nRF51822 Evaluation Kit

nRF51822

User Guide v1.2

Contents

1 Introduction

The nRF51822 *Bluetooth*® low energy/2.4 GHz proprietary Evaluation Kit (EK) 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 v4.56 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
August 2013	2.1	Upgraded the nRF51822 chip to build code FA/F0. Removed R1 on PCA10000 and PCA10001.
February 2013	2.0	Fixed known issue in kit v1.0. **New features on PCA10000 v2.0:** • Multicolor LED • Updated UART configuration.
September2012	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 Kit content

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

2.1 nRF51822 Evaluation Kit hardware content

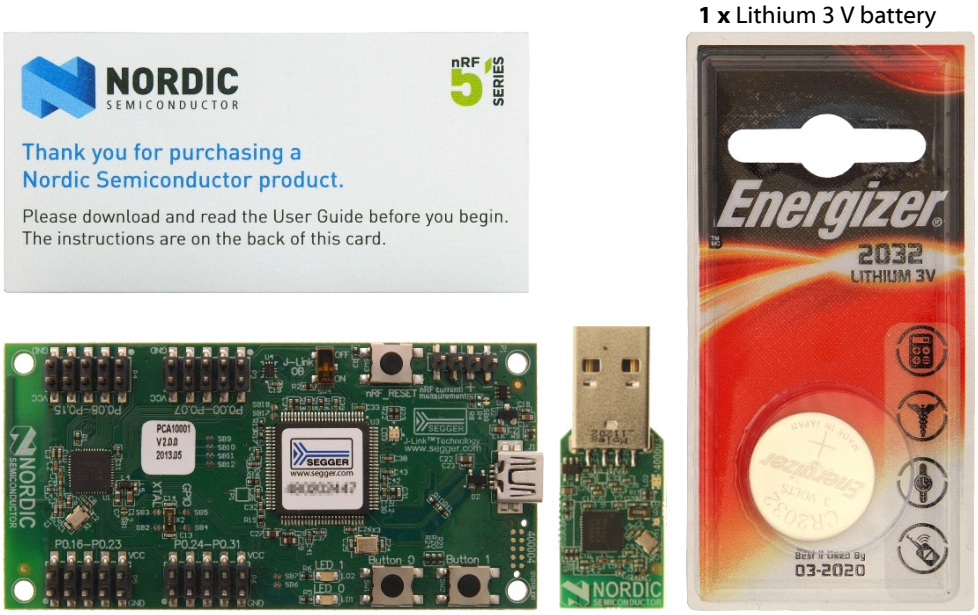


Figure 1 *nRF51822 Evaluation Kit hardware content*

2.2 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. Instructions can be found in ***Chapter 3 "Quick start"*** on page 6.

2.2.1 nRF51822 software

- nRFgo Studio
- nRF51 Software Development Kit (SDK)
  - Precompiled HEX files
  - Source code
  - Keil ARM project files
- S110 nRF51822 SoftDevice
- S110 SoftDevice programming tools
- Master Control Panel

2.2.2 nRF51822 documentation

- *nRF51822 Evaluation Kit User Guide*
- *nRF51 Series Reference Manual*
- *nRF51822 PS*
- *S110 nRF51822 SoftDevice Specification*
- *nRF51822 PAN*

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.

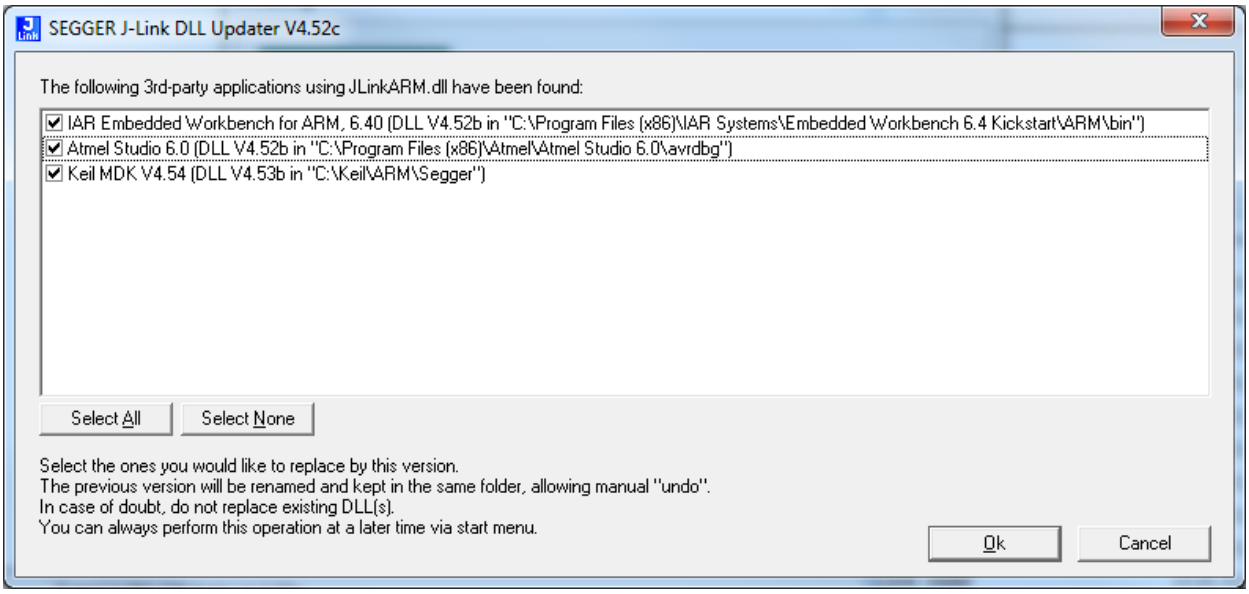
- Altium Designer files
- Schematics
- PCB layout files
- Production files
  - Assembly drawings
  - Drill files
  - Gerber files
  - Pick and Place files
  - Bill of Materials

3 Quick start

This section shows you how to set up the nRF51822 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.

