nRF51822 Evaluation Kit

nRF51822

User Guide v1.0

Copyright © 2012 Nordic Semiconductor ASA. All rights reserved.
Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

Contents

1 Introduction

The nRF51822 *Bluetooth*® low energy/2.4 GHz Proprietary Evaluation Kit provides a complete solution for testing and evaluating the nRF51822 device. The nRF51822 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

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.***

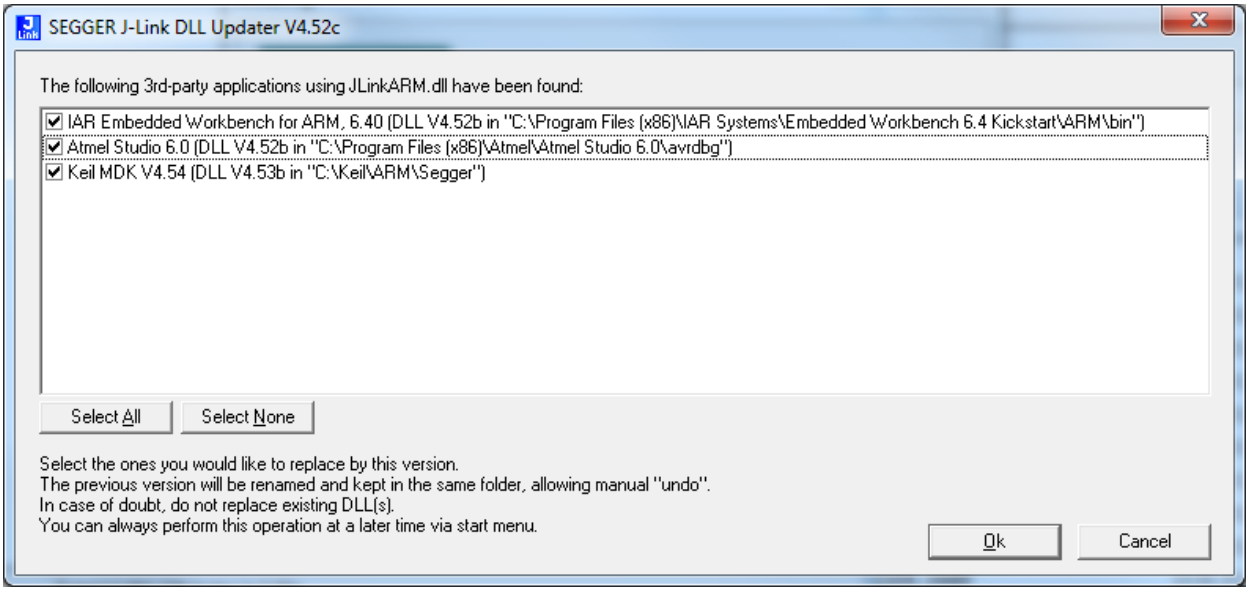
1.4 Evaluation kit release notes

Date	Version	Description
September 2012	1.0	**Known issues** • PCA10000 v1.0 and PCA10001 v1.0: • 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 Quick start

Register, download, and install

1. Download and install Keil MDK-ARM Lite from https://www.keil.com/demo/eval/arm.htm to your hard drive. If you have Keil MDK-ARM Lite version 4.54 or later already installed, go to step 2.
2. Download and run the J-Link Software (version 4.52b or later) 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.

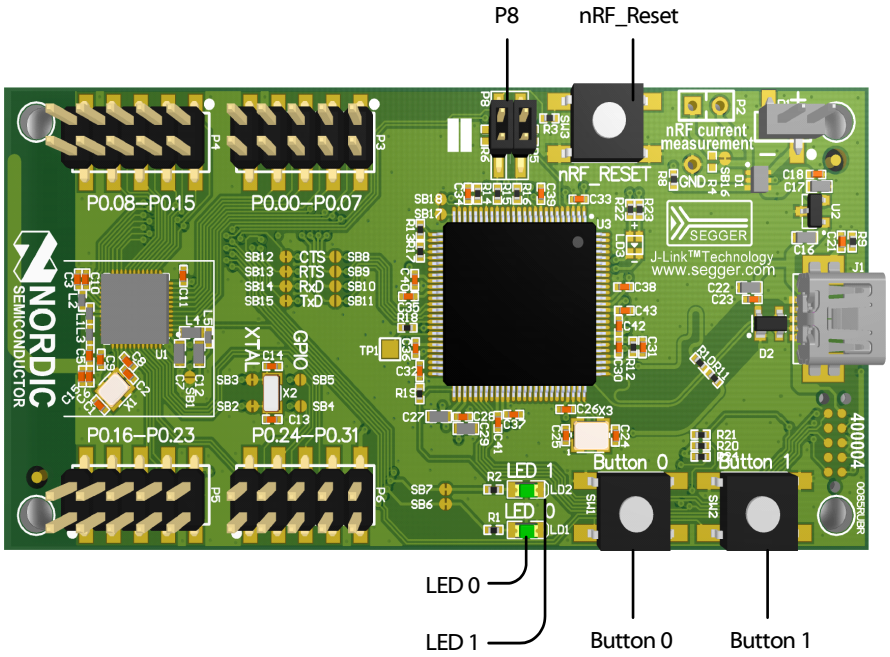


4. Go to http://www.segger.com/IDE_Integration_Keil.html#knownproblems for MDK v4.54. Download JL2CM3 and copy it to **<keil>/ARM/Segger**. This patch is necessary for the SEGGER debugger to work.
5. Go to www.nordicsemi.com and log in to your Nordic My Page account.
6. Select **My Products** from the left menu.
7. Enter the product key (included with this kit) into the Product Key field and click **Add**.
8. Click the **Downloads** link in the Overview, My Products table.
9. Download and run the nRF518 SDK installer. Make sure to choose the **Keil MDK-ARM** installer option.

2.1 Install the nRF51822 Evaluation board (PCA10001)

Connect the hardware

1. Ensure that the two jumpers at **P8** are mounted.
2. Connect a USB cable from the PCA10001 board to your computer.



Start the Blinky project

1. Locate the Blinky project found under <keil path> **\ARM\Device\Nordic\nRF51822\Board\PCA10001\blinky_example\arm**.
2. Open the Blinky project in Keil µVision by double clicking the **blinky.uvproj** file.
3. Select **nRF51822** from the Select Target list and click **Build** or press **F7** to compile the Blinky project.
4. Click **Load** to download and run the Blinky example firmware. **LED 0** and **LED 1** on the PCA10001 should now blink in a sequential order.

2.2 *Bluetooth* low energy heart rate monitor demo

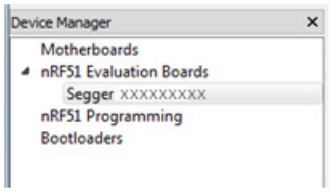
This section tells you how to program the heart rate monitor demo, giving you a simple application to get your device up and running.

Download and program the SoftDevice

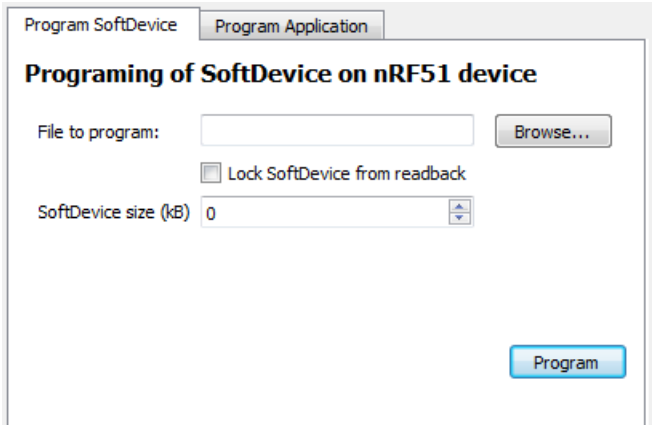
Type the product key (included with the Evaluation Kit) into the Product Key field in My Page to download the S110 nRF51822 SoftDevice.

Follow these steps to program your device:

1. Open nRFgo Studio.
2. In the Device Manager select the nRF51 Evaluation Board (identified by the SEGGER serial number).



3. Select the **Program SoftDevice** tab.



4. Click **Browse** and navigate to the SoftDevice file you downloaded.
5. Click **Program**.

Compile, program, and run the heart rate monitor demo

Note: The development dongle must be unplugged during these steps.

1. Locate the Heart Rate demo project found under <keil path> **\ARM\Device\Nordic\nrf51822\board\PCA10001\ble\ble_app_hrs\arm**.
2. Open the Heart Rate demo project in Keil µVision by double clicking the **ble_app_hrs.uvproj** file.
3. Click the **Build** icon or press **F7** to build the project.
4. Ensure that Evaluation board PCA10001 is the only module connected to your computer (that is, keep the nRF51822 development dongle unplugged). This is to ensure that you are downloading the heart rate application to the correct device.
5. Go to the **Flash** menu and click **Download** to load the program (or click the **Load** icon).
6. Press **Button 0** to start the Heart Rate demo. **LED 0** should be lit indicating it is advertising.

7. The application advertises for 3 minutes. If a connection isn't made within this period, the application sets nRF51822 to System Off.
8. To start advertising again press **Button 0**.

Install the nRF51822 development dongle (PCA10000)

1. Plug the nRF51822 development dongle into a USB port on your computer.
2. An icon will appear in the lower right corner of your monitor showing that the drivers are being installed. Wait until it is ready.

Scan for available *Bluetooth* low energy devices

1. Start the Master Control Panel from the Windows Start menu (**Start > All Programs > Nordic Semiconductor > Master Control Panel**).
2. Make sure the nRF51822 development dongle is detected. The Master Emulator item list should show COMnn-xxxxxxxxx (*nn* gives the COM port number; *xxxxxxxxx* is the SEGGER serial number printed on the dongle). Restart the application if it doesn't appear in the item list. Before continuing, make sure you have selected the correct device by verifying the serial number in the item list with the serial number printed on the nRF51822 development dongle.
3. When you use the nRF51822 development dongle for the first time, you must program it with the Master Emulator Firmware.
    a. In the Master Control Panel menu click **File** and select **Flash Programming**.
    b. Click **Browse**. This opens a browser that automatically points to the location of the **mefw_nrf51822_<version>_firmware.hex** (<version> will be replaced by a number giving the version of the actual firmware).
    c. Select the **Master Emulator Firmware** file and click **Open**.
    d. Click **Program** to start programming the selected device.
    e. When the programming is finished click **Exit**.
4. Ensure that the Heart Rate Demo running on PCA10001 is advertising, indicated by **LED 0** blinking.
5. Click **Start discovery**. The Master Emulator will scan for available *Bluetooth* low energy devices within range and list them.
6. Select the device with the CompleteLocalName 'HRS_EVAL' in the Discovered Devices list.
7. Click **Select device**.
8. Click **Service Discovery**. The Master Emulator will connect to the device and search for services and characteristics. In the **Service discovery** pane you will see the services and characteristics of the device. On PCA10001 **LED0** will turn off.
9. Click **Enable services**. You should see the Heart Rate Measurement characteristic and Battery Level being notified every few seconds with a different value (the Heart Rate Measurement/ Battery Level value line will blink green for each notification).
10. Pressing **BUTTON 0** or **BUTTON 1** will make the Heart Rate Measurement value increase or decrease by two.

3 Kit content

The nRF51822 Evaluation Kit consists of hardware and access to software components, documentation, and design files from www.nordicsemi.com.

3.1 nRF51822 Evaluation Kit hardware content

1 x Lithium 3 V battery

1 x nRF51822 Evaluation Kit board (PCA10001)

1 x nRF51822 development dongle (PCA10000)

Figure 1 *nRF51822 Evaluation Kit hardware content*

3.2 Downloadable content

The nRF51822 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.

3.2.1 nRF51822 software content

- nRFgo Studio
- nRF518 Software Development Kit (SDK)
  - Precompiled HEX files
  - Source code
  - Keil ARM project files
- S110 nRF51822 SoftDevice

3.2.2 nRF51822 documentation

- This User Guide
- nRF51 Series Reference Manual
- nRF51822 PPS
- S110 SoftDevice Specification
- nRF51822 PAN

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

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

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

4 Evaluation kit configuration

This chapter explains how to download third party content, the development environment setup, and how to program nRF51822.

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 45.***

4.1.1 Programming the nRF51822 device

The nRF51822 device can be programmed from several environments. In this section we will show how to program using Keil MDK-ARM. The nRF51822 EK can be configured to develop proprietary 2.4 GHz protocol-based applications and *Bluetooth* 4.0 single-mode applications.

Figure 2 and ***Figure 3 on page 13*** show the relationship between the hardware and software components for 2.4 GHz based and *Bluetooth* 4.0 single mode development.

