus the amount needed by the SoftDevice. It then writes the stack pointer at the first address of the application Reset Vector.

Note: Using Keil with the ARMCC toolchain, the call-stack size can be set using the `Stack_Size` definitions in your projects startup file, typically `arm_startup_nrf51.s`.

```
Stack_Size      EQU      0x400 ; The application call-stack size + protocol
call-stack size
Stack_Mem       AREA     STACK, NOINIT, READWRITE, ALIGN=3
__initial_sp    SPACE    Stack_Size
```


6.2.3 Debugger configuration

Project files delivered in the SDK are configured and ready for download and debugging. If a new application project is used, the debugger must be properly configured. To configure the debugger:

1. In Keil, select **Options for Target (ALT+F7)** from the **Project** menu. The **Options for Target** dialog box appears.
2. Select the **Debug** tab.
3. Apply the **Use** option and select the J-Link/J-Trace debugger from the list.
4. Set **Driver DLL** to SARMCM3.DLL.
5. Set **Dialog DLL** to TARMCM1.DLL.

Other options can be selected as needed. To take full advantage of the debugger and its features, the following are advised:

- Breakpoints
- Load Application at Startup
- Memory Display
- Toolbox
- Watch Windows

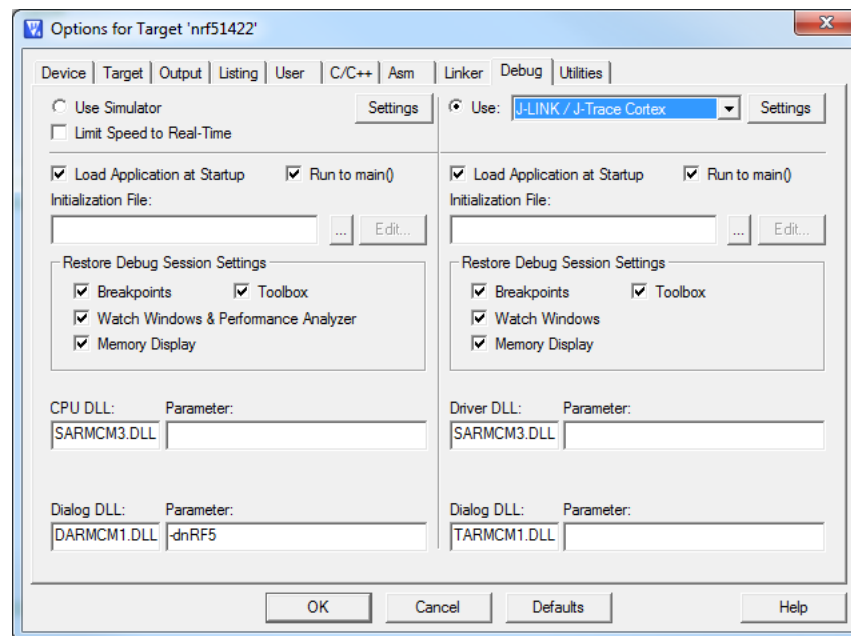


Figure 21 Debugger options

6. Click **Settings** next to the **Use** field in the top right of the window.
7. In the Target driver setup, provide information about debugging protocol and maximum speed. Select **SW** in the **Port** drop-down.
8. In **Max Clock** the maximum speed for the debugging port cannot be exceeded (1 MHz). A proper configuration is shown in **Figure 22 on page 30**.