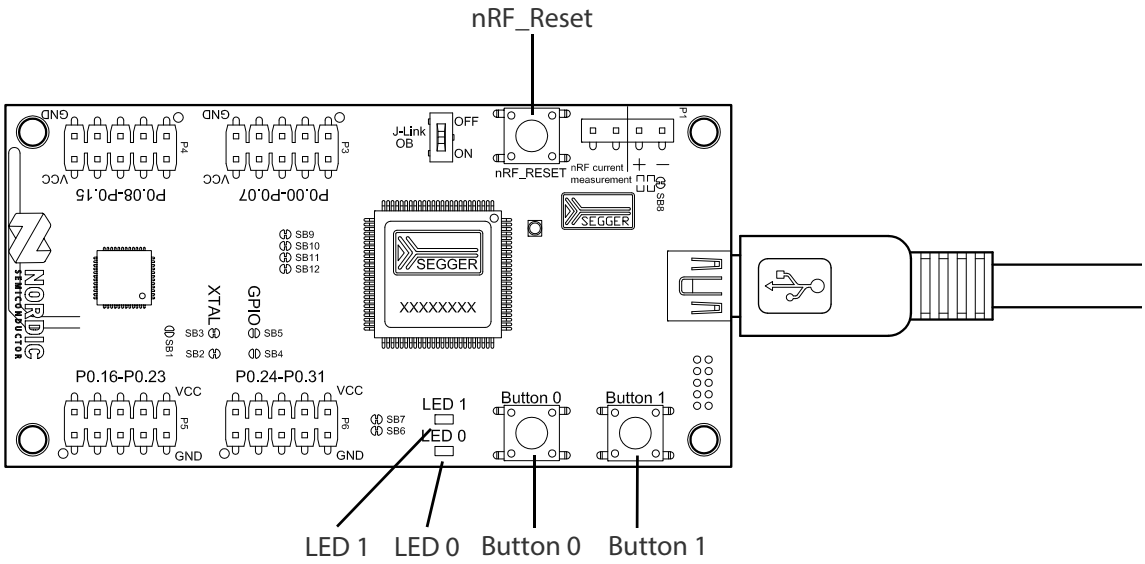


4. If you are using Keil v4.54, go to: http://www.segger.com/IDE_Integration_Keil.html#knownproblems. Download JL2CM3 and copy it to **<keil>/ARM/Segger**. This patch is necessary for the SEGGER debugger to work.
5. Go to www.nordicsemi.com and log in to your Nordic My Page account.
6. Select **MY KEYS** from the left menu. This takes you to the My Keys page.
7. Enter the product key (found on the label on the plastic package) into the **Product Key** field and click **Add**.
8. Click **MY PRODUCTS** in the left menu.
9. From the **Add product** dropdown, select the product name and click **Add**. The chosen product is now listed in the below **Overview, My Products** table.
10. In the **Overview, My Products** table click the **Downloads** link next to the product name to go directly to the relevant product page download section.
11. Download and run the nRF51 SDK installer. Make sure to choose the **Keil MDK-ARM** installer option.

3.1 Install the nRF51822 Evaluation board (PCA10001)

Connect the hardware

1. 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 the **Load** icon to download and run the Blinky example firmware. **LED 0** and **LED 1** on the PCA10001 should now blink sequentially.

3.2 *Bluetooth* low energy heart rate monitor demo

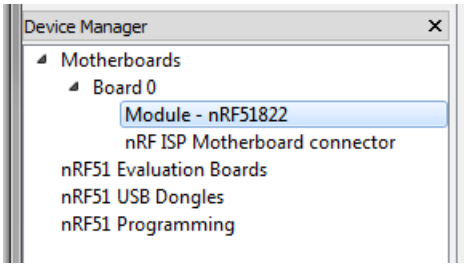
This section shows you how to program a heart rate monitor demo on top of a SoftDevice that will send data on a *Bluetooth* link from the heart rate monitor to the Master Control Panel, giving you a simple application to get your device up and running.

Download and program the SoftDevice

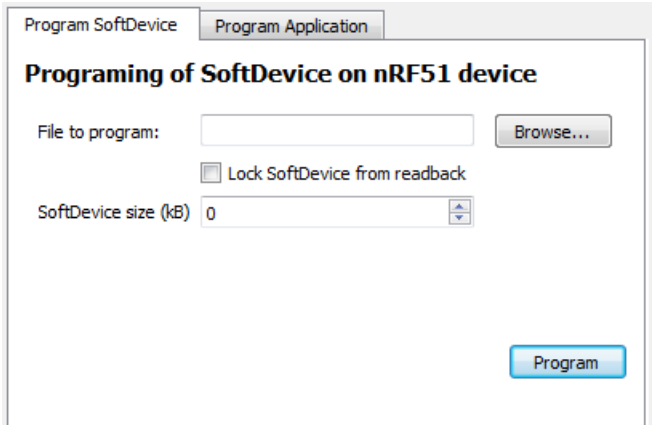
Type the product key (included with the Evaluation Kit) into the **Product Key** field in My Page to download the S110 nRF51822 SoftDevice.

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 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 Master Control Panel and the nRF51822 Development Dongle (PCA10000)

1. Download and install the Master Control Panel from our website. Make sure to select the version that corresponds to your hardware.
2. Plug the Development Dongle into a USB port on your computer.
3. 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 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 Development Dongle.
3. When you use the 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 **mef-w_nrf51822_<version>_firmware.hex** (<version> will be replaced by a number giving the version of the actual firmware).
    The Master Control Panel Firmware file is located in:
    <keil path>**\ARM\Device\Nordic\nrf51822\board\pca10000\ble\master_emulator**
    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** to go back to the main window.
4. Ensure that the Heart Rate Demo running on PCA10001 is advertising, indicated by **LED 0** blinking.
5. Click **Start discovery**. The Development Dongle will scan for available *Bluetooth* low energy devices within range and list them.
6. Select the device **CompleteLocalName 'Nordic_HRM'** in the Discovered Devices list.
7. Click **Select device**.
8. Click **Service Discovery**. The Development Dongle 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.

4 Evaluation kit configuration

This chapter includes downloadable third party content and information about 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. Select option "Install USB Driver for J-Link-OB with CDC" when installing. See ***Appendix A*** on page 45.

4.1.1 Development environment setup

The nRF51822 device can be programmed from several environments. This section shows the development setup 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 on page 12 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.