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

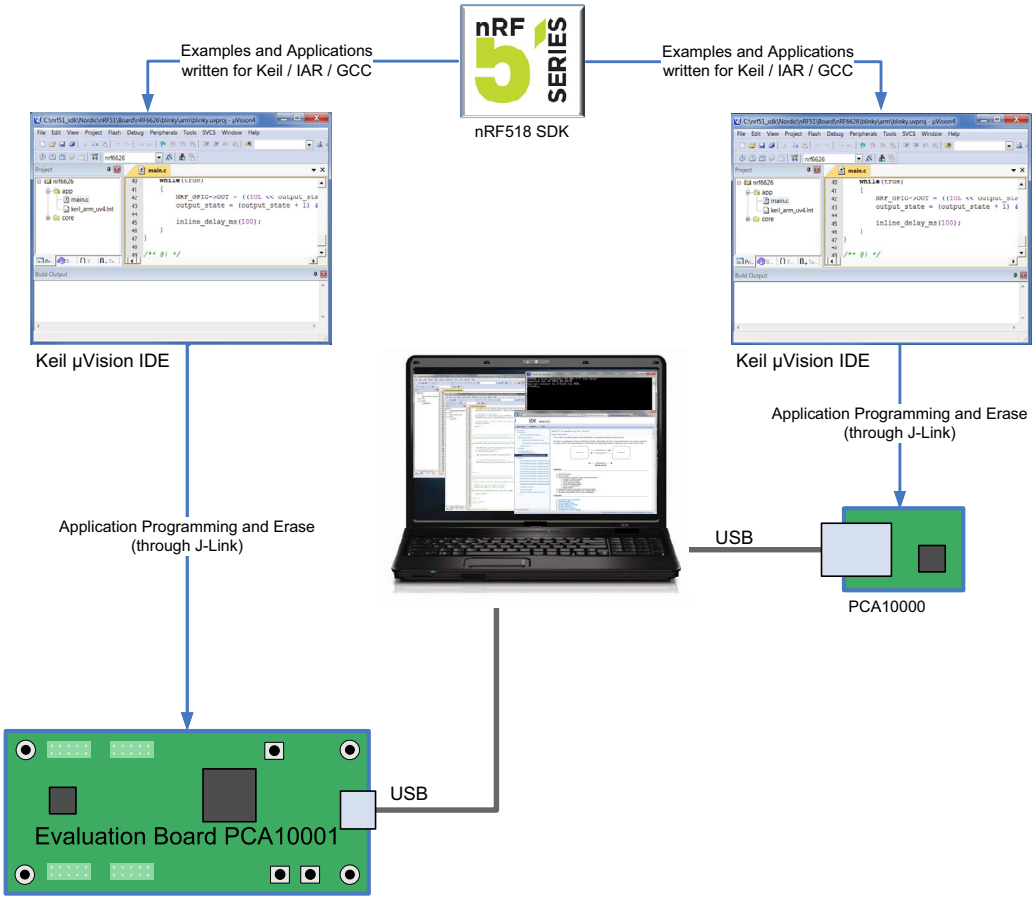


Figure 2 *nRF51822 Evaluation Kit configuration for 2.4 GHz based development*

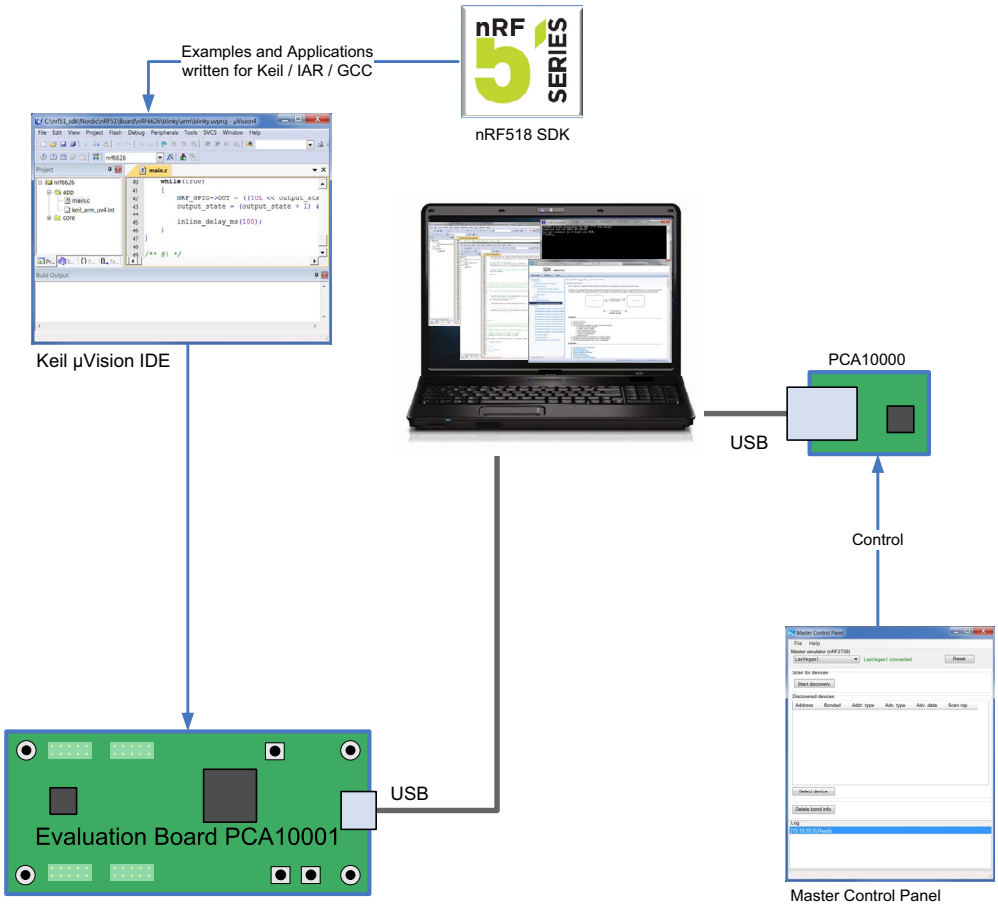


Figure 3 *nRF51822 Evaluation Kit configuration for Bluetooth 4.0 single-mode*

5 Hardware description

This chapter describes the nRF51822 Evaluation Kit hardware.

5.1 nRF51822 Evaluation Kit board (PCA10001)

The nRF51822 Evaluation Kit board (PCA10001) is delivered with an unprogrammed nRF51822 chip.

5.1.1 Key features

The nRF51822 EK board has the following key features:

- nRF51822 IC
- *Bluetooth* low energy compatible
- 2.4 GHz compatible with nRF24L devices
- Buttons and LEDs for user interaction
- I/O interface for plug-in modules
- J-Link OB programming and debugging capabilities
- USB to UART bridge

5.1.2 Hardware pictures

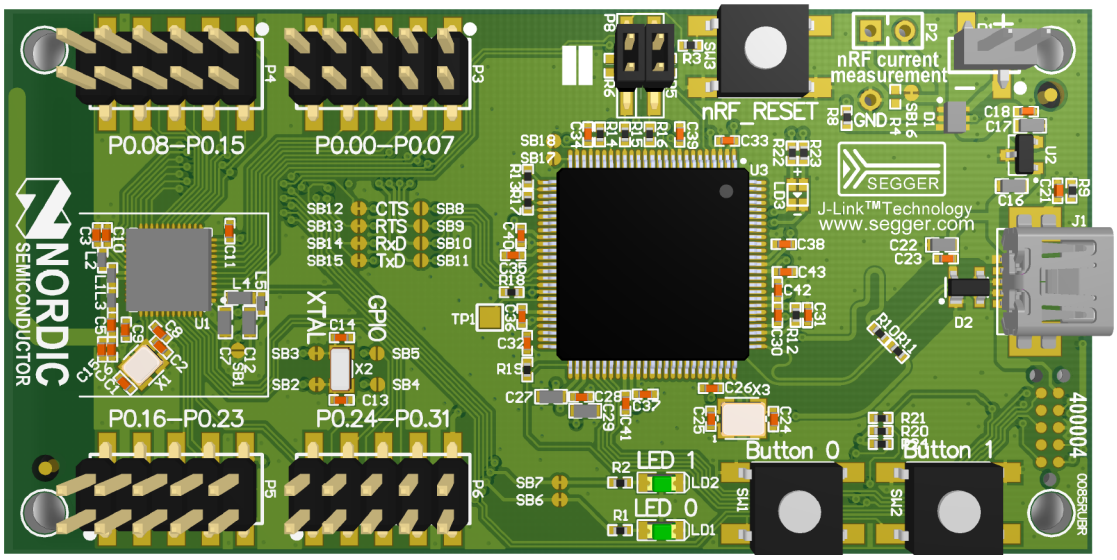


Figure 4 *PCA10001 top*

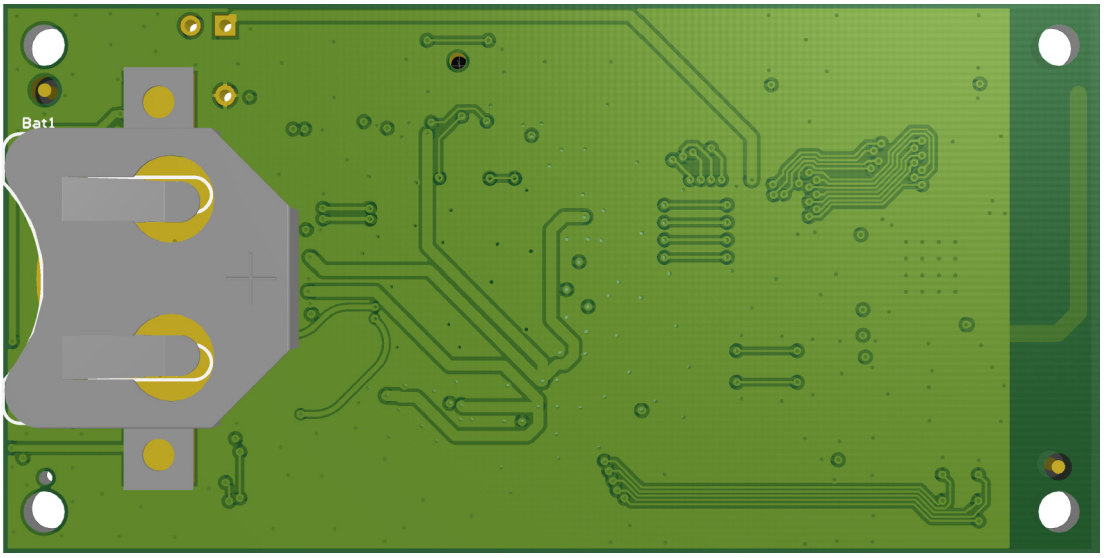


Figure 5 *PCA10001 bottom*

5.1.3 Block diagram

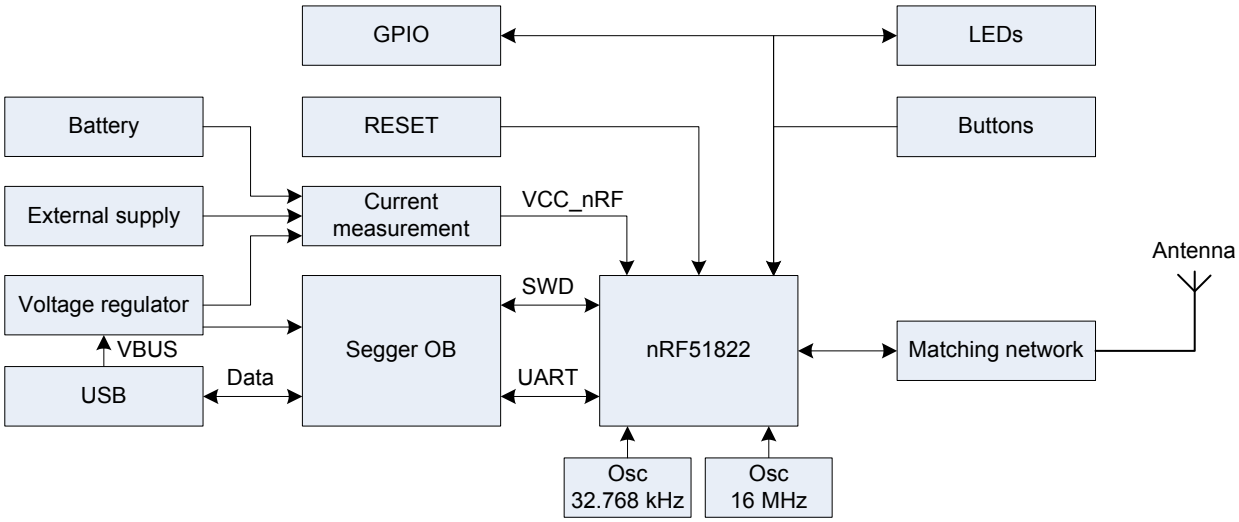


Figure 6 *PCA10001 block diagram*

5.1.4 Reset button

The PCA10001 is equipped with a reset button (**SW3**) for the nRF51822. When debugging the nRF51822 using the J-Link OB, you should use the reset functionality built into the computer software.

5.1.5 Power supply

PCA10001 can be powered through the external 1.8 V to 3.6 V supply (**P1**), from a USB, or from a CR2032 coin-cell battery as seen in ***Figure 7*** and ***Figure 8***.

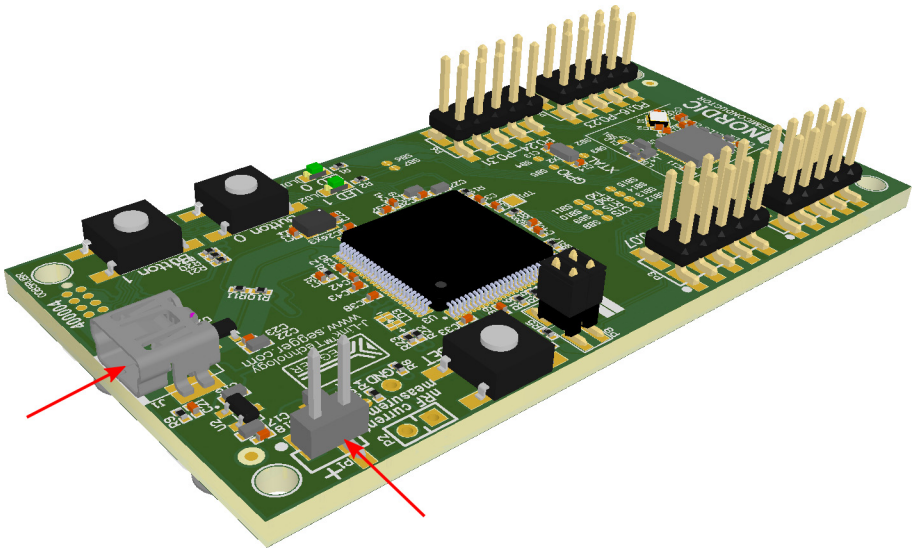


Figure 7 *USB and external power supply*

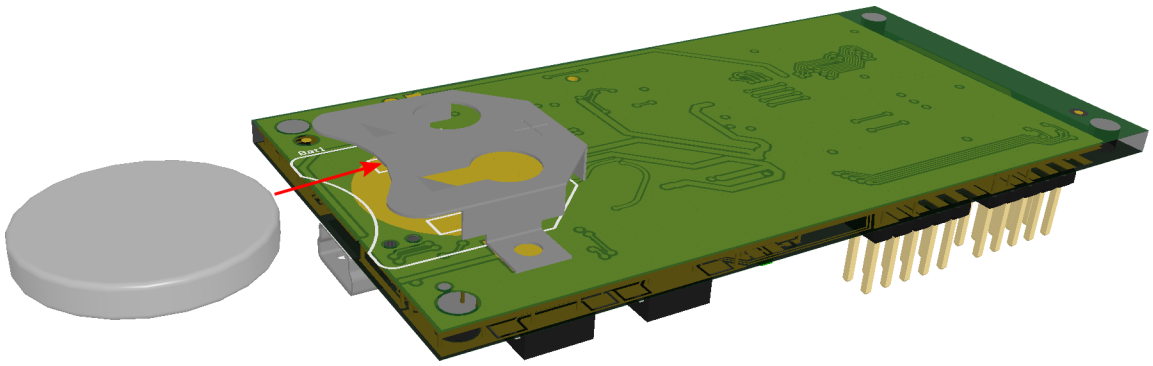


Figure 8 *Coin-cell battery supply*

The 5 V from the USB is regulated down to 3.3 V through an on-board voltage regulator. The battery and external 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.