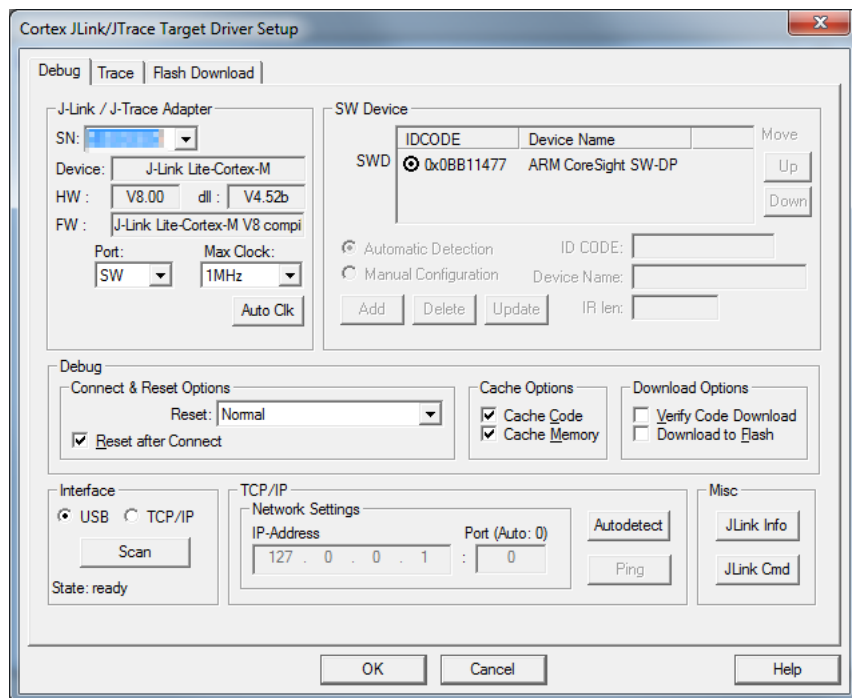


Figure 22 Target driver setup

9. Click **Start/Stop Debug Session (CTRL+F5)** in the Keil IDE to start debugging.

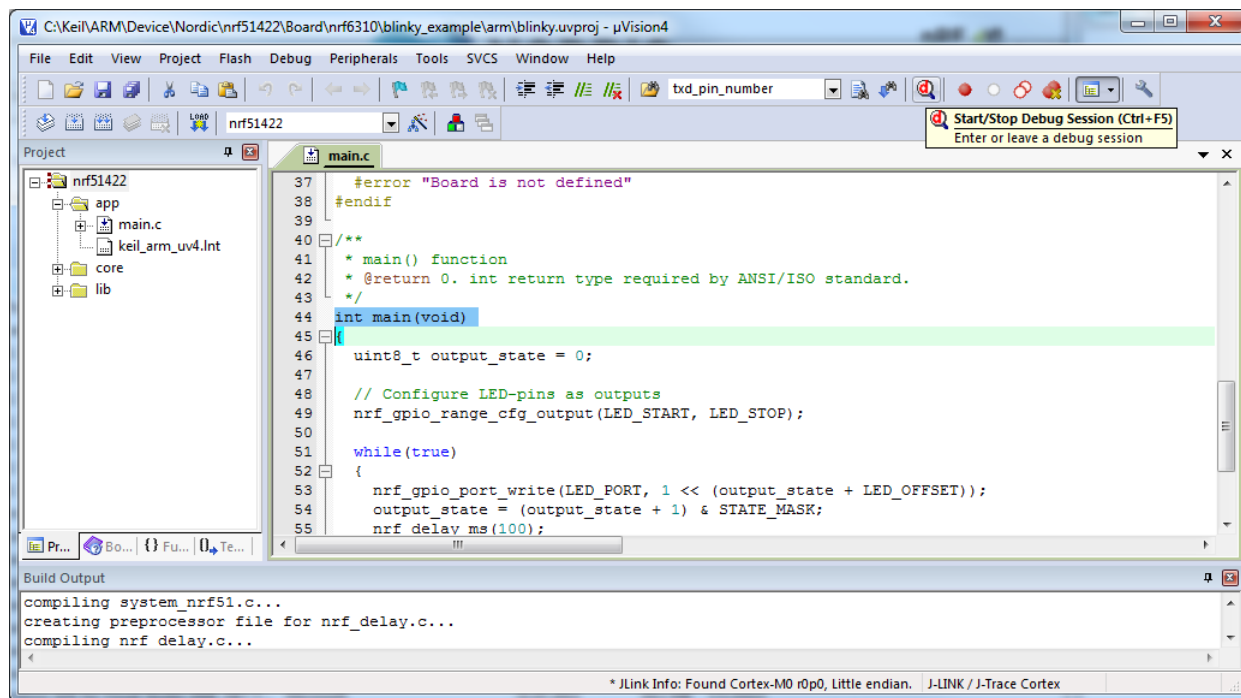


Figure 23 Debugger initiation

6.2.4 Limitation when debugging on a chip with a SoftDevice

When a SoftDevice is installed in a device, there are certain limitations when debugging.

The nRF51422 has a Memory Protection Unit that prevents the debugger from having read/write access to the SoftDevice flash area. The debugger reads this area as 0x0000 (no operation instruction).

6.2.5 Programming the device

The code memory area occupied by the SoftDevice is read, write, and erase protected. The debugger will read this area as 0x0000 (no operation instruction). When the SoftDevice is enabled, the Memory Protection Unit enables write protection for certain peripherals used by the protocol stack. Protected peripherals are described in the *S210 nRF51422 SoftDevice Specification*.

Configure the debugger and compile and link the application code. Download the application using the Keil IDE download button. To configure and start the download:

1. Select **Options for Target** in the **Project** menu.
2. Select the **Utilities** tab in the **Options for Target** dialog box.
3. Click **Settings**.
4. Select the **Program** check box.
5. Choose **Erase Full Chip**.
6. Click **Add** and select the nRF51xxx algorithm from the list to select the programming algorithm used by Keil IDE.
7. Click **Download**.

Note: The nRF51xxx algorithm is installed automatically during the SDK installation. This algorithm is a generic nRF51 series algorithm, which provides download capabilities to all series devices up to 2 MB of code memory.

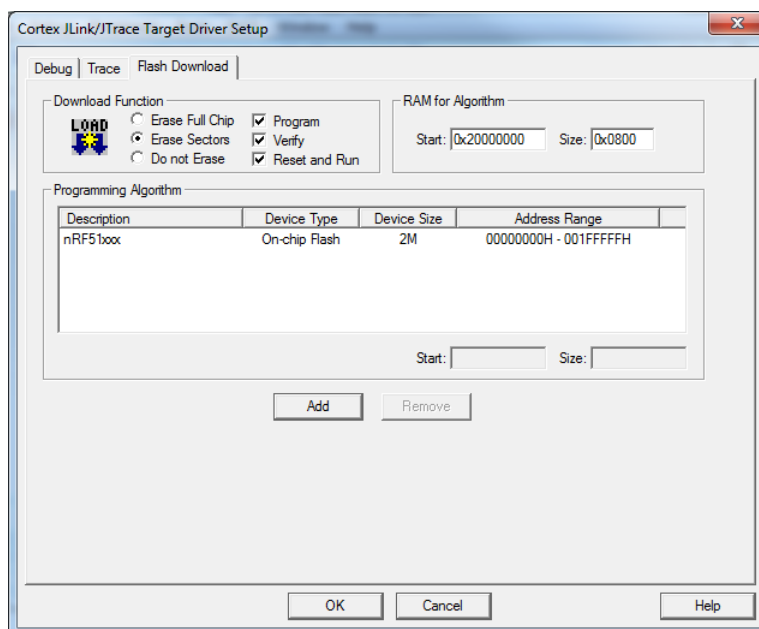


Figure 24 Selecting J-LINK/J-Trace Cortex

8. In the **Utilities** tab, select **Use Target Driver for Flash Programming**.
9. Choose the available debugger from the list as shown in *Figure 25*.

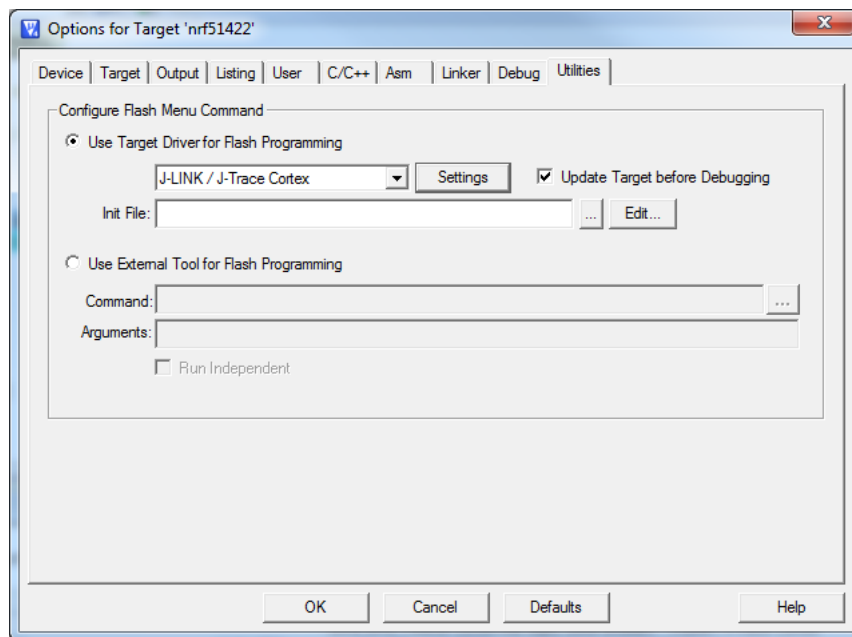


Figure 25 Debugger selection

7 Debugging the nRF51422 chip

For debugging with SEGGER J-Link, see *Appendix A: on page 40*. For general information on how to debug using the Keil μ Vision IDE, see <http://www.keil.com/uvision/debug.asp>.