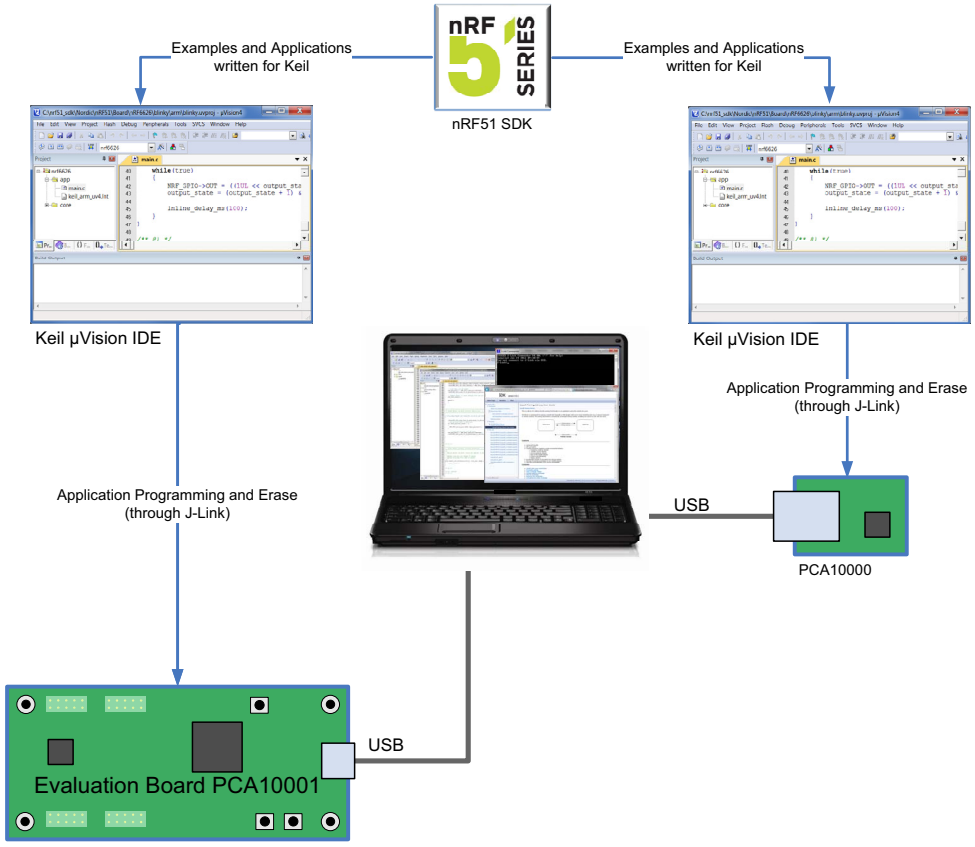


Figure 2 *nRF51822 Evaluation Kit configuration for 2.4 GHz based development*

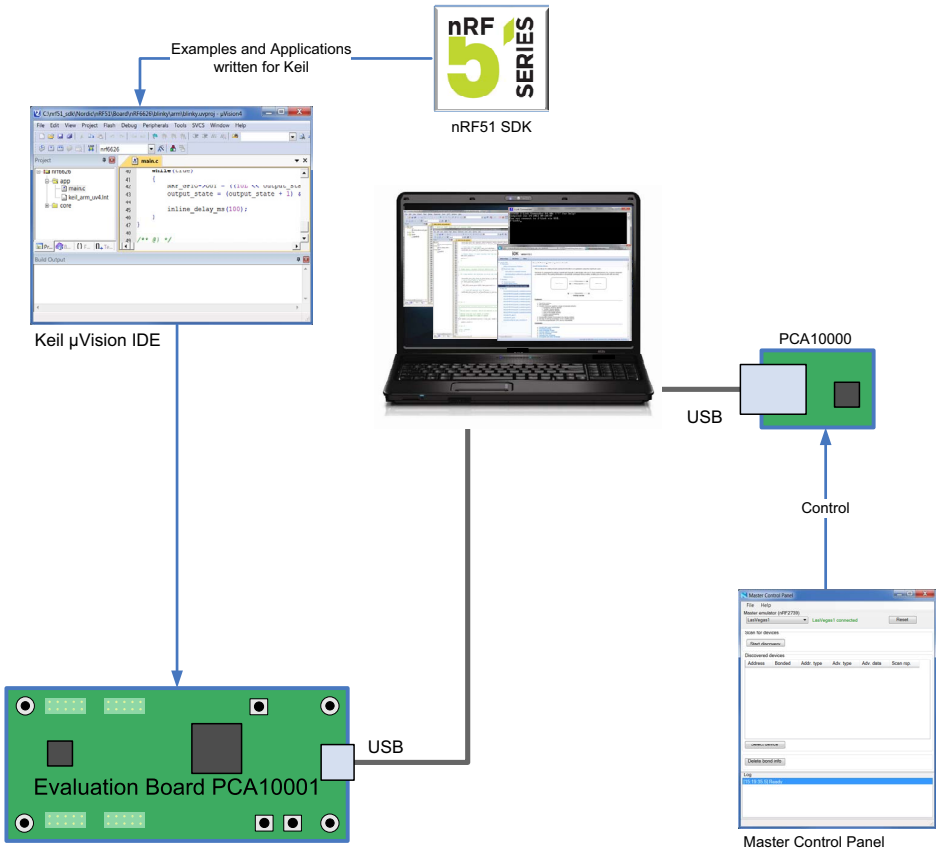


Figure 3 *nRF51822 Evaluation Kit configuration for Bluetooth 4.0 single-mode*

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 a standalone nRF51822 evaluation board with an integrated debugger from SEGGER. The board is delivered with an unprogrammed nRF51822 chip.

5.1.1 Key features

The nRF51822 EK board (PCA10001) has the following key features:

- nRF51822 flash based SoC solution
- *Bluetooth* low energy compatible
- 2.4 GHz compatible with nRF24L devices
- 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