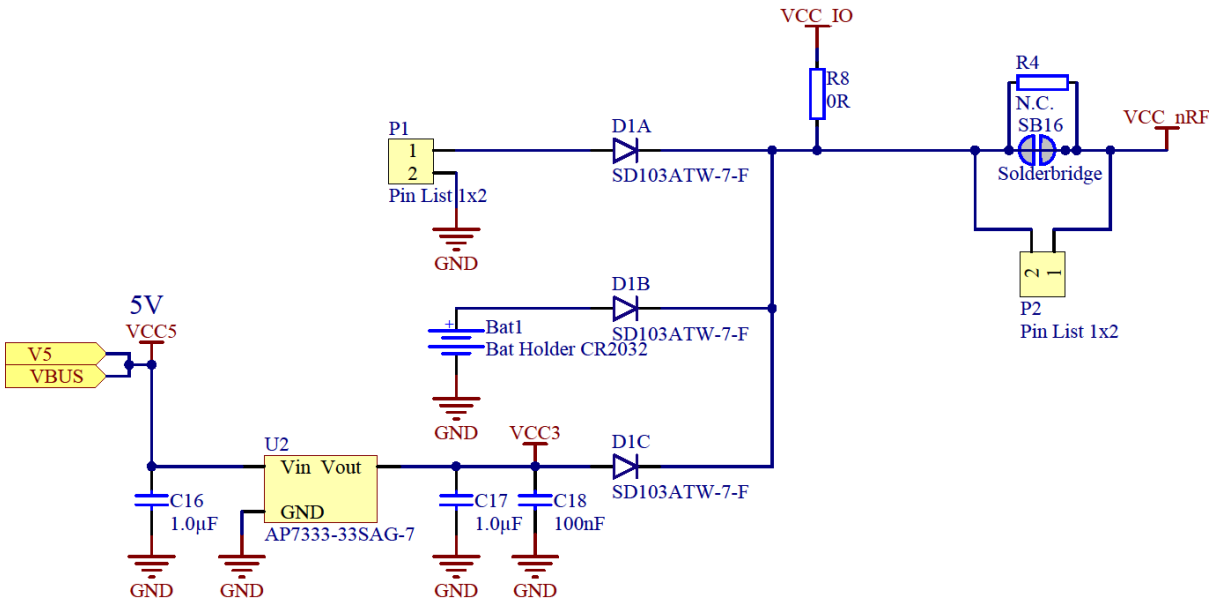


Figure 9 *Power supply circuitry*

Powering the board from a battery or external supply will not power up the J-Link OB circuit on the board and therefore will not work unless it is connected with a USB cable to your computer. To ensure that the J-Link OB will not hold any of the SWD lines while powered down, the two jumpers on connector **P8** should be removed.

5.1.6 GPIO interface

Access to the nRF51822 GPIOs is available at connectors **P3**, **P4**, **P5**, and **P6** on the PCA10001 board.

P3

P0.00	1	2	P0.01	
P0.02	3	4	P0.03	
P0.04	5	6	P0.05	
P0.06	7	8	P0.07	
GND	9	10	VCC	

P6

VCC	10	9	GND	
	8	7	P0.30	
P0.29	6	5	P0.28	
P0.27	4	3	P0.26	
P0.25	2	1	P0.24	

P4

P0.08	1	2	P0.09	
P0.10	3	4	P0.11	
P0.12	5	6	P0.13	
P0.14	7	8	P0.15	
GND	9	10	VCC	

P5

VCC	10	9	GND	
P0.23	8	7	P0.22	
P0.21	6	5	P0.20	
P0.19	4	3	P0.18	
P0.17	2	1	P0.16	

Figure 10 *PCA10001 GPIO pin headers*

Note: Some pins have default settings. Please review the following:

- 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 5.1.8 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 5.1.7*** for more information.
- P0.08, P0.09, P0.10, and P0.11 are by default used by the UART. Please see ***section 5.1.9 on page 19*** for more information.

5.1.7 Buttons and LEDs

The two buttons and two LEDs on PCA10001 are connected to dedicated I/Os on nRF51822. The connections are shown in ***Table 1***.

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

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***.

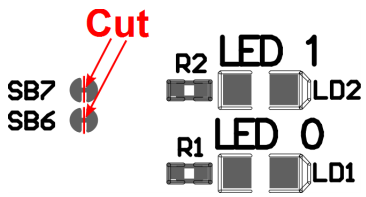


Figure 11 *Disconnecting the LEDs*

The buttons 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.

5.1.8 32.768 kHz crystal

nRF51822 can use an optional 32.768 kHz crystal (X2) for higher accuracy and lower average power consumption. On the PCA10001 module, P0.26 and P0.27 are by default used for the 32 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 kHz crystal can be disconnected and the GIPOs routed to the **P6** connector. Cut the shorting track on SB2 and SB3, and solder SB4 and SB5. See ***Figure 12 on page 19*** for reference.

Note: The 32.768 kHz crystal has to be selected for the *Bluetooth* examples included in the SDK to work.

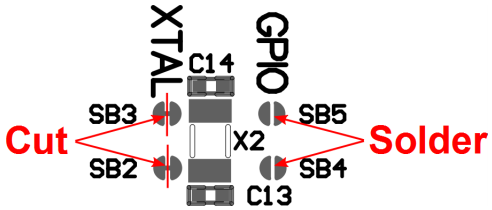


Figure 12 Disconnecting 32.768 kHz crystal and connecting P0.26 and P0.27 to P6

5.1.9 UART configuration

Table 2 shows an overview of the UART connections on nRF51822 and the SEGGER IC. As an option the UART lines can be configured to connect to pin P0.00 to P0.03.

nRF51822			SEGGER IC
Default GPIO	Altered GPIO	UART	UART
P0.08	P0.00	RTS	CTS
P0.09	P0.01	TXD	RXD
P0.10	P0.02	CTS	RTS
P0.11	P0.03	RXD	TXD

Table 2 Relationship of UART connections on nRF51822 and SEGGER

To change configuration, the shorting of solder bridges SB12 to SB15 must be cut and the solder bridges SB8 to SB11 must be soldered. See *Figure 13*.

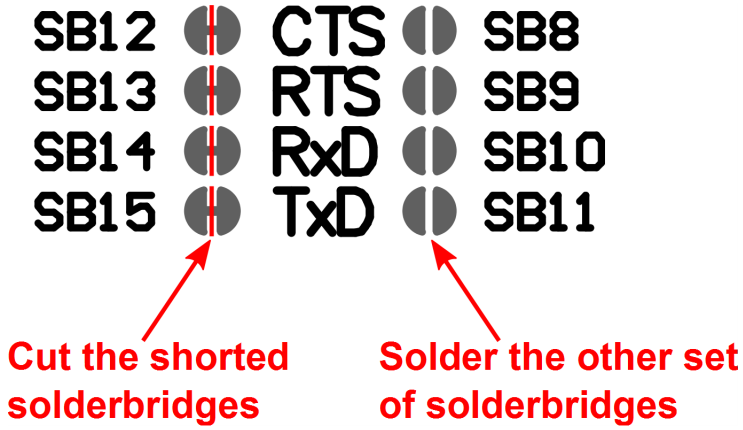


Figure 13 Reconfiguring UART lines

Note: The UART signals by default are routed directly to the SEGGER chip. The pins should only be used for UART without breaking the soldering bridge. To use pins P0.08 to P0.11 (default UART) or P0.00 to P0.03 (optional UART) for other purposes, the shorting of the solder bridges should be removed.

Note: In order to use the USB to UART bridge, the software on the nRF51822 must enable flow control. For details on how to set up the UART with flow control see the nRF51 Series Reference Manual.

5.1.10 Current measurements

The current drawn by the nRF51822 device can be monitored on the PCA10001. To prepare the board for current measurement follow these steps:

- Cut the shorting of solder bridge SB16.
- Mount a 2 pin header on the board in the holes for the P2 connector.

After these modifications there are two ways of measuring the current consumption:

1. Connect an ampere meter between the two pins of connector P2. This will monitor the current directly.
2. Mount a resistor on the footprint for R4. The resistor should not be larger than 10 ohm. Connect an oscilloscope or similar with two probes on the pins on the P2 connector and measure the voltage drop. The voltage drop will be promotional with the current consumption. If a 1 ohm resistor is chosen, 1 mV equals 1 mA.

Figure 14 shows the location of the components related to current measurements.

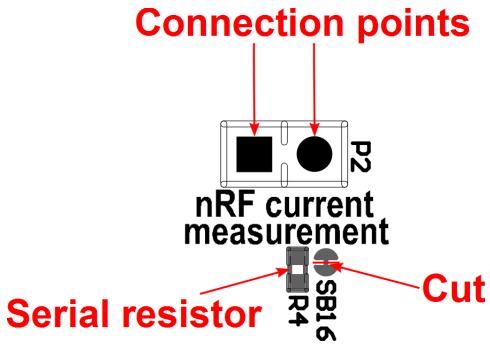


Figure 14 *Current measurements*

5.2 nRF51822 development dongle(PCA10000)

The nRF51822 development dongle (PCA10000) can be used as a development platform for nRF51822. It features an on-board programming and debugging solution from SEGGER (J-Link OB). In addition to radio communication, the nRF51822 device can communicate with a computer through a virtual COM port provided by the J-Link OB microcontroller. The PCA10000 can be loaded with *Bluetooth* Low Energy Master Emulator firmware that when combined with the Master Control Panel, gives you a peer device for nRF51822 that you can use to test the wireless connection.

5.2.1 Key features

The PCA10000 has the following key features:

- nRF51822 IC
- *Bluetooth* low energy compatible
- 2.4 GHz compatible with nRF24L devices
- USB to UART bridge
- SEGGER J-Link OB programming and debugging capabilities

5.2.2 Hardware pictures

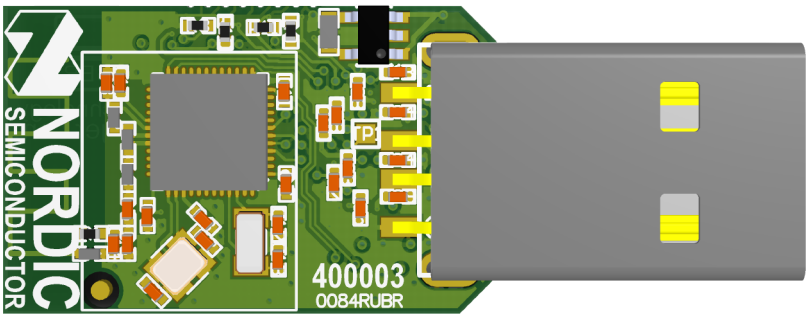


Figure 15 *PCA10000 top side*

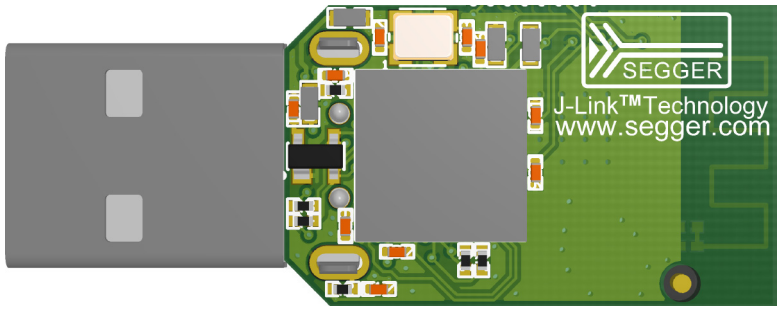


Figure 16 *PCA10000 bottom side*

5.2.3 Block diagram

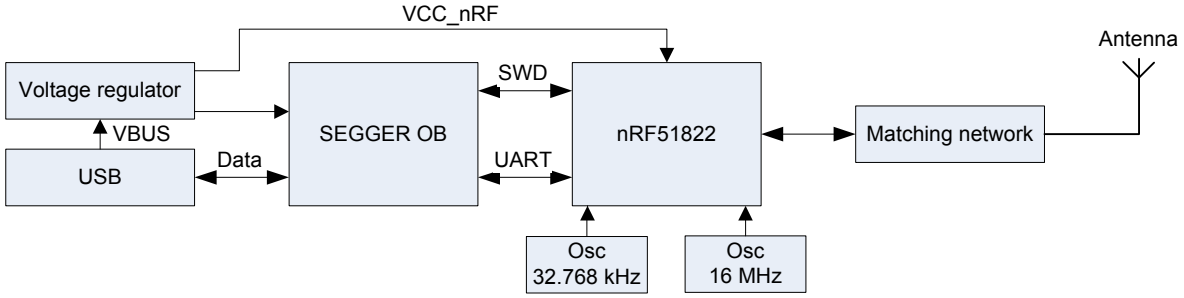


Figure 17 *PCA10000 block diagram*

5.2.4 UART configuration

The UART lines are connected to the pins P0.00 to P0.03 as shown in ***Table 3***.

nRF51822		SEGGER IC
GPIO	**UART**	**UART**
P0.00	RTS	CTS
P0.01	TXD	RXD
P0.02	CTS	RTS
P0.03	RXD	TXD

Table 3 *nRF51822 development dongle UART configuration*

Note: The UART signals are routed directly to the SEGGER chip. The pins should only be used for UART. In order to use the USB to UART bridge, the software on the nRF51822 has to enable flow control. For details on how to set up the UART with flow control see the nRF51 Series Reference Manual.