The following steps tell how to configure the debugger in Keil.

1. In the **Project** menu click **Options for Target (CTRL+F7)**.
2. Select the **Debug** tab.
3. To enter debugging mode click **Start/Stop Debug Session** or **CTRL+F5**.

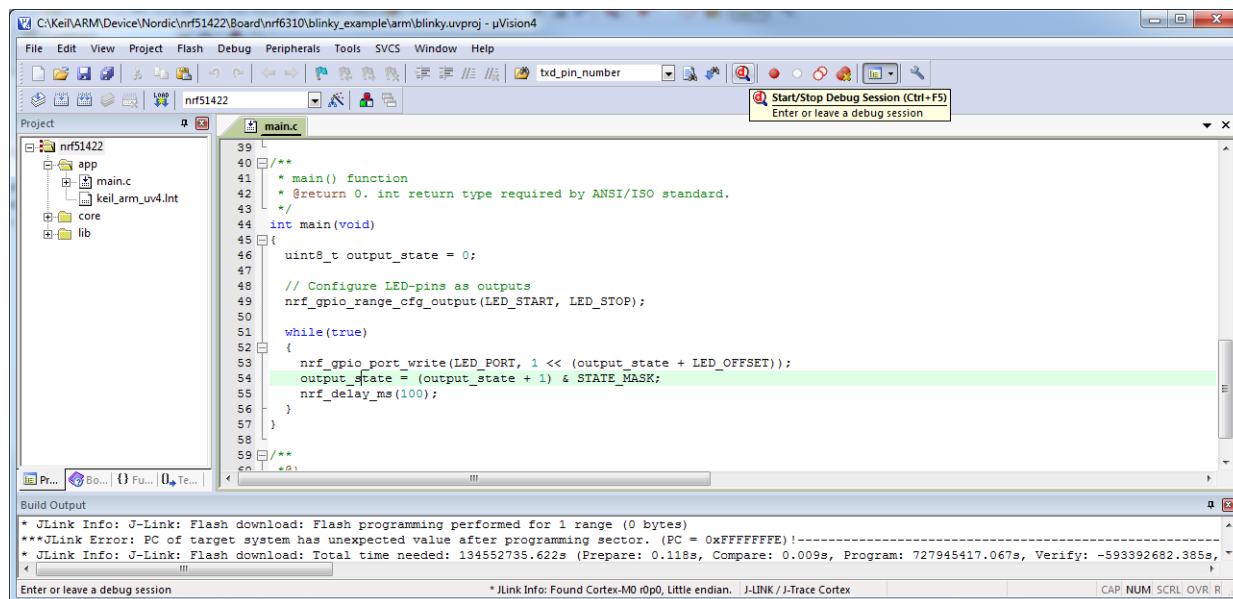


Figure 26 Start debugging mode

7.1 nRF51422 debug features and precautions

This section contains information about the System Viewer Windows, debugging an application when a readback protected SoftDevice is present, and setting a breakpoint using a SEGGER J-Link debugger.

7.1.1 System Viewer windows

The System Viewer enables you to select device peripherals and see their contents in separate windows, see *Figure 27*. In the **View** menu point to **System Viewer** and select the peripheral you want to see. The peripheral register values are displayed in their respective pane in Keil, as seen in *Figure 28 on page 35*. More information on System Viewer can be found at: http://www.keil.com/uvision/db_view_sysview.asp.