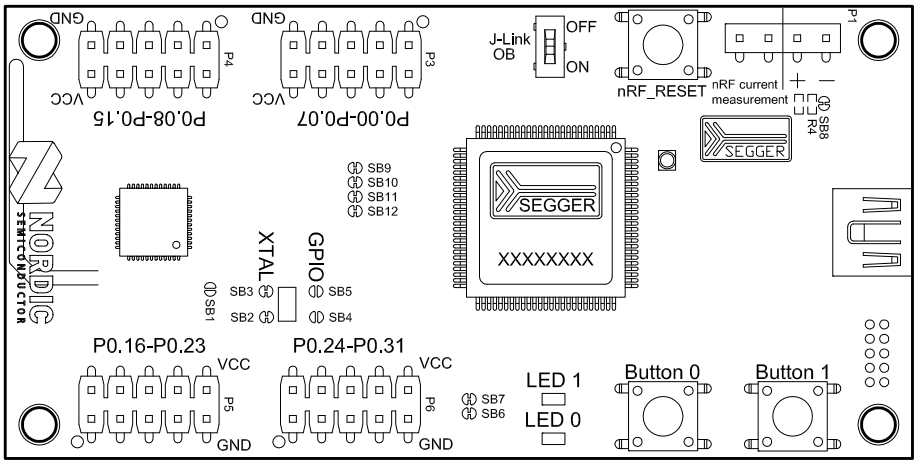


Figure 4 *PCA10001 top*

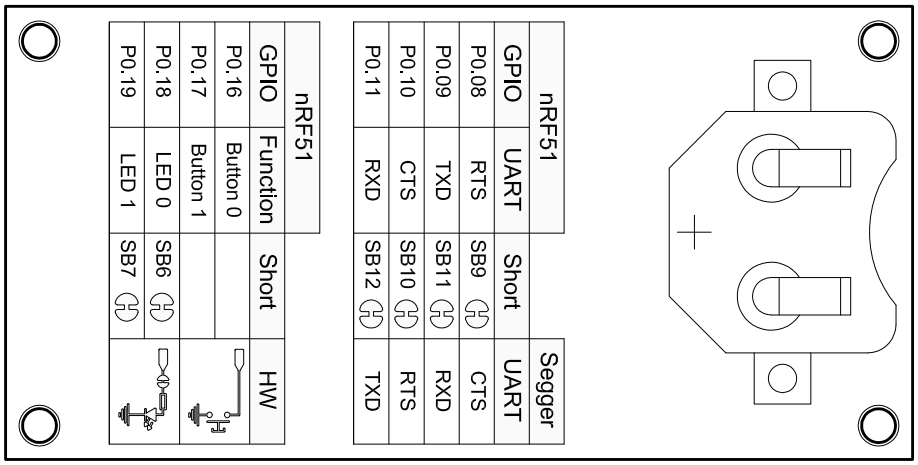


Figure 5 *PCA10001 bottom*

5.1.3 Block diagram

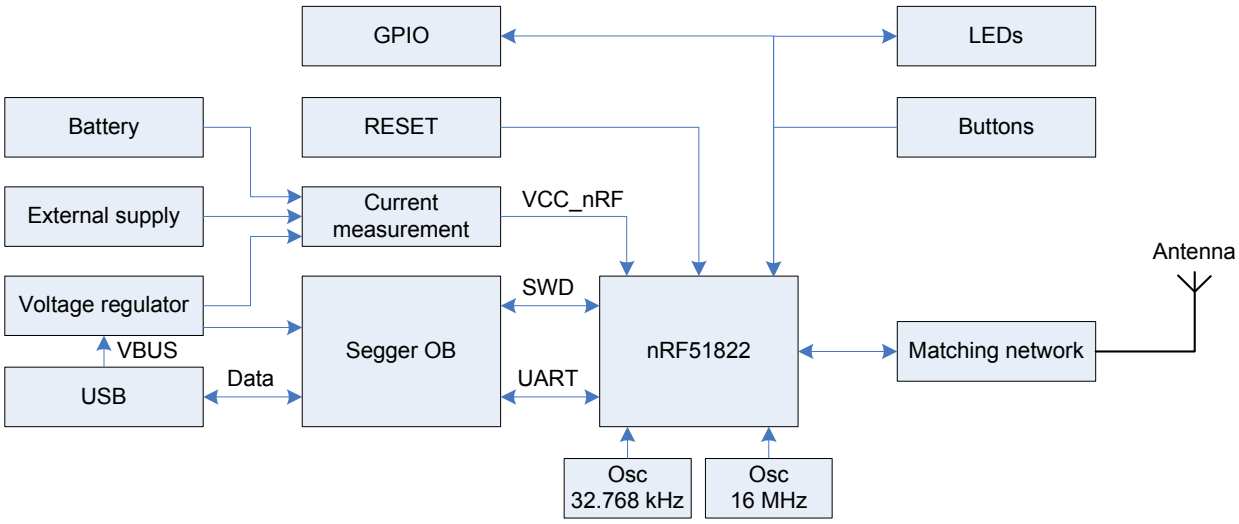


Figure 6 *PCA10001 block diagram*

5.1.4 Reset button

The EK board (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

The EK board (PCA10001) has several power options:

- USB (see ***Figure 7***)
- External power supply through **P1** (1.8 V to 3.6 V)
- CR2032 coin cell battery (see ***Figure 8***)