6 Flash programming and application development

nRF51822 is shipped without pre-programmed software. This gives you the option of developing your application directly onto the chip or alternatively, by using our S110 SoftDevice, which is a *Bluetooth* low energy peripheral protocol stack solution. For more information, see the S110 SoftDevice Specification.

In this chapter we describe how to program the S110 SoftDevice or another application HEX file onto the nRF51822 chip and also how to erase it.

If you want to start developing on the nRF51822 chip without using the S110 SoftDevice see ***section 6.1.5 on page 26***.

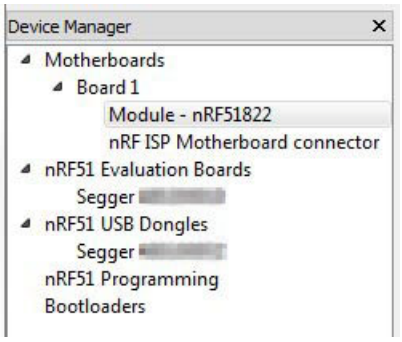
6.1 Programming and erasing flash using nRFgo Studio

Use nRFgo Studio to program a SoftDevice or application HEX file onto the nRF51822 chip or to erase the flash content on a chip that has been programmed.

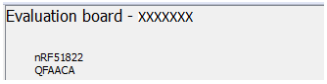
Note: For details on memory organization and protection see the nRF51 Series Reference Manual.

6.1.1 Selecting a board to program

1. Open nRFgo Studio.
2. Select which board to program or erase in the Device Manager pane in nRFgo Studio.



3. The nRF51822 Evaluation Board (PCA10001) and nRF51822 development dongle (PCA10000) hardware will show up under the respective sections ***nRF51 Evaluation Boards*** and ***nRF51 USB Dongles***. The devices are identified by the SEGGER serial number.
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.



5. The nRF51822 Development Kit modules (PCA10004/ PCA10005) cannot be selected directly. These boards must be selected through the J-Link debugger connected to them. To do this, you either select the actual Module located under ***Motherboard > Board****x* or by clicking on **nRF51 Programming**.
6. Select the debugger you want to use from **Segger to use** of available J-Link debuggers.

6.1.2 Identifying the nRF51 chip and chip content

When you select a board, nRFgo Studio identifies the nRF51 chip and how its memory is organized. The following chip and memory information is displayed:

- **Identifying the nRF51 chip:** Identifies the chip by name and variant code (for example, nRF51822 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, did you connect the Segger correctly to a board?*".
- **Memory organization:** Shows how the flash memory is divided, whether into one or two regions and the size of each region. A SoftDevice will always be located at Region 0. You can see the memory size used by the SoftDevice and how much memory is remaining for use.
- **Memory read back protection:** Shows how the read back protection is set. The two possible options are read back protection on Region 0 or read back protection of the whole flash memory. If there is only one region the option is read back protection on (All) or off.
- **Identifying the SoftDevice:** nRFgo Studio tries to identify the firmware located in the chip at Region 0. For the firmware that it recognizes it prints the ID (in clear text) for the unrecognized firmware it prints the FWID number.

Figure 18 *Flash memory organized as one region without read back protection*

Figure 19 *Flash memory organized in two regions with the BTLE stack located a Region 0 and with read back protection on region 0*

6.1.3 Erasing all

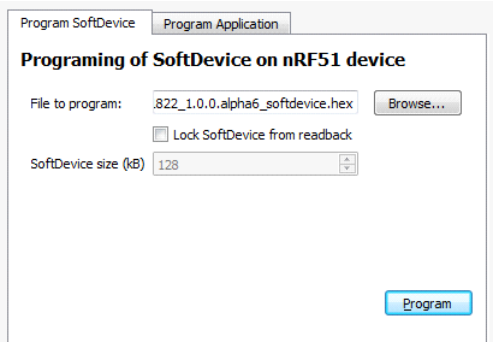
You should use the Erase All function in the following three situations:

- You have a chip that is programmed with a SoftDevice but you want to remove it and have a blank chip.
- You have programmed an application on top of a SoftDevice and used the option "Lock entire chip from read back". Once you have performed Erase All, both the application and the SoftDevice will be erased.
- You have programmed an application on a clean chip using nRFgo Studio with the option "Lock entire chip from read back".

To use the Erase All function, follow the steps in ***section 6.1.1 on page 23***. Then click **Erase all**.

6.1.4 Programming a SoftDevice

This function lets you program the SoftDevice into the chip.



1. Follow the steps in ***section 6.1.1 on page 23*** and then select the **Program SoftDevice** tab.
2. Click **Browse** and select the HEX file to program.
3. Select whether to enable or disable read back protection of Region 0.
4. Set the SoftDevice size. This sets the size of the flash region 0 and will not be available if the size is defined by the HEX file.

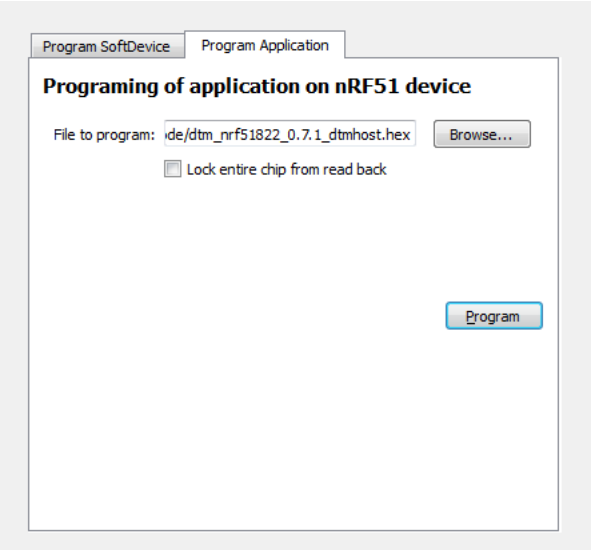
Note: The S110 SoftDevice can be downloaded from www.nordicsemi.com by logging into your My Page account and entering your product key which is printed on the sticker on the outside of your Evaluation Kit.

6.1.5 Programming an application

This function lets you program an application onto the chip.

Before nRFgo Studio starts programming it verifies that the HEX file matches the actual memory configuration, for example, that an application that requires a SoftDevice is being programmed onto a chip that contains a SoftDevice If it matches it continues with the programming, if not it stops the programming and returns an error message.

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 read back protection of the entire chip. If you enable read back protection, you will have to do an Erase All to reprogram the chip again.

Note: A chip that is programmed with the "Lock entire chip from read back" enabled will not work with a development toolchain. To make it work you must perform an Erase All.
The "Lock entire chip from read back" function 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 used data area.

The table below 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 flash and 16 kB of RAM.

Device configuration	SoftDevice		Appl. code start address	Available flash	Appl. data start address	Available RAM
	Flash usage	RAM usage				
Blank chip	0 kB	0 kB	0x0	0x40000	0x2000.0000	0x4000
SoftDevice A	64 kB	2 kB	0x10000	0x30000	0x2000.0800	0x3800
SoftDevice B	128 kB	6 kB	0x20000	0x20000	0x2000.2000	0x2000

Table 4 *SoftDevice memory layout*

The Product Specification describes the total flash memory and RAM available in the device. The amount of flash memory and RAM used by the SoftDevice is described in the SoftDevice Specification.

There are two ways to configure the memory layout:

- Using Keil IDE
- Using a Scatter file

Note: The example code given by Nordic Semiconductor configures the memory layout in Keil IDE. Scatter file loading is not available when using the evaluation version of 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.** The Options for Target dialog box opens.
2. Select the **Linker** tab.

3. Check **Use memory layout from Target Dialog** and click **OK**.

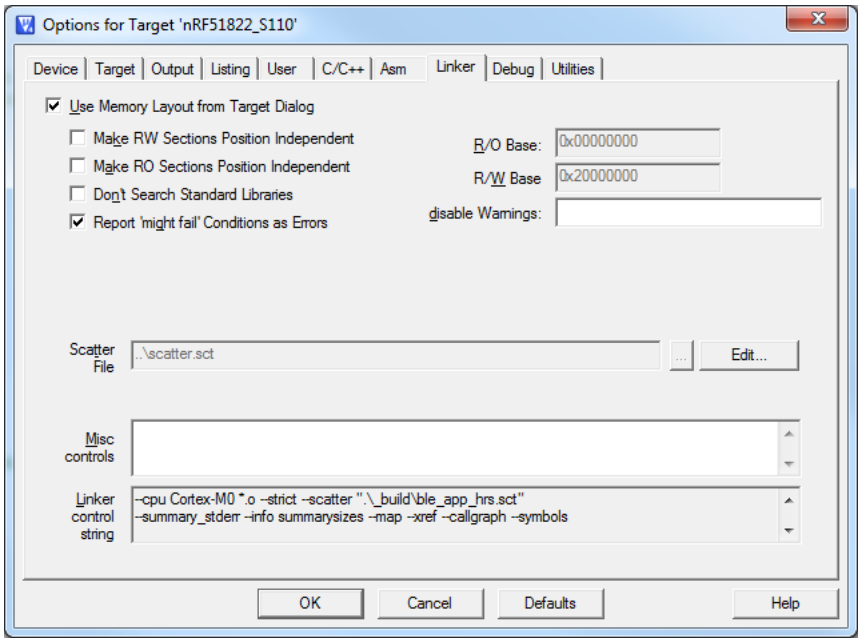


Figure 20 *Keil memory layout*

Values for **Start** and **Size** in **Read/Only Memory Areas** and **Read/Write Memory Areas** must be defined.

Below is an example configuration for an application using a chip with 256 kB of flash and 16 kB of RAM and a SoftDevice using 128 kB of flash and 8 kB of RAM (SoftDevice B described in ***Table 4 on page 28***).

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

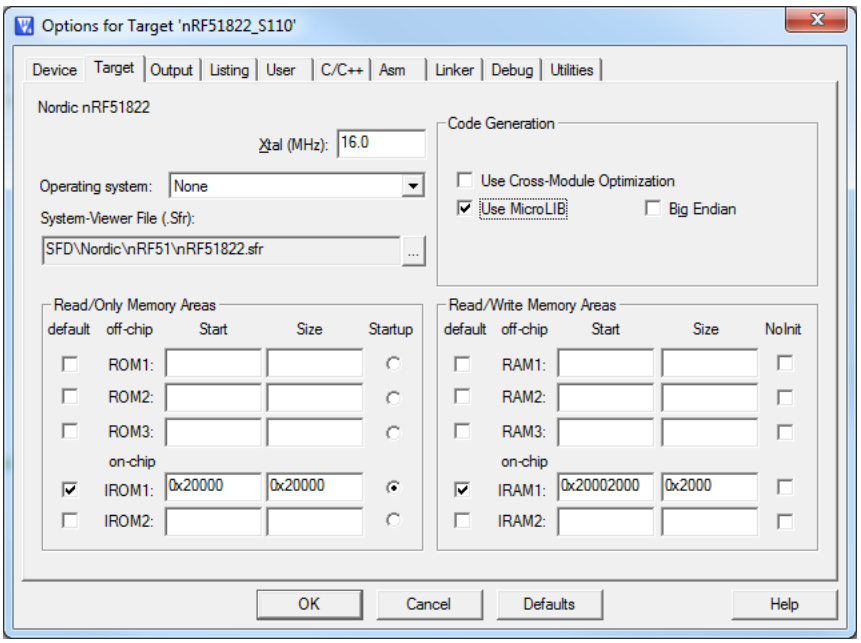


Figure 21 *Memory layout with example SoftDevice*

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

Table 5 *Memory layout*

6.2.1.2 Scatter file

A scatter file specifies to the linker how to gather re-locatable files into load regions (areas in flash/ROM memory). It also informs the linker where, at run time, data (and possibly code) should be located into execution regions. Read/write data may have to be copied from flash to RAM (if compile time initialized) or zeroed, during the boot process. This is done by a library function called scatter loader, which works according to the descriptions set up by the scatter file in the flash/ROM.

Note: The scatter file feature is not available with a Keil Lite installation.

The application scatter file must specify addresses for the load region(s) that are within the physical address range of the flash memory, but not overlap the address range used by the SoftDevice. Similarly, the RAM execution region addresses must be within the physical RAM available, and not overlap the RAM area used by the SoftDevice.

The application interrupt vector should be located at the first available flash address on the application. To locate the reset vector at the beginning of the available flash to the application, the assembler file defining the reset vector declares an area RESET. That definition can then be used in the scatter file as shown in the example below. A template assembler file defining the reset vector is provided in the nRF51 SDK.

Note: The Product Specification describes the total flash memory and RAM available in the device. The amount of flash memory and RAM used by the SoftDevice is described in the SoftDevice Specification.

In the following scatter file example, the flash size of the device is assumed to be 256 kB (0x00040000) and the RAM size of the device is assumed to be 16 kB (0x00004000). RAM is always located from execution address 0x20000000. The example will assume SoftDevice B described above will be used. The range of flash from 0 to 0x00020000 (128 kB) is used by the SoftDevice protocol stack. The range of RAM from 0x20000000 to 0x20002000 (8 kB) is used by the SoftDevice protocol stack.