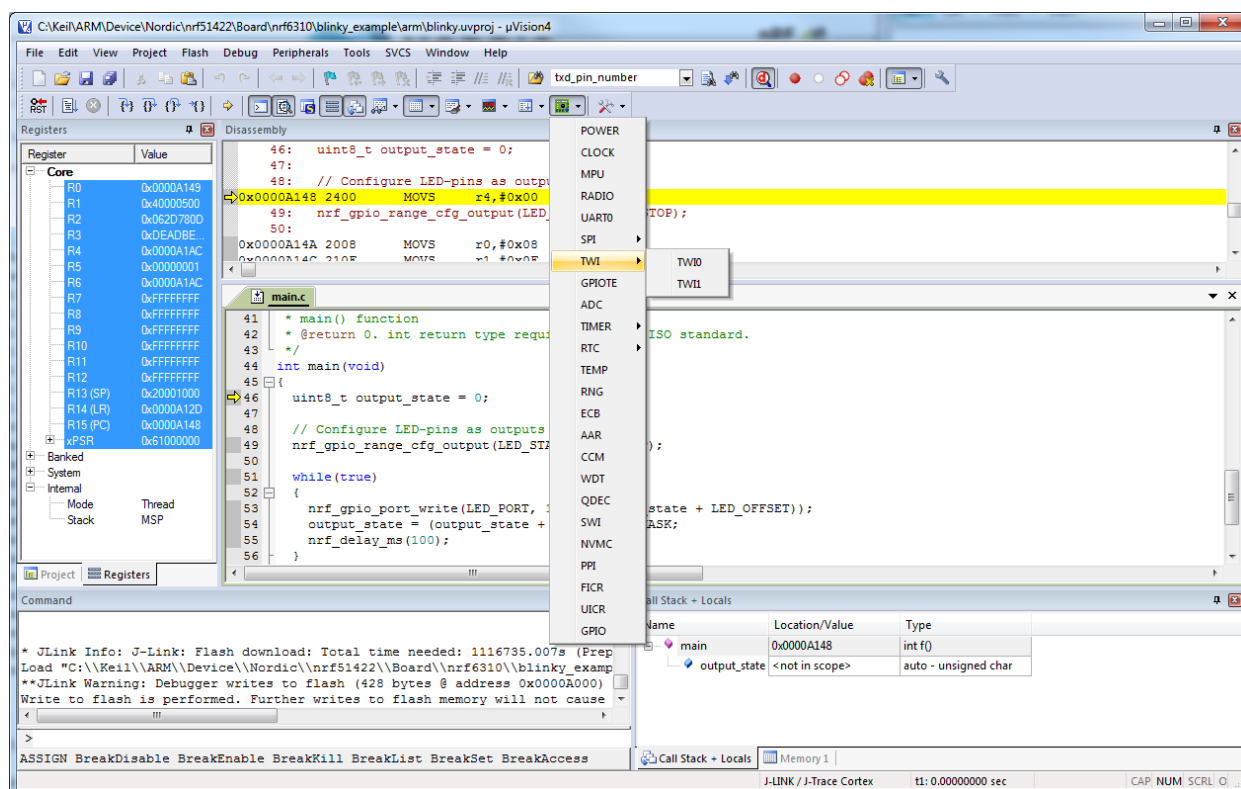


Figure 27 System Viewer Windows

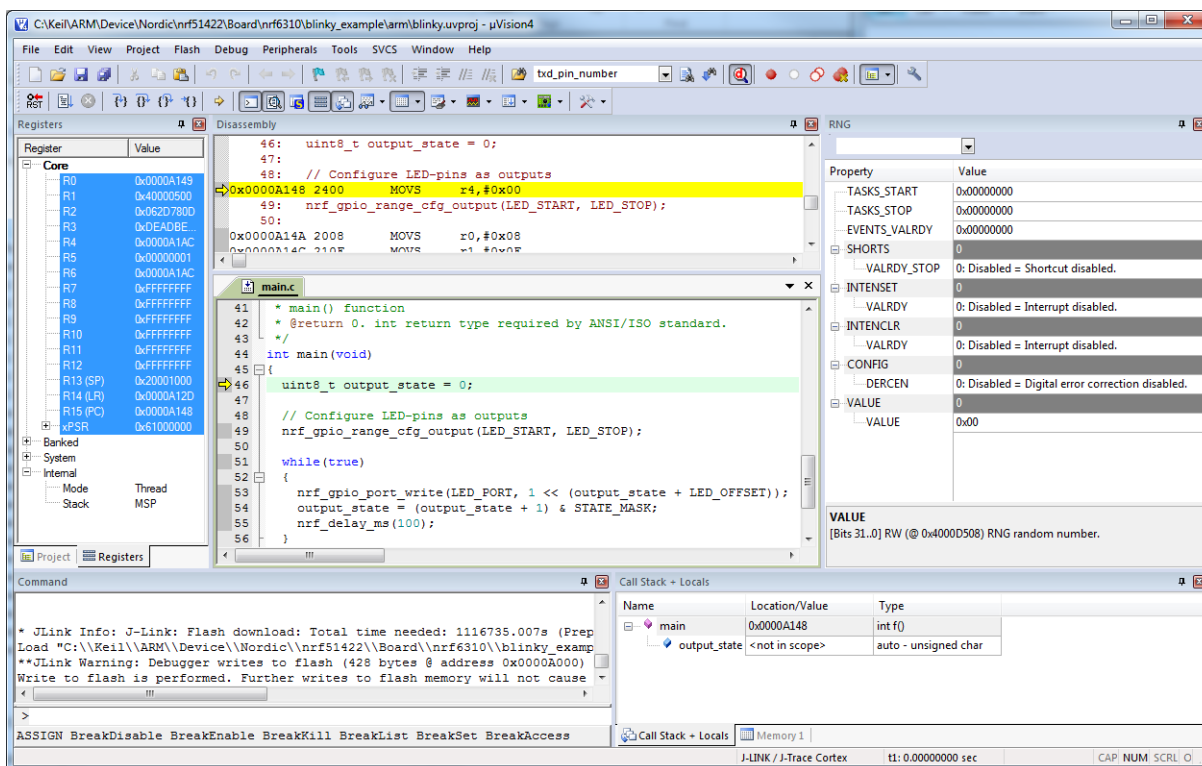


Figure 28 System viewer window of the RNG peripheral

7.1.2 Debugging an application when a readback protected SoftDevice is present

Debugging applications with a SoftDevice present behaves as described in <http://www.keil.com/uvision/debug.asp>, except when the program counter is in Region 0 on a SoftDevice with readback protection enabled. Code words from addresses in the protected area will always return zero to the debugger.

Any values in peripheral registers that are restricted or blocked by the SoftDevice will be invisible to the debugger as well. Information on the SoftDevice configuration and memory resource mapping can be found in the *S210 nRF51422 SoftDevice Specification*.

Note: Avoid single stepping to the protected area. Instead, set the breakpoint right after SVC calls while debugging and run the application to the actual breakpoint, see *Figure 30 on page 37*. The “step over” function (**F10**) may also be used instead to step over SVC calls to avoid delays when entering the readback protected area.