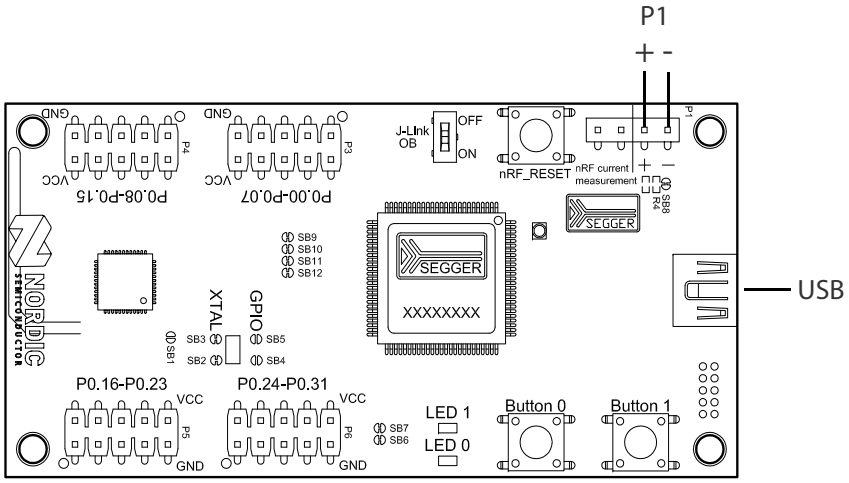


Figure 7 *USB and external power supply*

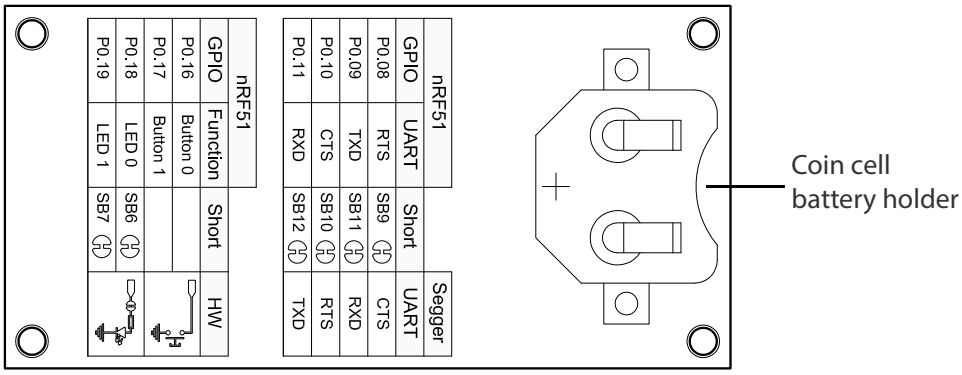


Figure 8 *Coin cell battery supply*

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

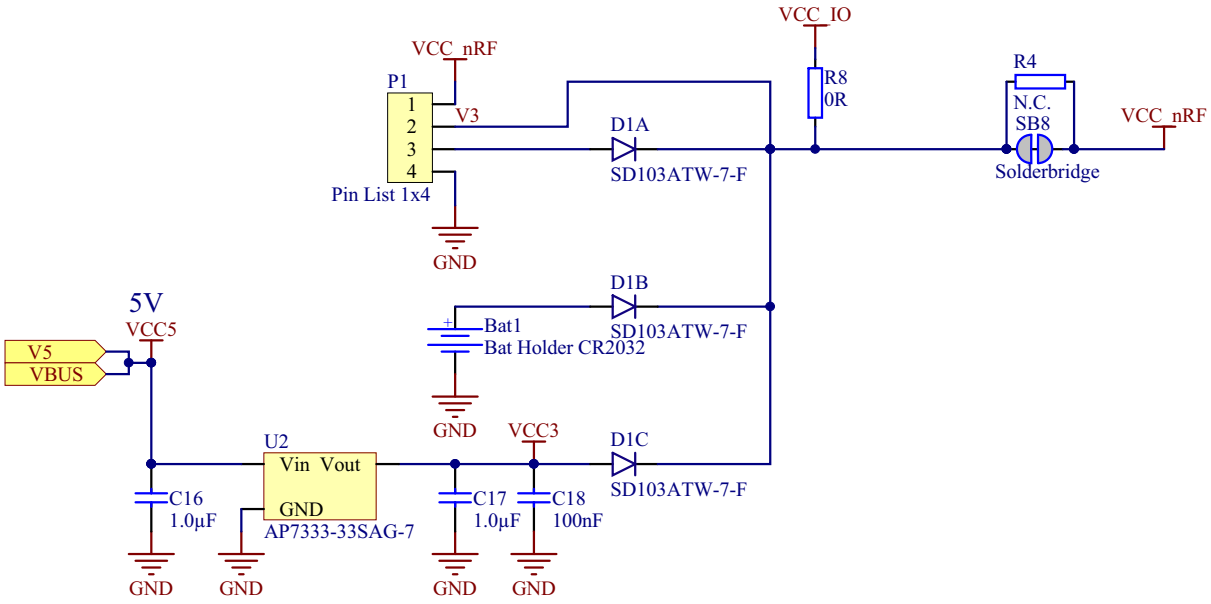


Figure 9 *Power supply circuitry*

5.1.6 SEGGER SWD disconnect

The SEGGER J-Link OB circuit on the EK board (PCA10001) 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 10***.

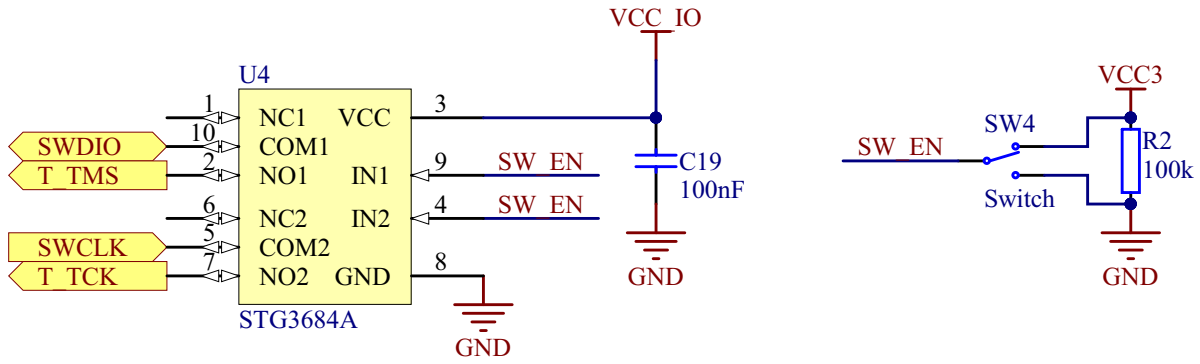


Figure 10 *SWD disconnect logic*

Note: To program or debug the nRF51822 using the SEGGER J-Link OB, the **SW4** must be switched to **ON**.

5.1.7 GPIO interface

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

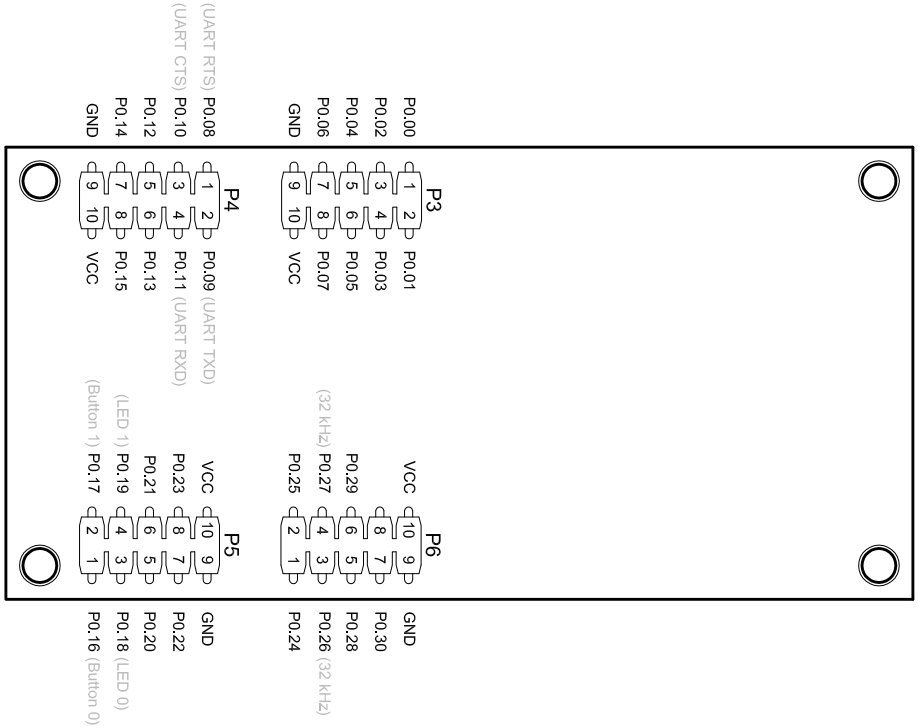


Figure 11 *PCA10001 GPIO pin headers*

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 "32.768 kHz crystal"*** on page 20 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 "Buttons and LEDs"*** on page 19 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 "UART configuration"*** on page 20 for more information.