An application may define multiple load and execution regions if it is desired.

```
Example Scatter file
; *************************************************************
; *** Scatter-Loading Description File                      ***
; *************************************************************
; Scatter file for a program IN APPLICATION SPACE,
; accessing the SoftDevice B through SVCs.
; Make sure that neither Load Region addresses below 0x00020000 nor
; RAM Execution Region addresses below 0x20002000 are used -
; Those regions are reserved for the SoftDevice (protocol stack).

LR_IROM1 0x00020000 0x000020000  {    ; load region size_region
  ER_IROM1 0x00020000 0x000020000  {  ; load address = execution address
   *.o (RESET, +First)
   *(InRoot$$Sections)
   .ANY (+RO)
  }

  RW_IRAM1 0x20002000 0x00002000 { ; RW data
    .ANY (+RW +ZI)
  }
}
```

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. Call stack size required by the SoftDevice varies between devices and protocol stack versions, and is supplied in the 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

                AREA    STACK, NOINIT, READWRITE, ALIGN=3

Stack_Mem       SPACE   Stack_Size

__initial_sp
```

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** 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. For a proper debugging experience the following are advised:

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

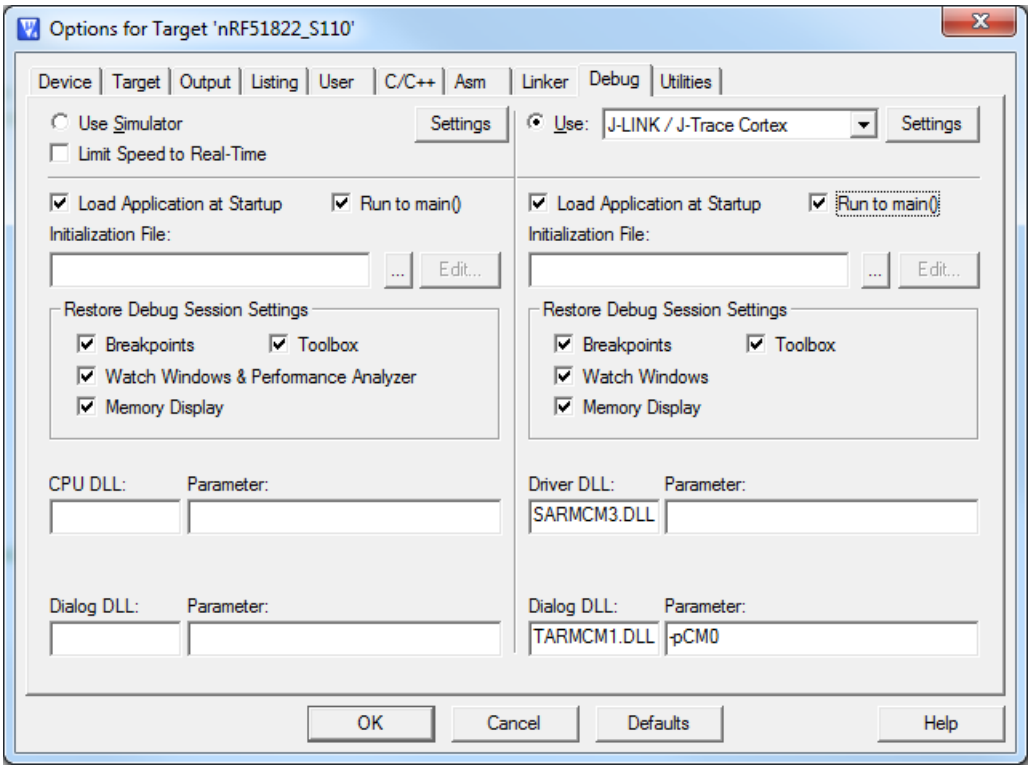


***Figure 22** Debugger options*

6. Click the **Settings** button next to the Use field. Information about debugging protocol and maximum speed needs to be provided.
7. In the **Port** drop-down, **SW** must be selected.
8. In **Max Clock** the maximum speed for the debugging port cannot be exceeded (1 MHz).

A proper configuration is shown in ***Figure 23 on page 33***.

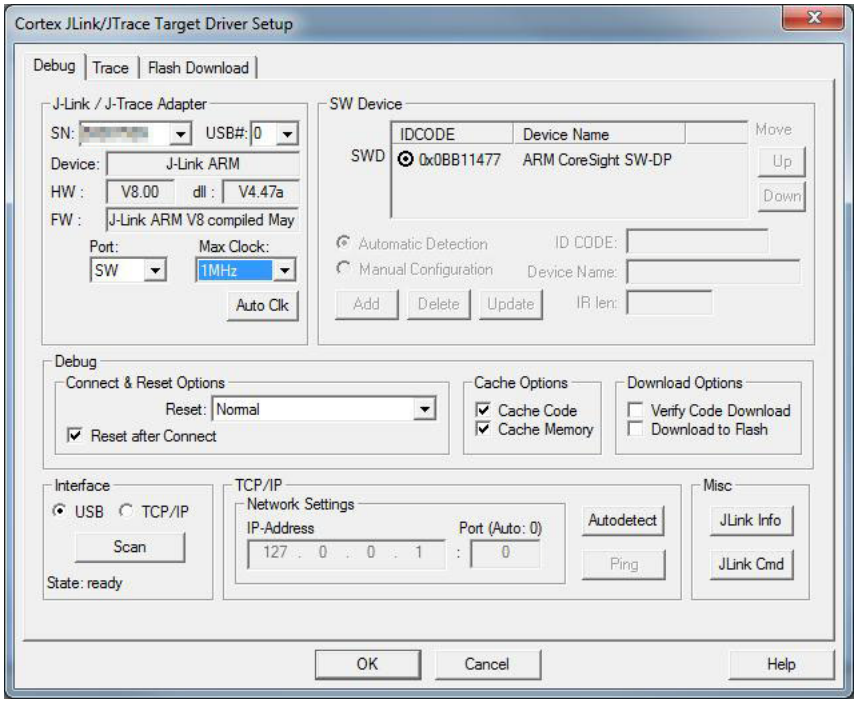


***Figure 23** Target driver setup*

Debugging is initiated by clicking **Start/Stop Debug Session** in the Keil IDE.

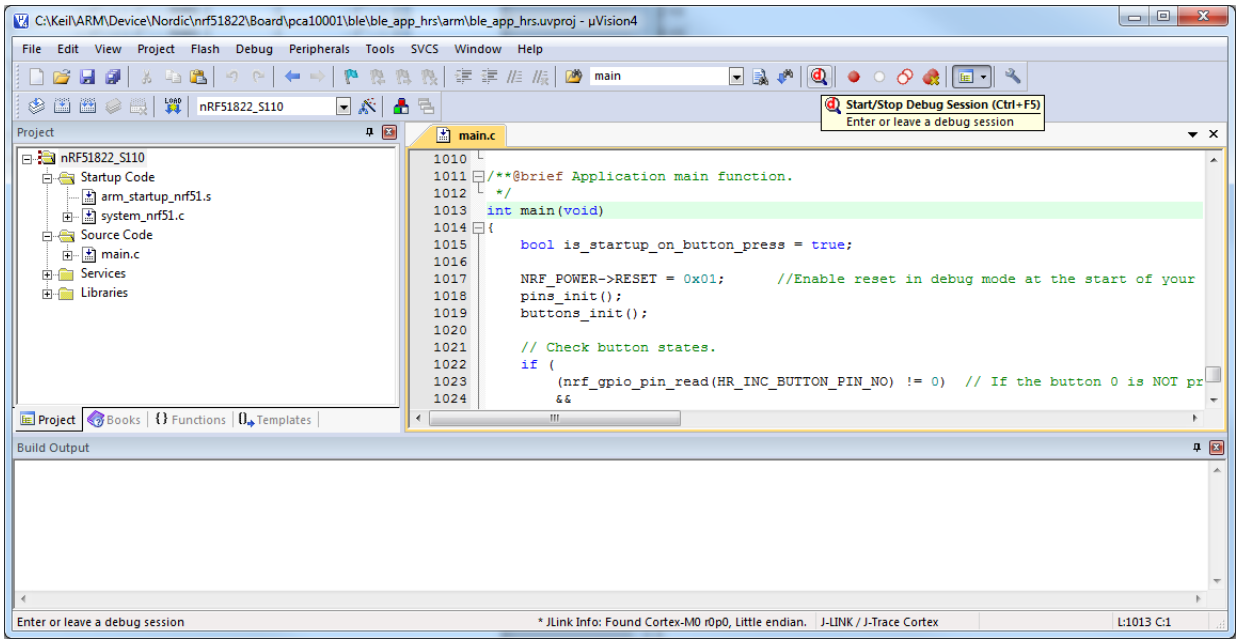


***Figure 24** Debugger initiation*

Note: If you are using Keil v4.53 or earlier, the default **JLinkSettings.ini** file created automatically by the debugger in the project root is not adequate for debugging and downloading. Copy the **JLinkSettings.ini** file, present in any Nordic Semiconductor project, into the root folder before debugging. The file includes the proper settings for use with nRF51 series devices.

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.

6.2.5 Programming the device

To guarantee the correct functionality of the SoftDevice, the microcontroller includes a Memory Protection Unit that prevents access to certain resources. The debugger will read this area as 0x0000 (no operation instruction).The flash area occupied by the SoftDevice is write and erase protected. When the SoftDevice radio stack is enabled, the Memory Protection Unit implements a write protection to certain peripherals used by the protocol stack. Protected peripherals are described in the SoftDevice specification.Once the debugger is properly configured and the application code compiled and linked, downloading the application can be easily done using the Keil IDE download button.

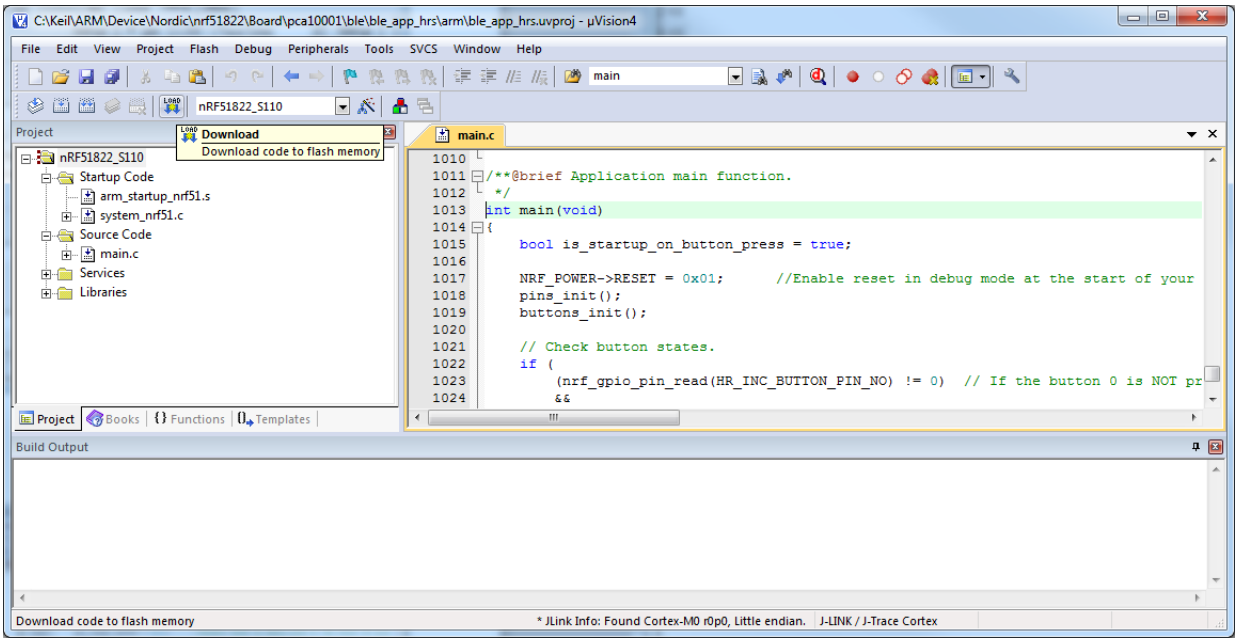


Figure 25 *Download using Keil IDE*

To configure 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 the **Settings** button to configure.
4. Select the **Program** check box.