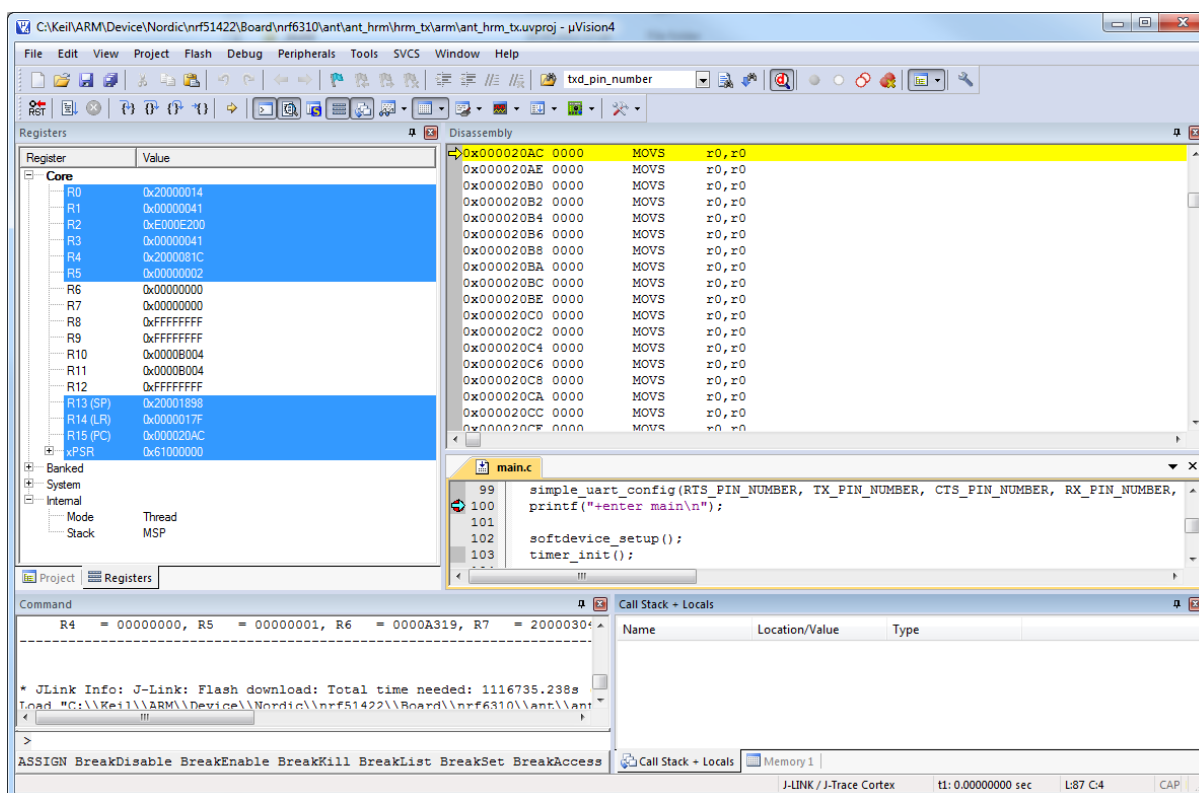


Figure 29 Debugger information for a setup with a SoftDevice enabled in the protected area

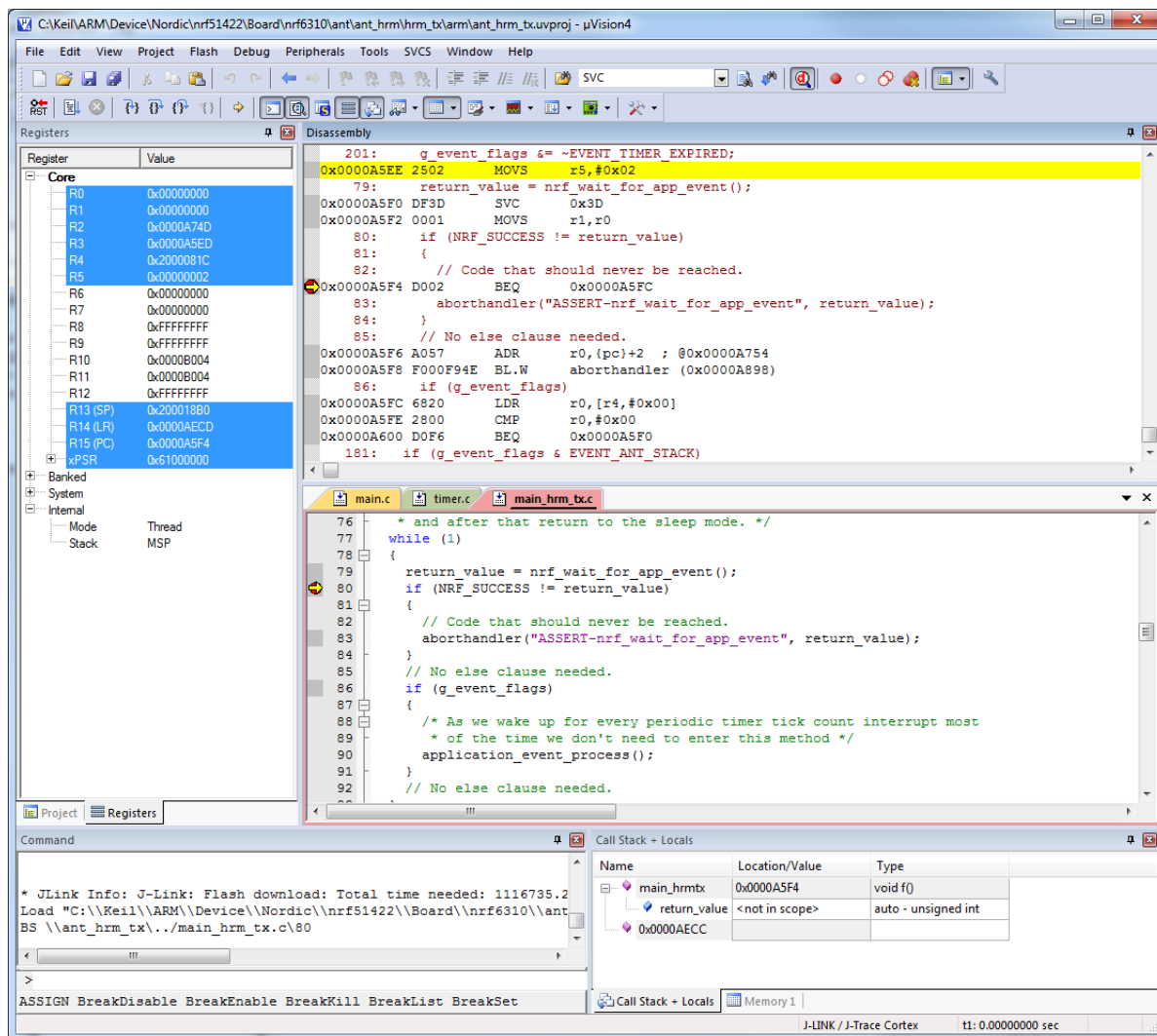


Figure 30 Setup with a breakpoint after an SVC call

8 Software Development Kit

The nRF51 Software Development Kit (SDK) enables you to develop applications for the following protocol stacks:

- ANT
- ANT+

Note: ANT+ keys are needed to use ANT+ profiles included in the SDK. Please register as an ANT+ adopter at <http://thisisant.com> to get the keys.

8.1 Installing the nRF51 SDK

The nRF51 SDK is a part of the downloadable content available from your My Page account, see *section 2.2 on page 5*. The SDK is downloaded as a MSI file (a Windows Installer) and is installed by running the application. When installing the SDK you can select: Keil MDK Support and/or Custom install. The following describes each installation option:

- **Keil MDK support** - installs Keil μ Vision example project files, the code memory programming algorithm for the J-Link debugger, and the Nordic nRF51 series device database file for Keil.
- **Custom install** - installs a software archive to a customized location.
- **nrfjprog** - installs a command line programming interface to be used with SEGGER debuggers.