5.1.8 Buttons and LEDs

The two buttons and two LEDs on EK board PCA10001 are connected to dedicated I/Os on the nRF51822chip. 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 12***.

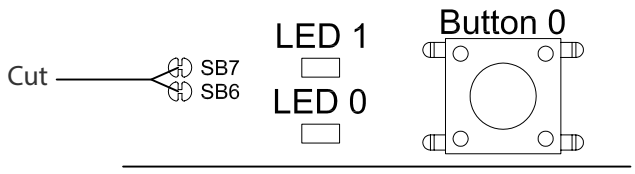


Figure 12 *Disconnecting the LEDs*

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.

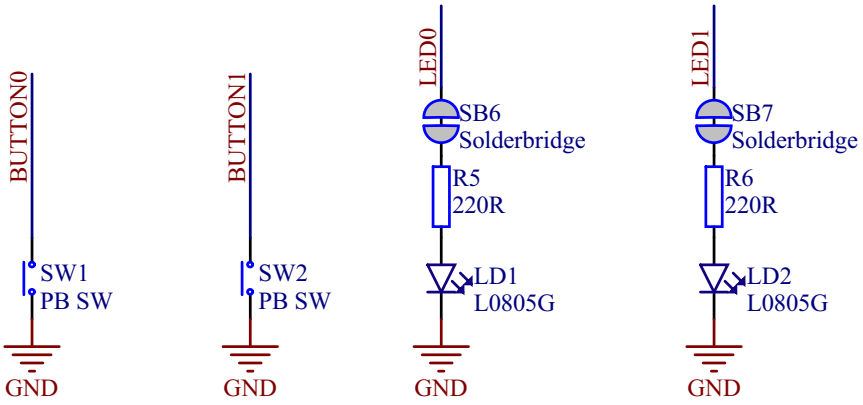


Figure 13 *Button and LED configuration*

5.1.9 32.768 kHz crystal

nRF51822 can use an optional 32.768 kHz crystal (X2) for higher accuracy and lower average power consumption. On the EK board PCA10001 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 14*** on page 20 for reference.

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

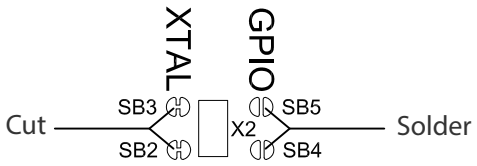


Figure 14 *Disconnecting 32.768 kHz crystal and connecting P0.26 and P0.27 to P6*

5.1.10 UART configuration

Table 2 shows an overview of the UART connections on nRF51822 and the SEGGER IC.

nRF51822			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 nRF51822 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 15***.

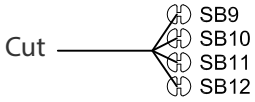


Figure 15 *Disconnecting UART lines*

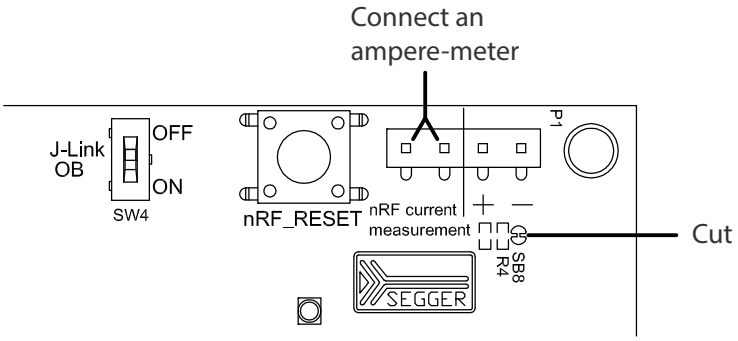
Note: In order to use the USB to UART bridge, the software on the nRF51822 must enable flow control. For details on how to set up the UART with flow control see the *nRF51 Series Reference Manual*.

5.1.11 Measuring current

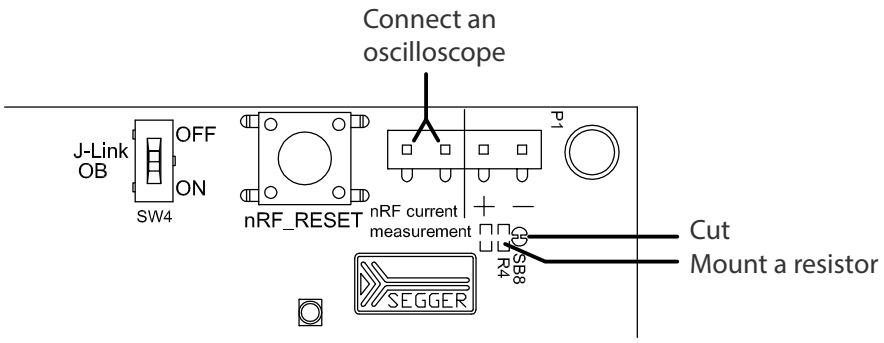
The current drawn by the nRF51822 device can be monitored on the PCA10001. To measure the current, you must first prepare the board by cutting the shorting of solder bridge **SB8**.

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.



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.



Note: When measuring ultra low power (uA level) SW4 should be switch off to have accurate current measurement.

5.2 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. In addition to radio communication, the nRF51822 device can communicate with a computer through a virtual COM port provided by the SEGGER chip. The PCA10000 can be loaded with 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.

Note: PCA10000 can be reprogrammed if overwritten.