5. Choose your desired erase option.
6. Select the download algorithm used by Keil IDE.
7. Click **Add** and select the nRF51xxx algorithm from the list.

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 flash.

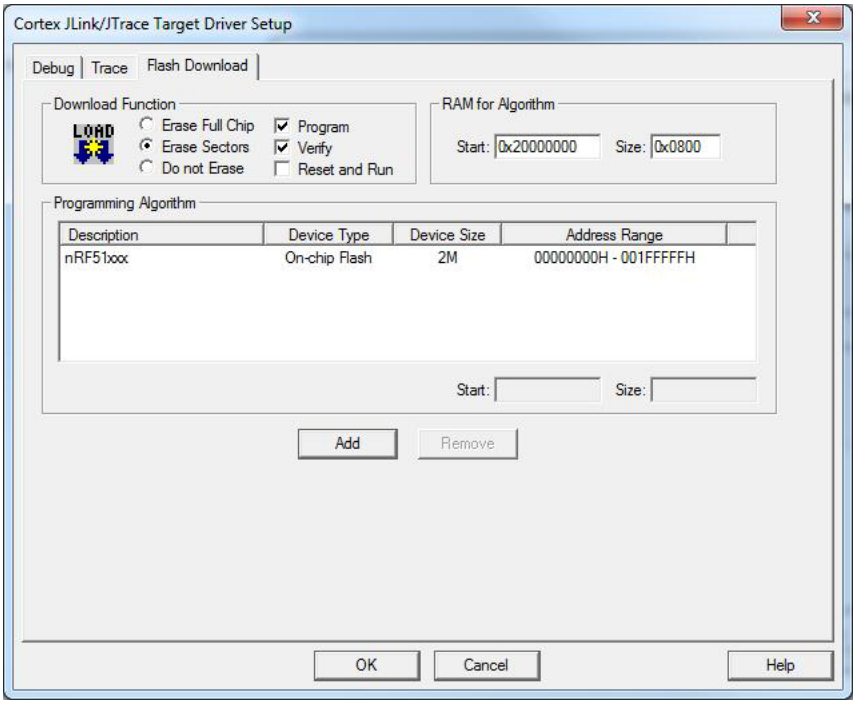


Figure 26 *Selecting erase option*

In the Utilities tab, **Use Target Driver for Flash Programming** option must be selected and the available debugger chosen from the list as shown in ***Figure 27***.

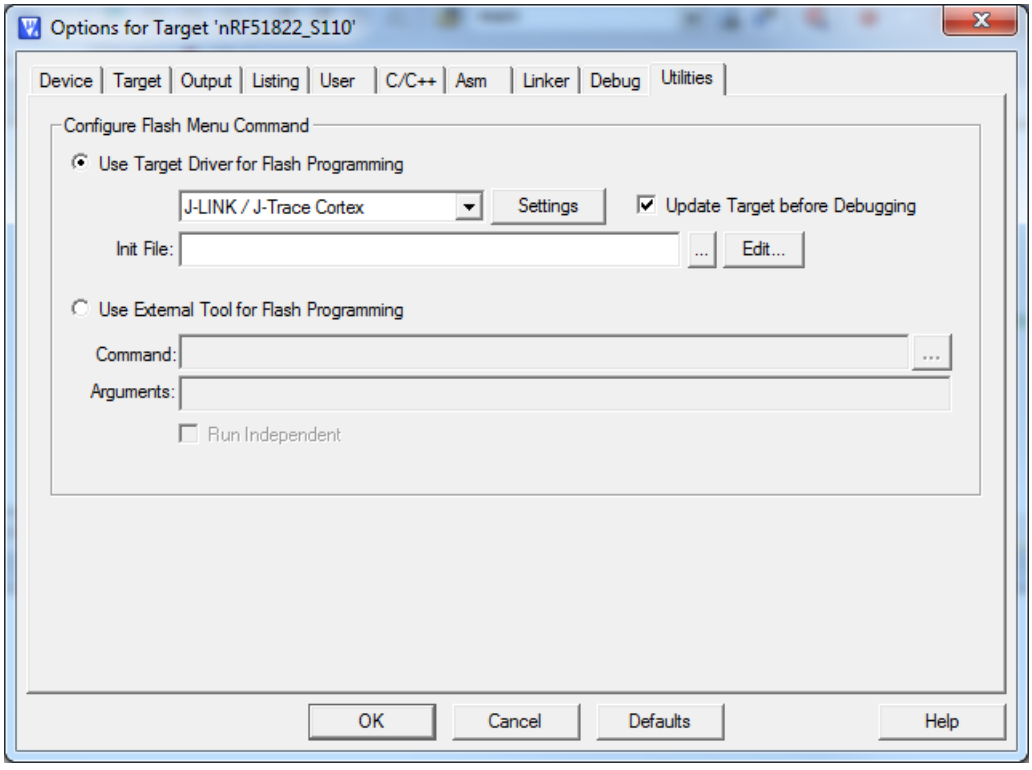


Figure 27 *Debugger selection*

6.2.6 Erasing the device

The flash area available for user application can be erased using the download function in Keil IDE.

1. In the Settings menu select the **Utilities** tab.
2. Select **Erase Full Chip**.
3. Uncheck **Program** and **Verify**.

A normal download procedure will erase the device application flash area.

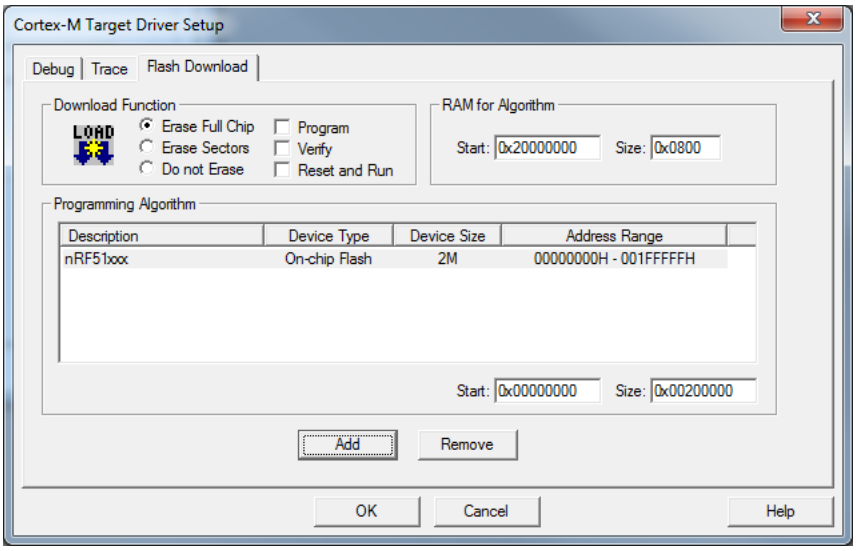


***Figure 28** Erasing*

To erase the whole device, including the SoftDevice, refer to section ***section 6.1.3 on page 25*** for instructions.

7 Debugging the nRF51822chip

For debugging with SEGGER J-Link, see ***Appendix A: on page 45***. For general information of how to debug using Keil µVision IDE, we refer to online documentation from Keil at http://www.keil.com/uvision/debug.asp.

Configure the debugger in the **Project** menu and selecting **Options for Target**. Select **Debug**. When you have set this up, you can enter debugging mode by clicking the **Debug** button or **CTRL**+**F5**.

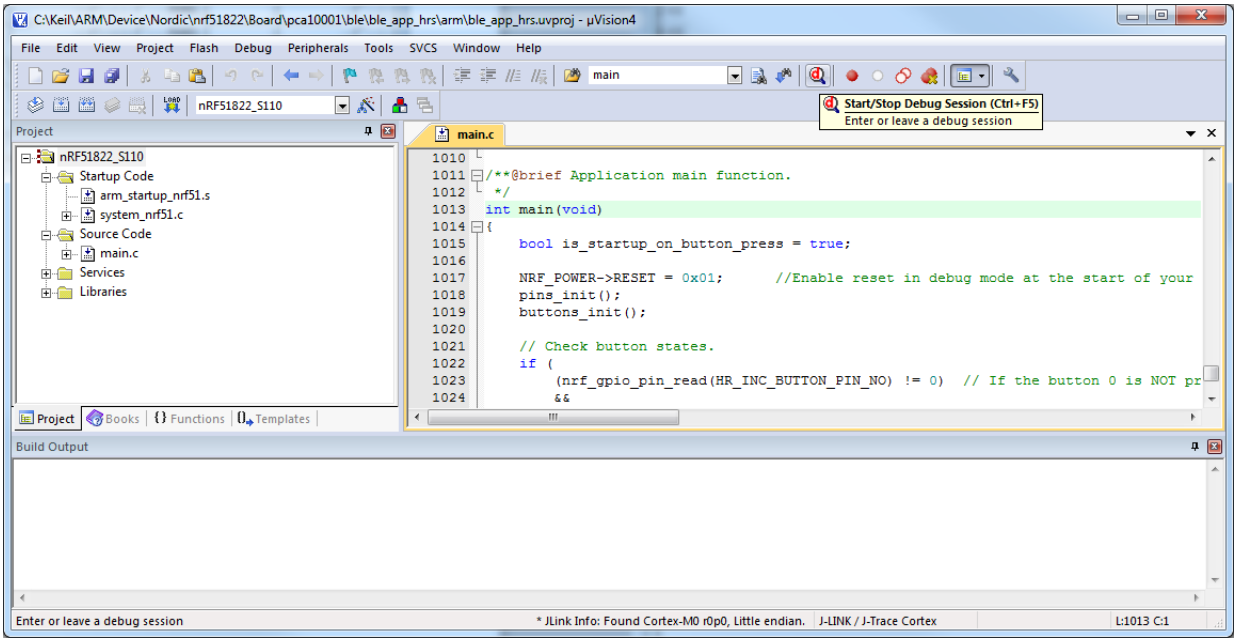


Figure 29 *Start debugging mode*

7.1 nRF51 debug features and precautions

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

7.1.1 System Viewer Windows

The System Viewer Windows enables access to view device peripheral contents, see ***Figure 30 on page 38***. Any values in the registers may be viewed from its respected window, as seen in ***Figure 31 on page 39***. More information can be found at: http://www.keil.com/uvision/db_view_sysview.asp.

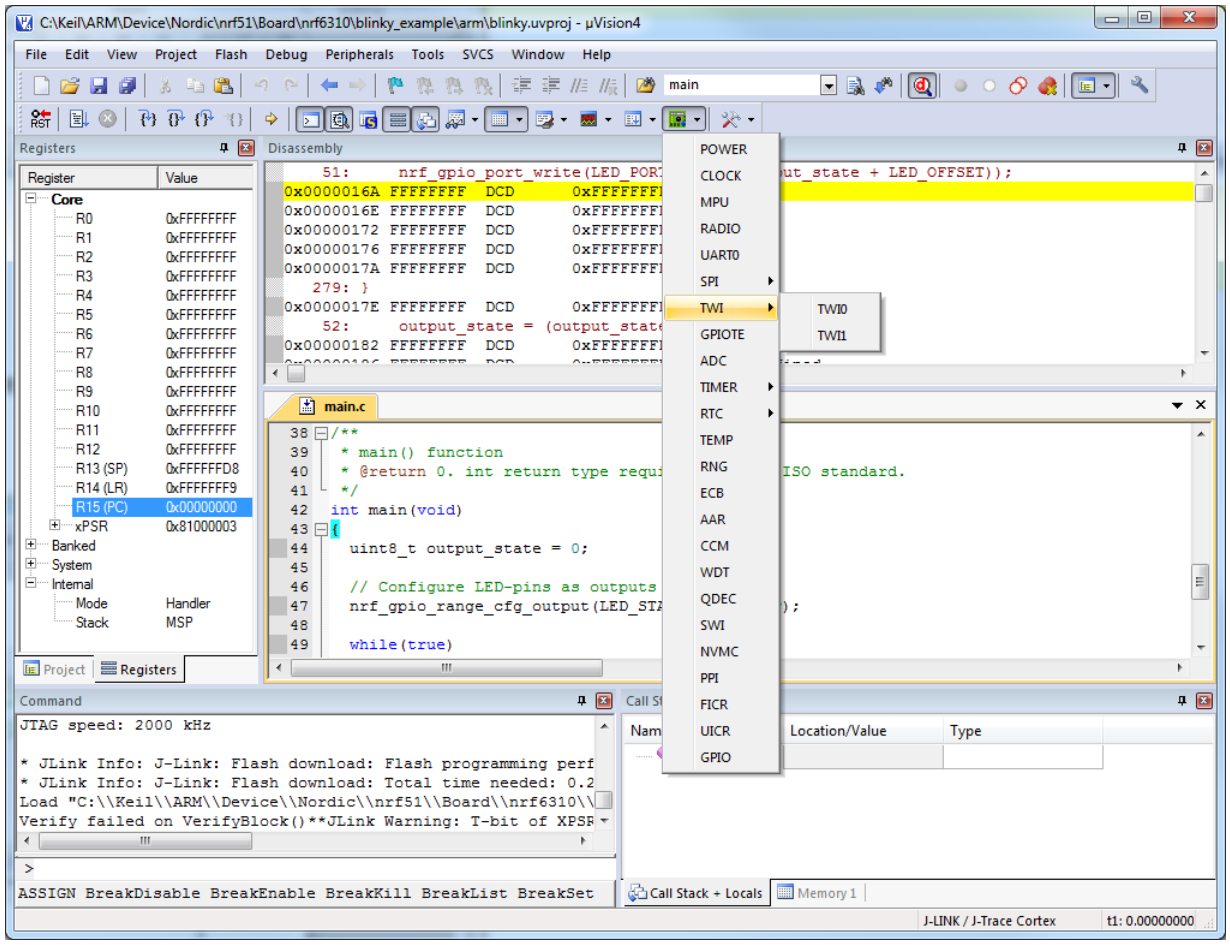


Figure 30 *System Viewer Windows*

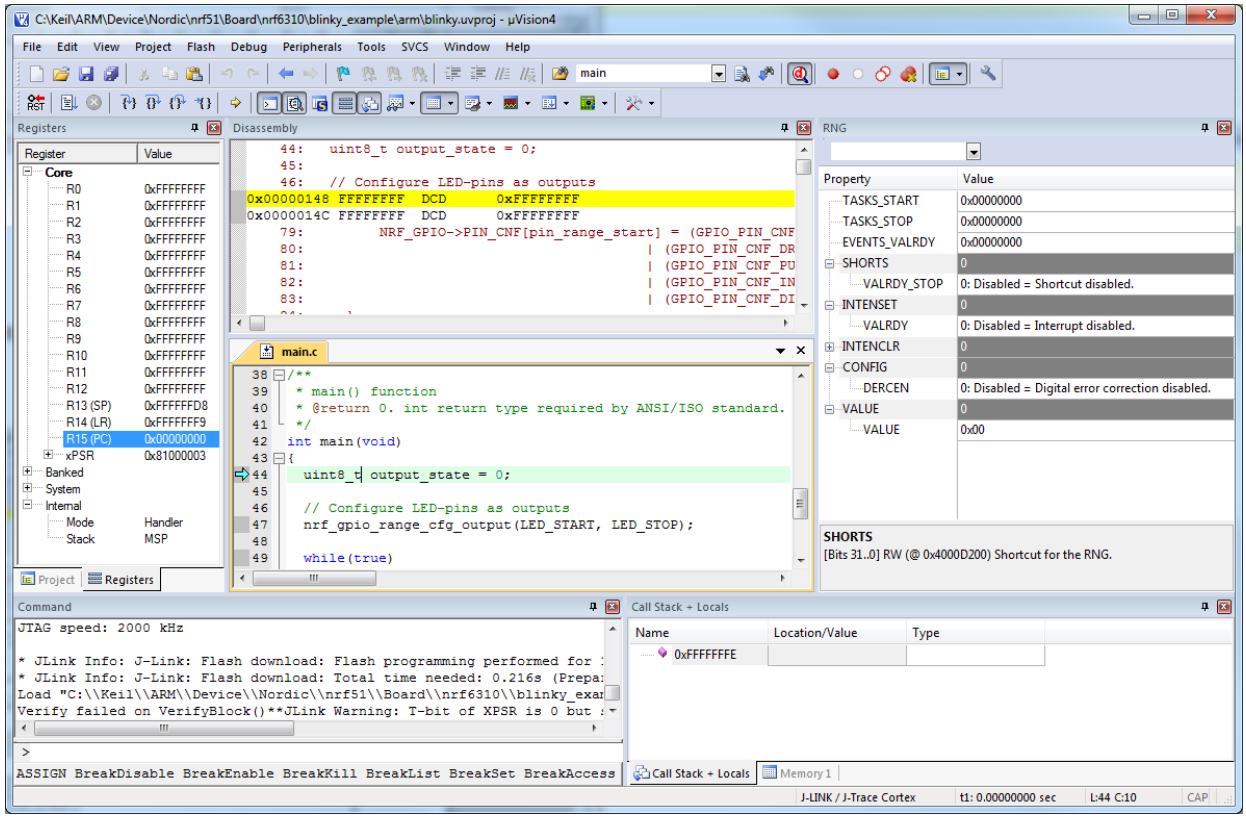


Figure 31 System viewer window of RNG register

7.1.2 Debugging an application when a read back protected SoftDevice is present

Debugging applications with a SoftDevice present behaves as described in http://www.keil.com/uvision/debug.asp, except when program counter is in Region 0 on a SoftDevice with read back 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. See the SoftDevice Specification for more information.

Information on the SoftDevice configuration and memory resource mapping can be found in the SoftDevice Specification.

Note: Stepping into instructions that interact with the protected area (SVC calls) may use a lot of time before returning to the next instruction (sometimes greater than 10 minutes). We recommend avoiding single stepping to the protected area, but instead set break point right after SVC calls while debugging and run the application to the actual breakpoint see ***Figure 33 on page 41***. The “step over” function **F10** may also be used instead to step over SVC calls to avoid delays when entering read back protected area.

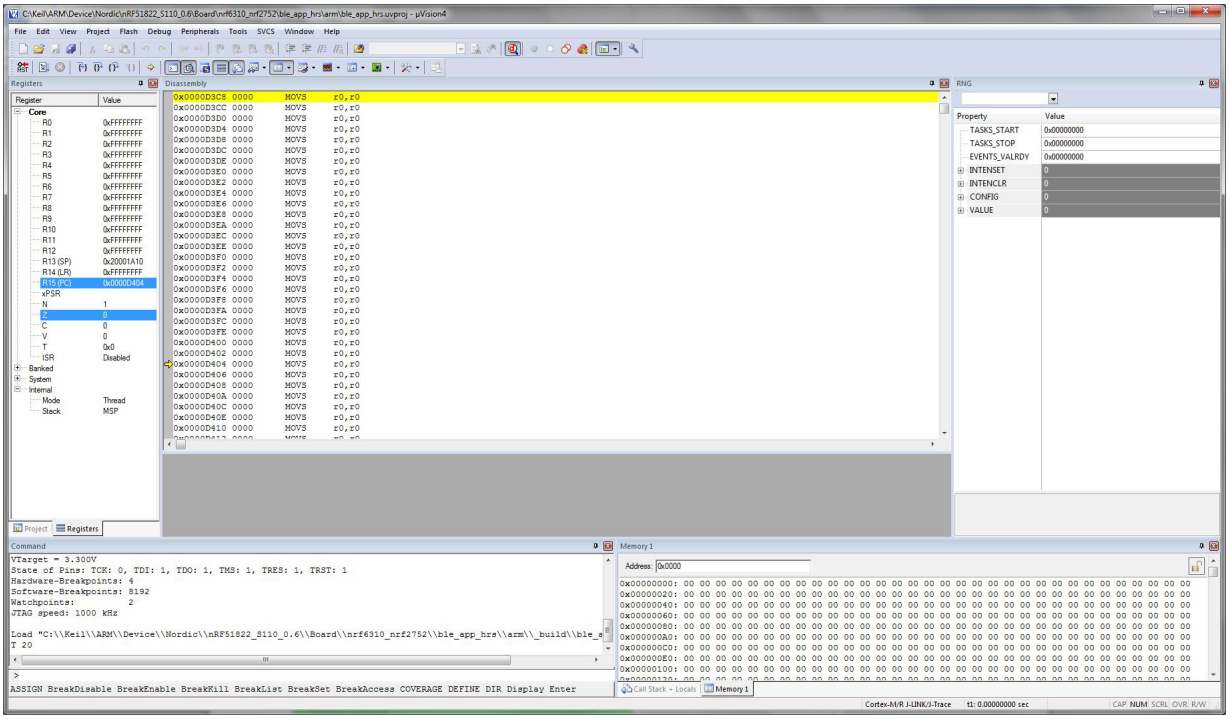


Figure 32 *Debugger information for a setup with SoftDevice enabled in protected area*

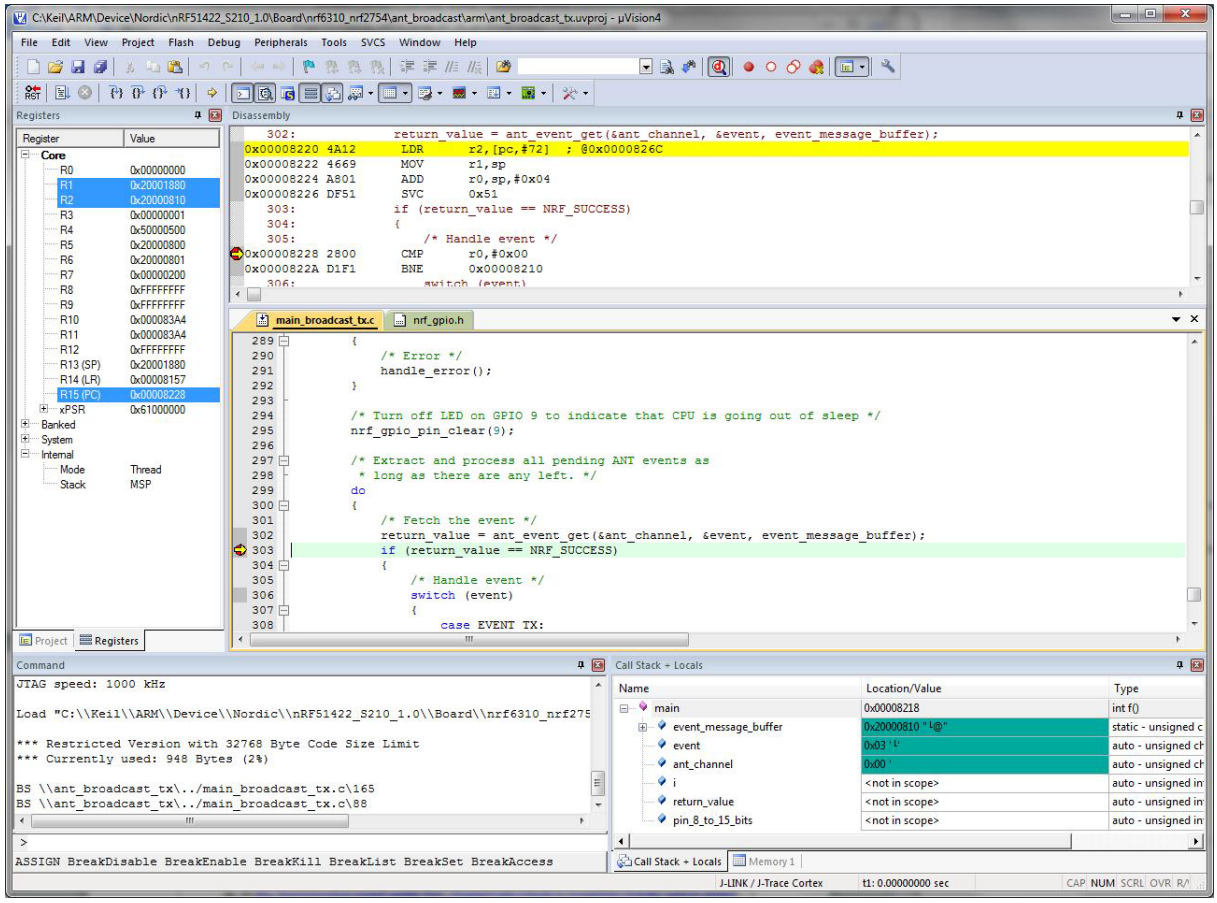


Figure 33 *Setup with breakpoint after SVC call*

7.1.3 Setting a breakpoint using SEGGER J-Link debugger