Note: The Keil MDK Support option will only be available if you already have the Keil MDK toolchain installed.

9 Troubleshooting

The nRF51422 device on hardware v1.0 of the PCA10003 does not respond when I try to contact it. What has happened?

Verify that both jumpers on connector **P8** on the PCA10003 are in place.

When I connect multiple SEGGER J-Link debugger boards to my computer, µVision does not recognize them correctly.

This is a known limitation with µVision in MDK v4.53 or earlier that is fixed in later versions. Upgrade to version 4.54 or later.

On my 32 bit Windows XP machine, I get an error message with code 2908 when reinstalling either the nRF514 SDK or nRF518 SDK.

Installing or reinstalling either the nRF518 SDK or the nRF514 SDK *after* the nRF518 SDK has been uninstalled will cause error message code 2908 during installation. Use the Task manager (**Ctrl+Shift+Esc**) to end the task **nRF514/8 SDK Setup**. Drivers included in nRF518 will still be installed (if they are not already installed).

The debugger seems to freeze while debugging.

If running a SoftDevice that has been programmed with the “Lock SoftDevice from Readback” enabled (see section **section 6.1 on page 22**), the debugger will halt while stepping to an SVC instruction. You should set the breakpoint after the SVC instruction and run the application to the breakpoint, or step over any SVC instructions. See **section 7.1.2 on page 36** for more details.

The debugger does not halt on breakpoints.

Some Keil projects in the SDK have **Optimization level 3 (-O3)** and **Optimize for time** checked. If you are debugging an application with these settings, your breakpoint set might have no effect.

1. Press **Alt+F7** to open the Target options dialog.
2. Select **C/C++**.
3. Select **Optimization level 0** from the scroll down list.
4. Uncheck **Optimize for time**.

Software gets out of sync while debugging.

Setting/modifying breakpoints on a running system using the SEGGER debugger halts the CPU, which may result in software that is out of sync. You should avoid setting breakpoints while the system is running.

The debugger is not able to detect my nRF51 device after I have downloaded my firmware.

If the nRF51 device goes to SystemOff too soon after reset, it will have a problem communicating with the J-Link debugger. You can recover using the **Recover** button in nRFgo Studio.

1. Cycle the power to the nRF51 chip before you start the Recover application.

Note: The Recover function will erase all application firmware on the chip.

Appendix A: Installing drivers and configuring KEIL projects for the SEGGER debugger

This appendix describes the steps for installing the software and using the SEGGER J-Link Lite debugger with Keil µVision for nRF51 series devices, based on J-Link software version 4.52b or later.

Prerequisite

You need Keil µVision with ARM-MDK that you have tested to be working with MDK version 4.54.

Note: All projects in the nRF514 SDK are preset to work with the SEGGER debugger. Only the following step **Download and install SEGGER drivers** is needed.

Download and install SEGGER drivers

1. Download the latest SEGGER J-Link software and documentation pack from <http://www.segger.com/jlink-software.html>.
2. Download and run the J-Link Software and documentation pack for Windows from <http://www.segger.com/jlink-software.html>. The serial number from your SEGGER J-Link hardware is needed, see **Figure 32 on page 41**.
3. During installation you will be prompted to select the IDE that you want updated with the latest SEGGER DLLs. Check the box for **Keil MDK** and any other IDEs you want to use with SEGGER.

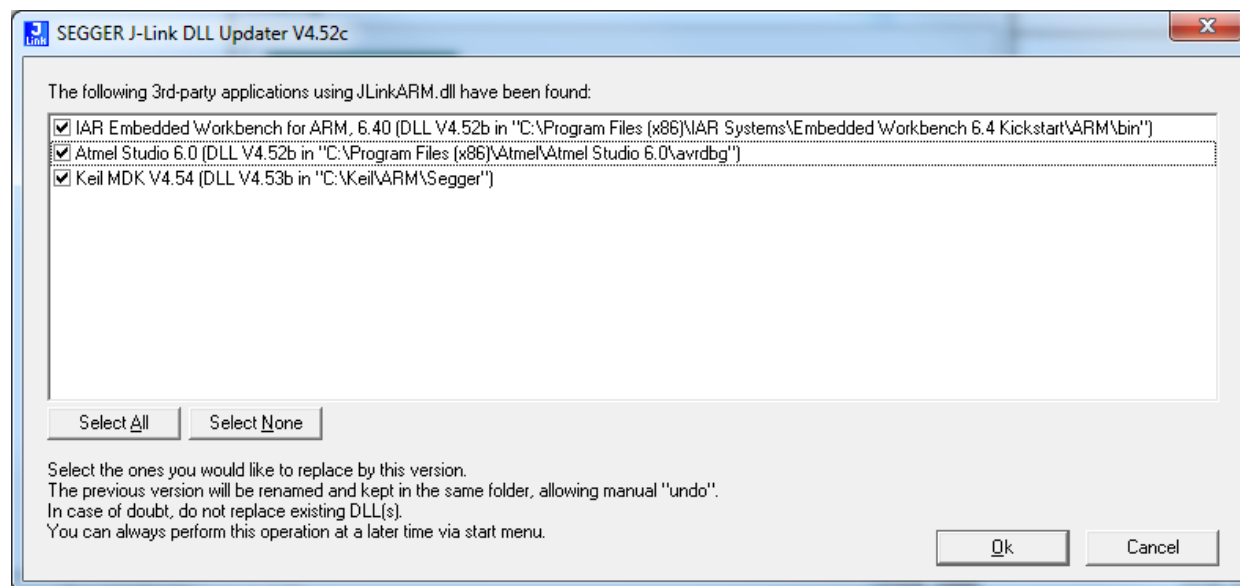


Figure 31 IDEs selected for updating to the latest SEGGER DLLs

4. Go to http://www.segger.com/IDE_Integration_Keil.html#knownproblems for MDK v4.54. Download JL2CM3 and copy it to <keil>/ARM/Segger. This patch is necessary for the SEGGER debugger to work.
5. Plug the Evaluation Board (PCA10003) into your computer with a USB cable. The LD3 LED will blink while the driver installation occurs. Wait until the LED is continuously lit