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

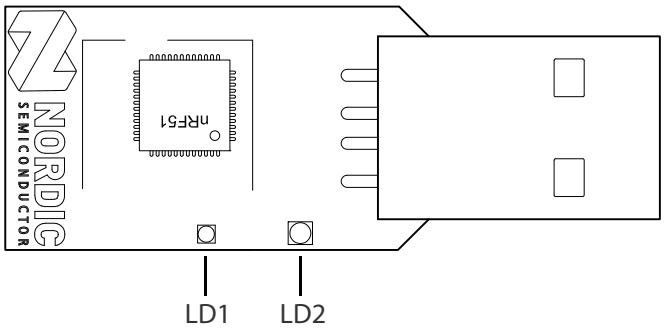


Figure 16 *PCA10000 top side*

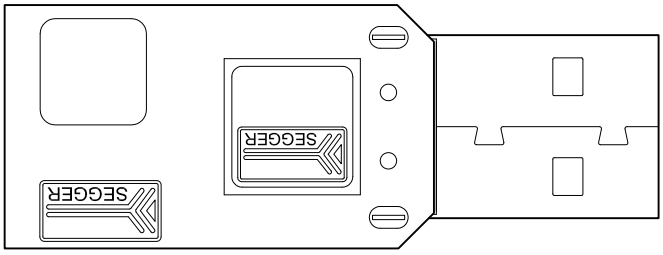


Figure 17 *PCA10000 bottom side*

5.2.3 Block diagram

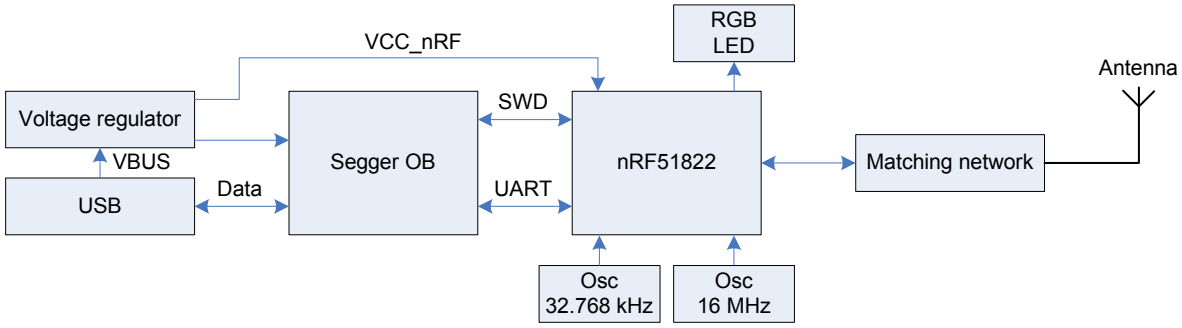


Figure 18 *PCA10000 block diagram*

5.2.4 Multicolor LED

The Development Dongle (PCA10000) is equipped with a multicolor RGB LED. The LED is connected to dedicated I/Os on the nRF51822 chip. The connections are shown in ***Table 3***.

Color	GPIO
Red	P0.21
Green	P0.22
Blue	P0.23

Table 3 *LED connection*

The LEDs are active low, meaning that writing a logical zero '0' to the output pin will illuminate the LED.

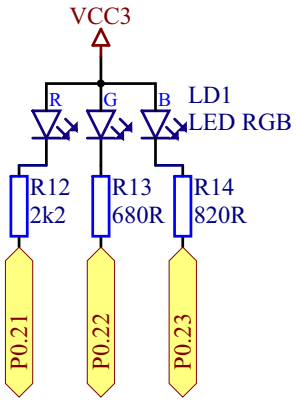


Figure 19 *LED configuration*

5.2.5 UART configuration

The Development Dongle v1.0 UART lines are connected to pins P0.00 to P0.03 as shown in ***Table 4***.

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

Table 4 *Development Dongle v1.0 UART configuration*

The Development Dongle v2.0 UART lines are connected to pins P0.08 to P0.11 as shown in ***Table 5***.

nRF51822		SEGGER IC
GPIO	**UART**	**UART**
P0.08	RTS	CTS
P0.09	TXD	RXD
P0.10	CTS	RTS
P0.11	RXD	TXD

Table 5 *Development Dongle v2.0 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

The nRF51822 chip 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 nRF51822 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 and erase the S110 nRF51822 SoftDevice or another application HEX file on the nRF51822 chip.

If you want to start developing on the nRF51822 chip without using the S110 SoftDevice see ***Section 6.1.5 "Programming an application"*** on page 28.

6.1 Programming and erasing flash using nRFgo Studio

Use nRFgo Studio to program or erase a SoftDevice or application HEX file onto the nRF51822 chip.