For SEGGER version 4.52c or earlier, if a breakpoint is set using the SEGGER J-Link debugger, while the system is running, the CPU will be halted for approximately 5 - 10 ms. The unavailability of the CPU during this time may cause the SoftDevice to go out of sync, leading to an invalid state. When this is discovered, the SoftDevice will assert. We encourage you to avoid setting breakpoints on a running system.

8 Software Development Kit

The nRF518 Software Development Kit (SDK) for the nRF51822 chip and the S110 SoftDevice enables you to develop applications for the following protocol stacks:

- *Bluetooth* low energy (using the S110_nRF51822 SoftDevice)
- Proprietary 2.4 GHz, including Nordic's Gazell protocol
- Non-concurrent combinations of *Bluetooth* low energy and proprietary 2.4 GHz

8.1 Installing the nRF518 SDK

The nRF518 SDK is a part of the downloadable content available from your My Page account, see ***section 3.2 on page 10***. 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, Master Control Panel, and/or Custom install.

- **Keil MDK support** - installs Keil µVision 4 example project files, flash programming algorithm for J-Link debugger, and Nordic nRF51 series device database file for Keil.
- **Custom install** - installs a software archive to a customized location.

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

9 Troubleshooting

The nRF51822 device on the PCA10001 does not respond when I try to contact it. What has happened?

Verify that both jumpers on connector **P8** on the PCA10001 are in place.

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

This is a known limitation with µVision in MDK v4.53 or earlier that will be fixed on 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 or nRF518 SDK.

Installing or reinstalling either the nRF518 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).

I have a problem sending/receiving data using the USB to UART bridge.

In order to use the USB to UART bridge the software on nRF51822 has to enable flow control.

When reconnecting the PCA10001 (using the USB cable) the terminal program running on your computer has to be restarted (you should wait for it to end before disconnecting). Otherwise it locks up the serial port and the terminal.

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 23)***, 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 39*** for more details.

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

1. Cycle the power to the nRF51 chip before you start the Recover application.
2. Under Device Manager select the J-Link LITE CortexM, nRF51 Evaluation Board, or nRF51 development dongle (identified by the SEGGER device serial number) to access the Recovery button. It is recommended that you configure the SWD debug interface to run at a high frequency– for example 5 MHz to 10 MHz.

Note: The Recover function will erase all application firmware apart from the one that comes pre-programmed on the chip.

The drop-down menu in the Master Control Panel doesn't display any serial numbers. What has happened?

Verify that the Master Control Panel software and the drivers for SEGGER OB (JLinkCDCInstaller) have been installed and that the nRF51 development dongle (PCA10000) has been plugged into a USB port on your computer.

The Master Control Panel connects to the nRF51 development dongle (PCA10000) but reports "No response from master emulator" in the Log?