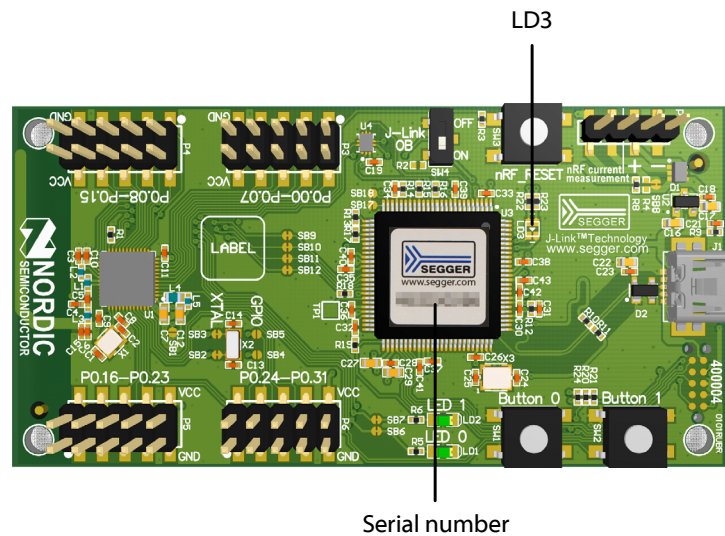


Figure 32 J-Link Lite CortexM-9 serial number location

Configuring KEIL projects for the SEGGER debugger for first time use

Create the **JLinkSettings.ini** file with the contents shown in *Figure 38 on page 45*. The file **JLinkSettings.ini** should be saved in the same folder as the Keil μ Vision project (uvproj) file.

1. Double-click an example project file to open the Keil μ Vision IDE.
2. Click **Target Options** on the toolbar or click **Project** menu and select **Options for Target**

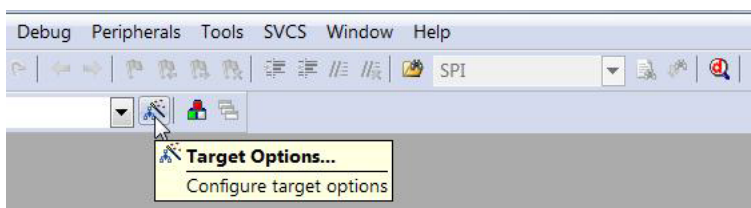


Figure 33 Keil Target configuration

3. Under the **Debug** tab in the Use list, select **J-LINK / J-Trace Cortex** option as shown in *Figure 34*.
4. Click **Settings** as shown in *Figure 34*. Both the SEGGER Control Panel and the Keil Target Driver Setup will open.

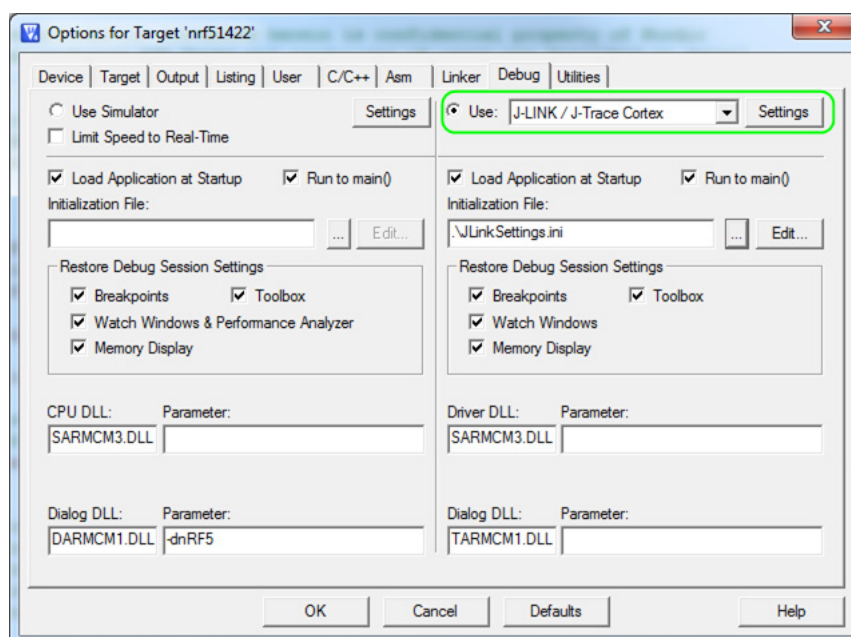


Figure 34 Selecting JLink debugger in Keil

Note: If the SEGGER J-Link Lite firmware requires an update you will be prompted with the message "A new firmware version is available for the connected emulator". In this case, click **OK**.

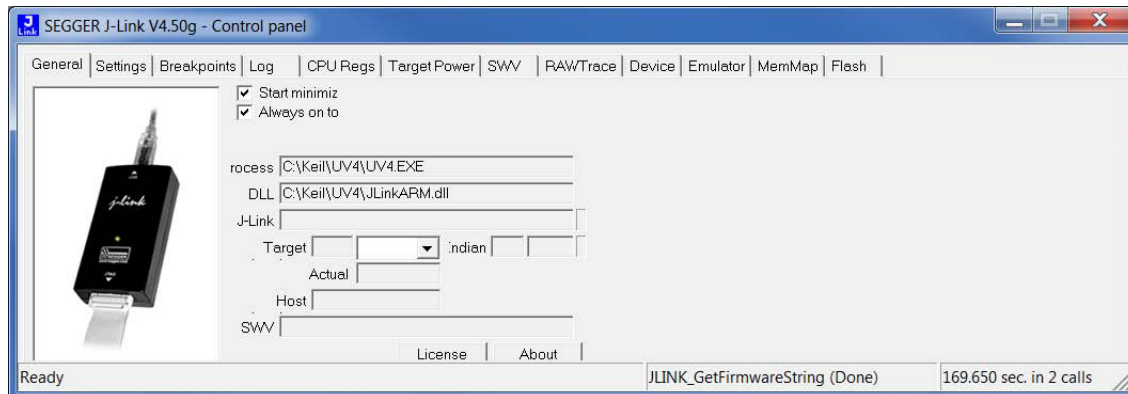


Figure 35 Segger control panel

- Click the **Debug** tab shown in **Figure 36**. Set Port to **SW** and Max Clock to **1 MHz**, as shown in **Figure 36**. Make sure that **SN** and **IDCODE** are populated as seen in **Figure 36** and click **OK**.

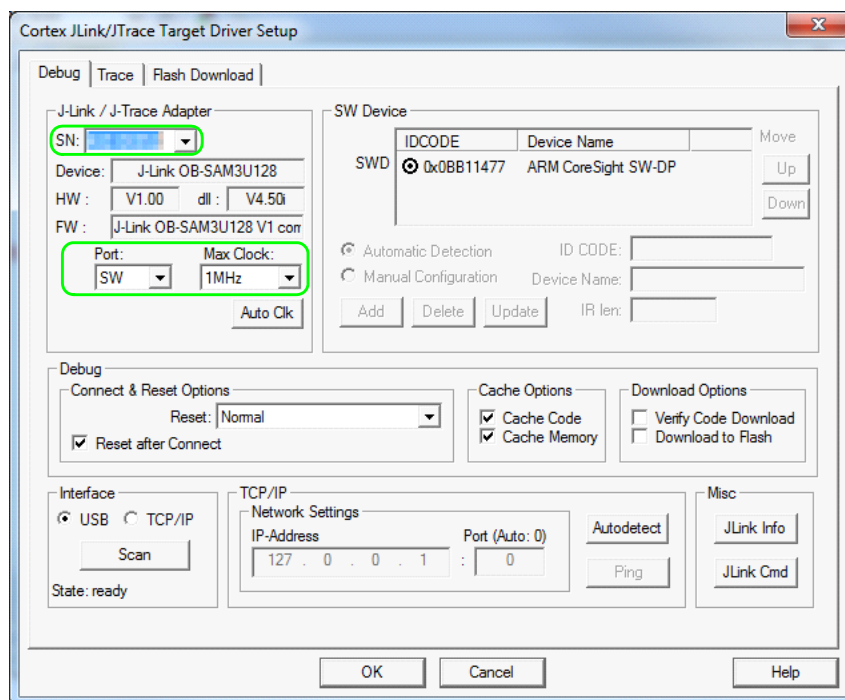


Figure 36 Debug settings

- Select the J-Link device for target programming and provide the appropriate code memory algorithm.

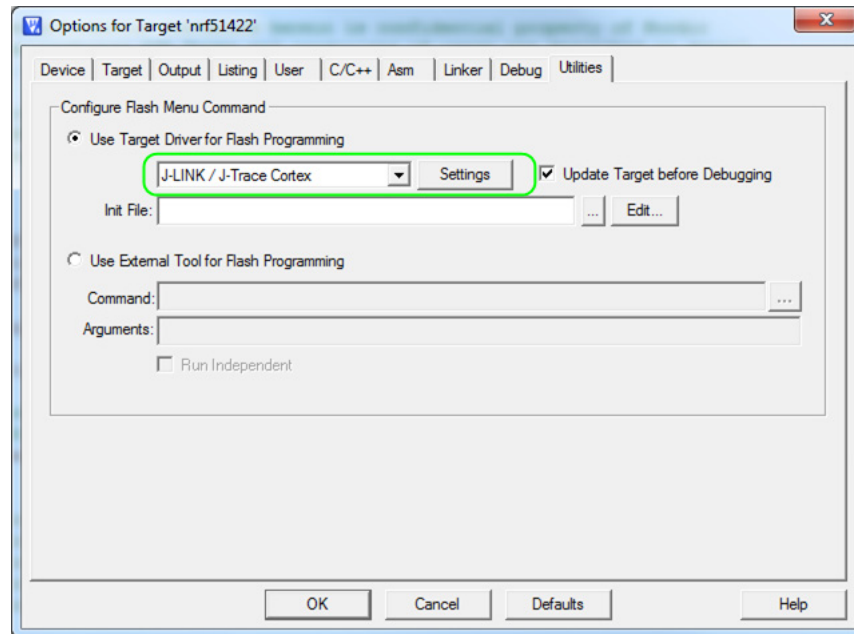


Figure 37 Flash settings

7. If the J-Link serial number appears in the SN field, the device is properly installed. The default settings can be accepted by clicking **OK**, closing both the SEGGER Control Panel and Keil target Driver Setup.

JLinkSettings.ini file

While debugging with a SoftDevice, change AllowSimulation = 1 to AllowSimulation = 0 in your default **JLinkSettings.ini** file under your project.

1. Enter the utilities settings from KEIL target options (**ALT+F7**) and click the ... button to select the **JLinkSettings.ini** file.
2. Click **Edit** and change the AllowSimulation as specified above.

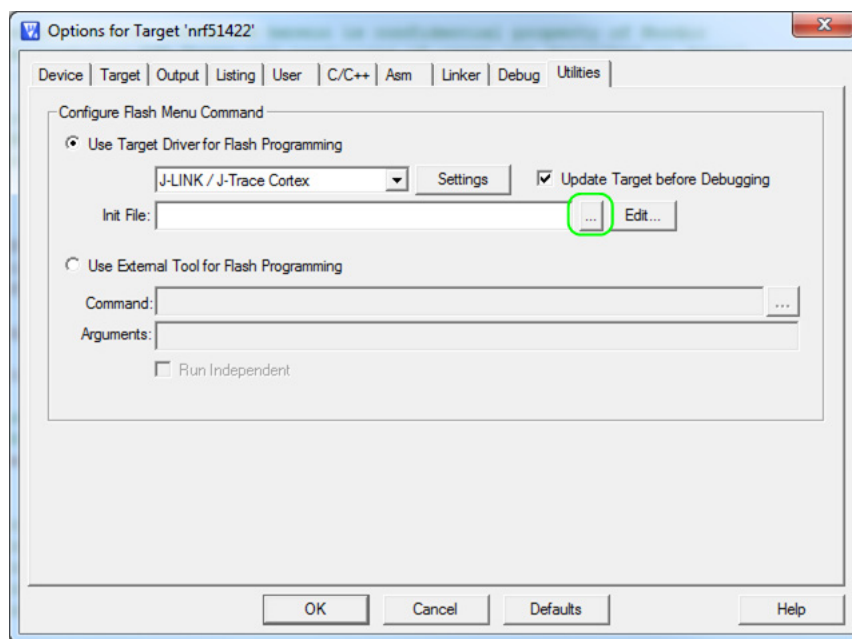


Figure 38 Locating JLinkSettings.ini

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



Revision history

Date	Version	Description
February 2013	1.1	Updated content to match v2.0 of the hardware. Updated hardware information in <i>section 5 on page 12</i> .
November 2012	1.0	First release

ARM statement

Keil, μ Vision, and Cortex are trademarks of ARM Limited. All other brands or product names are the property of their respective holders.