You haven't programmed the nRF51 development dongle with the Master Emulator Firmware before starting to use it. See **Scan for available Bluetooth low energy devices** in ***chapter 2 on page 5*** for details on how to program the Master Emulator Firmware.

My project used to work, but after trying out another project using the SoftDevice, it fails.

Ensure that the memory layout in your project matches the memory layout on the chip. See ***section 6.2.1 on page 28*** on how to set up memory configuration.

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

The following describes the steps required to install the software and use the SEGGER J-Link Lite debugger with Keil µVision for nRF51 series devices based on J-Link software version 4.52b or later.

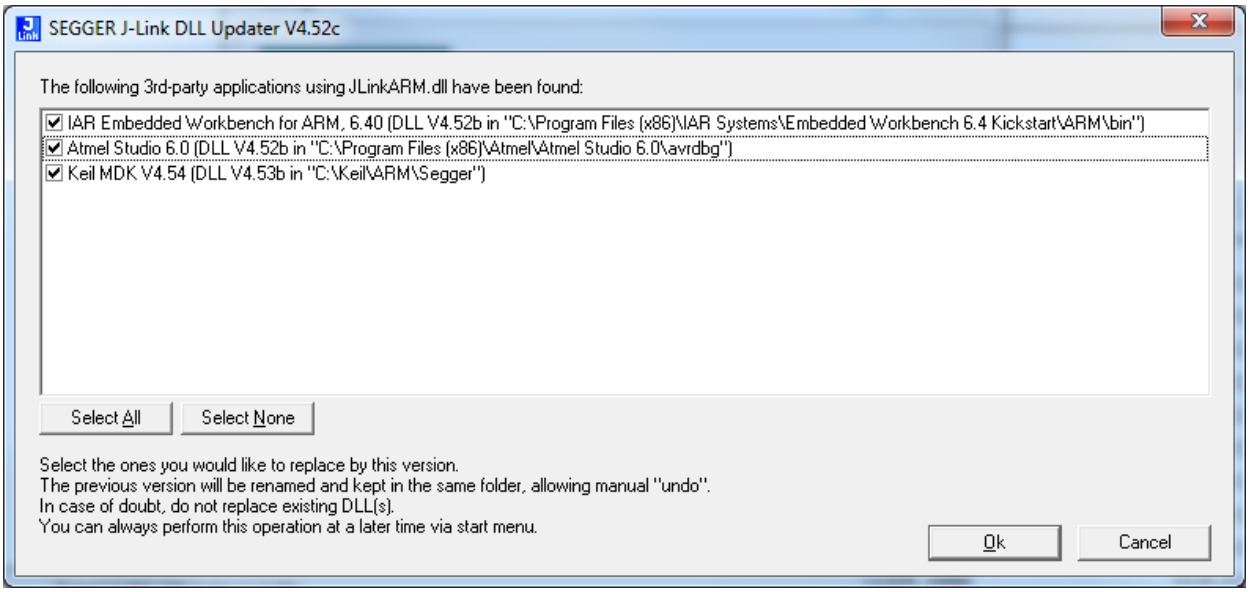
Prerequisite

You should have Keil µVision with ARM-MDK that is tested and working with MDK version 4.54.

Note: All projects in the nRF518SDK 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 (version 4.52b or later) and documentation pack for Windows from http://www.segger.com/jlink-software.html. The serial number from the SEGGER chip on the board.
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 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 in the Evaluation Board (PCA10001) with a USB cable. The LD3 LED will blink while the driver installation occurs. Wait until the LED is continually lit, without blinking

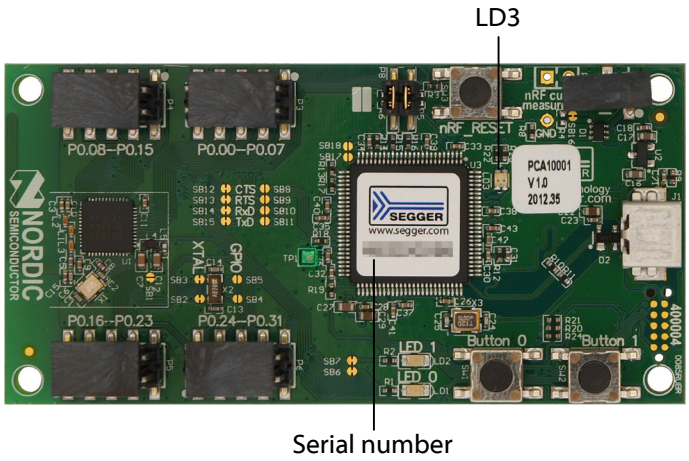


Figure 34 *J-Link Lite CortexM-9 serial number location*

Configuring KEIL projects for SEGGER debugger for first time use

Create **JLinkSettings.ini** file with the contents shown in ***Figure 40 on page 50***. The file **JLinkSettings.ini** should be saved in the same folder as Keil µVision project (uvproj) file.

1. Open Keil µVision IDE by double-clicking an example project file. The Target Options window will open.
2. Click the **Target Options** button on the toolbar or click **Project** menu and select **Options for Target**

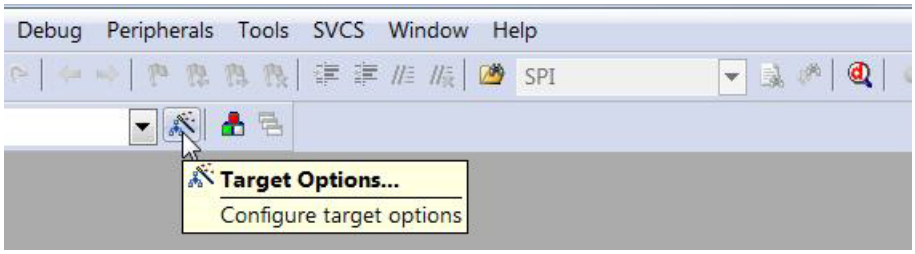


Figure 35 *Keil Target configuration*

3. Under the **Debug** tab in the Use list, select **J-LINK / J-Trace Cortex** option as shown in ***Figure 36***.

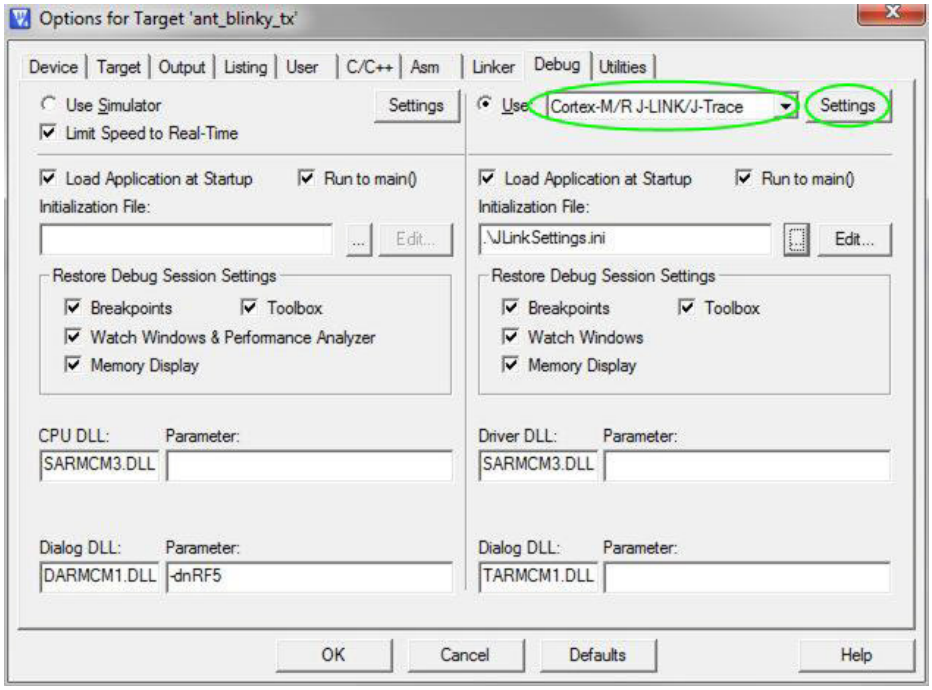


Figure 36 *Selecting JLink debugger in Keil*

4. Click the **Settings** button shown in ***Figure 36.*** Both the SEGGER Control Panel and the Keil Target Driver Setup will open.

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

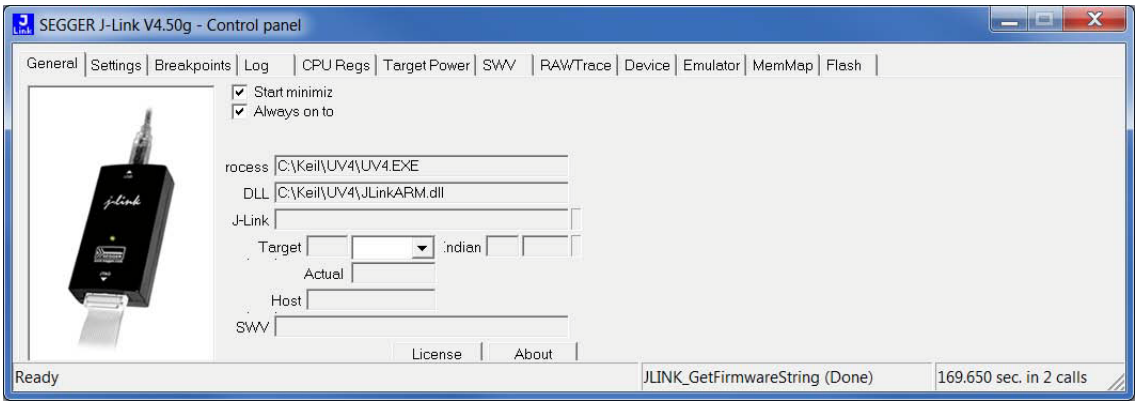


***Figure 37** SEGGER control panel*

5. Click the **Debug** tab shown in the figure. Set Port to **SW** and Max Clock to **1MHz**, as shown in ***Figure 38***. Make sure that SN and IDCODE are populated properly and click **OK**.

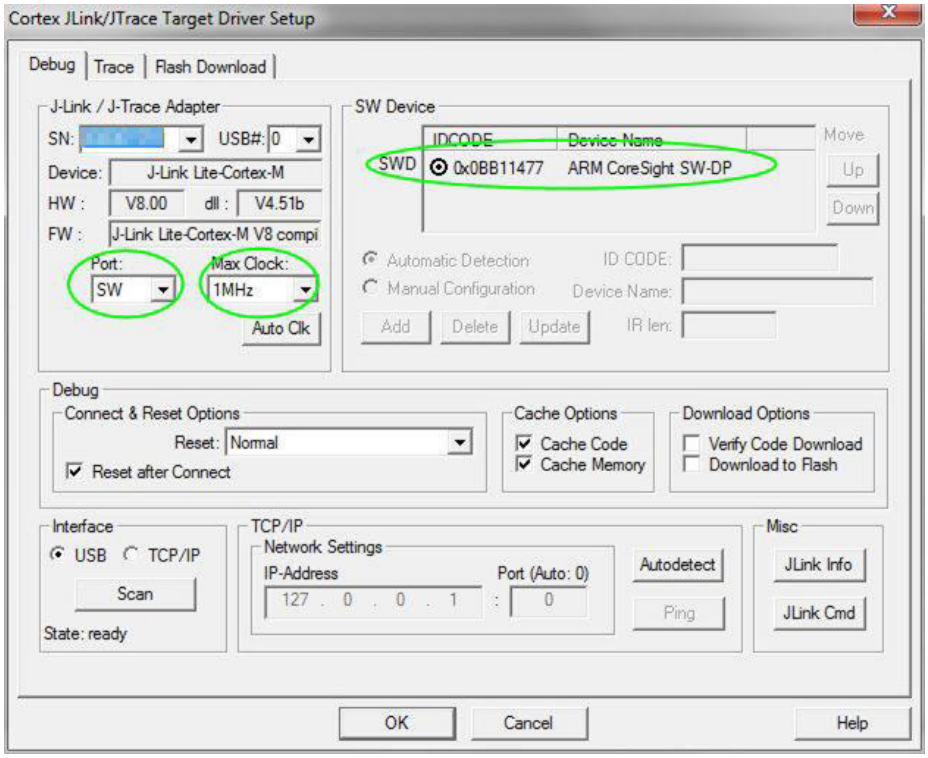


***Figure 38** Debug settings*

6. Select the J-Link device for target programing and provide the appropriate flash algorithm.

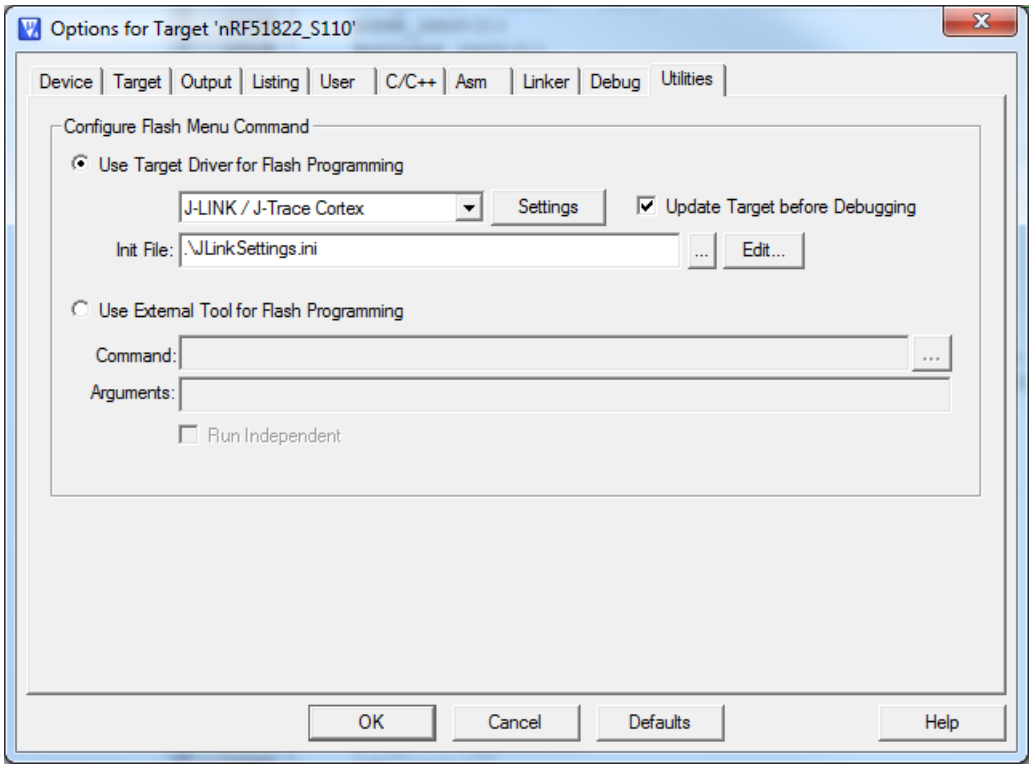


Figure 39 *Flash settings*

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

To improve the debug experience 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.

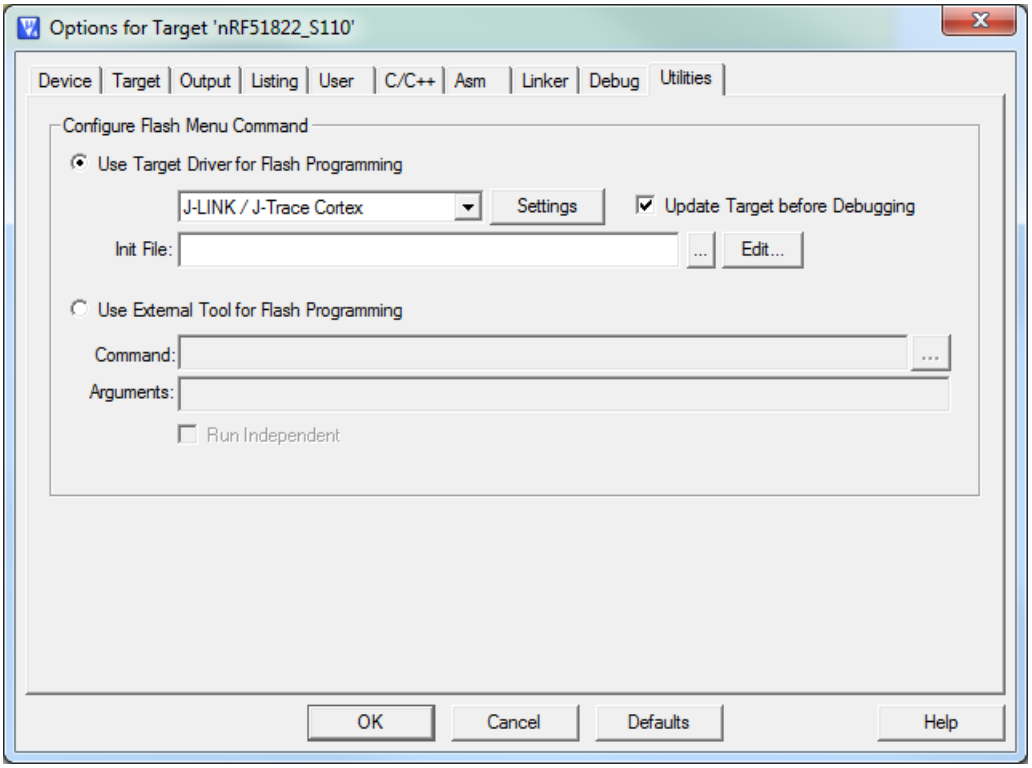


Figure 40 *Locating JLinkSettings.ini*

2. Click **Edit** and set the JLinkSettings.ini settings as specified in JLinkSettings.ini table.

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
September 2012	1.0	

ARM statement

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