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

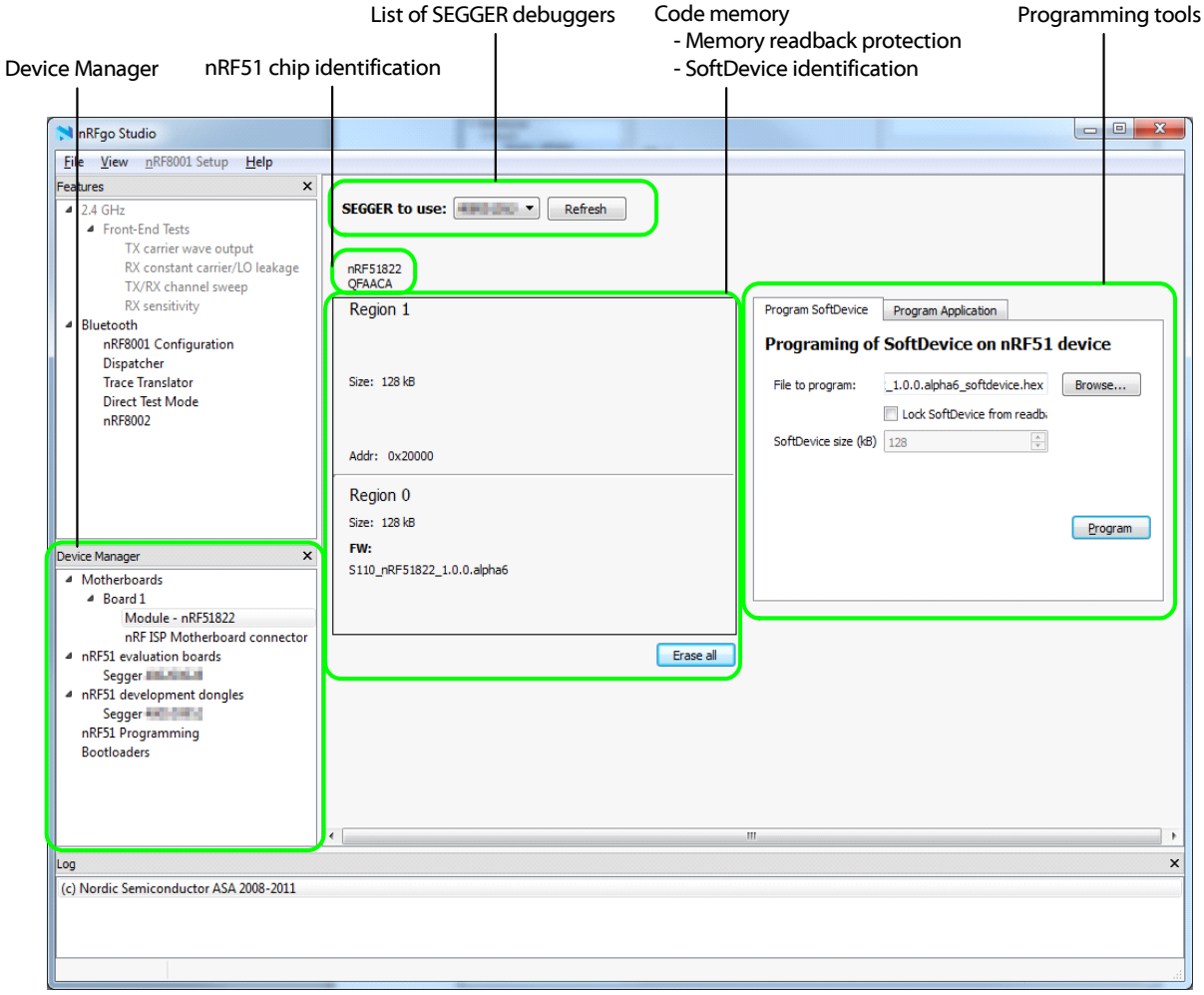


Figure 20 *nRFgo Studio dashboard*

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 nRF51822 Evaluation Board (PCA10001) and Development Dongle (PCA10000) hardware will show up under the respective sections **nRF51 evaluation boards** and **nRF51 development dongles**. The SEGGER serial number identifies the devices.
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 nRF51822 chip and chip content

When you select a board, nRFgo Studio identifies the nRF51822 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, 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. 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. 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. If there is only one region the option is readback protection on (All) or off.
  - 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

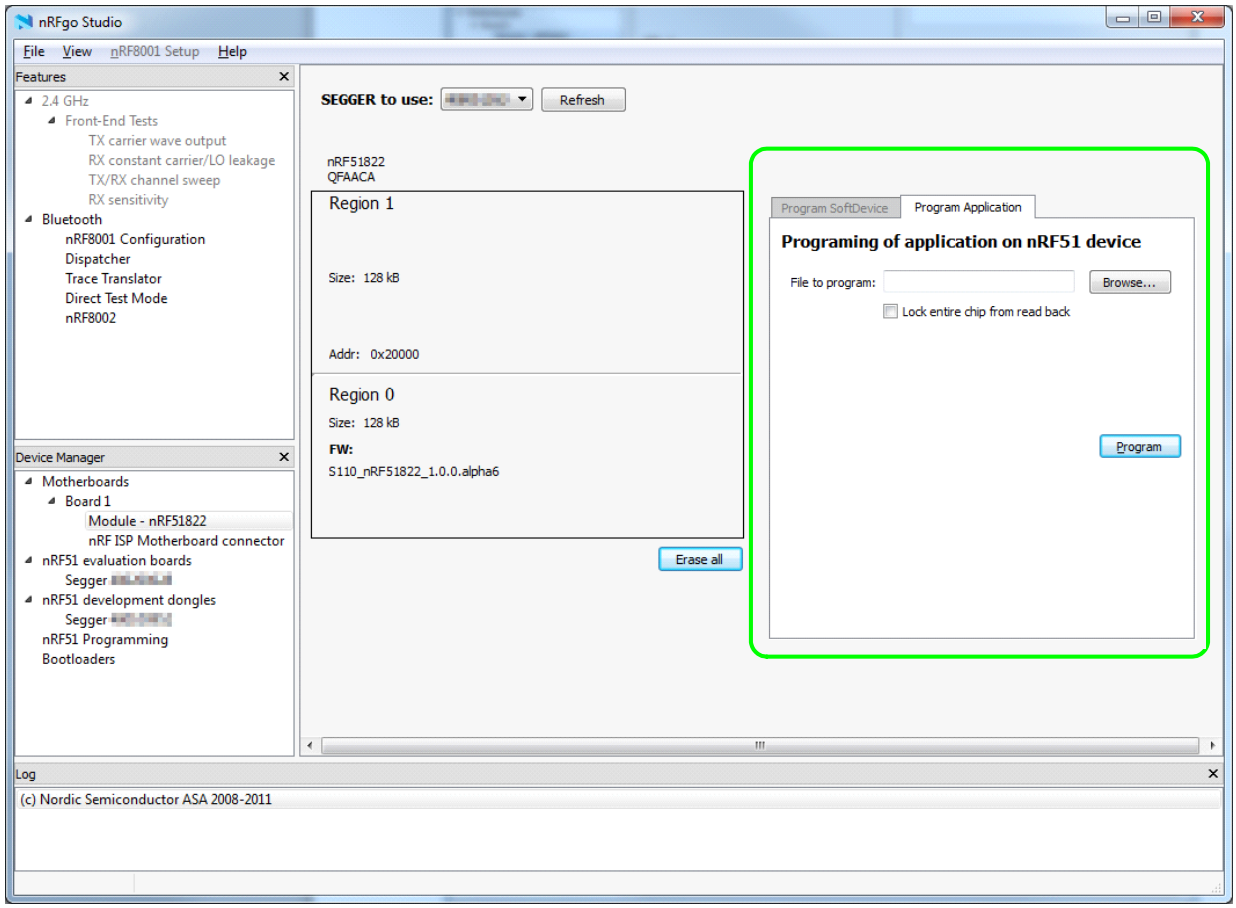
You should use the Erase all function in the following situations:

- 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 a clean chip using nRFgo Studio with the option "Lock entire chip from readback".

To use the Erase All function, follow the steps in ***Section 6.1.1 "Selecting a board to program"*** on page 26. Then click **Erase all**.

6.1.4 Programming a SoftDevice

This function lets you program the SoftDevice onto the chip.



1. Follow the steps in ***Section 6.1.1 "Selecting a board to program"*** on page 26 and then select the **Program SoftDevice** tab.
2. Click **Browse** and select the HEX file to program.
3. Select whether to enable or disable readback 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.

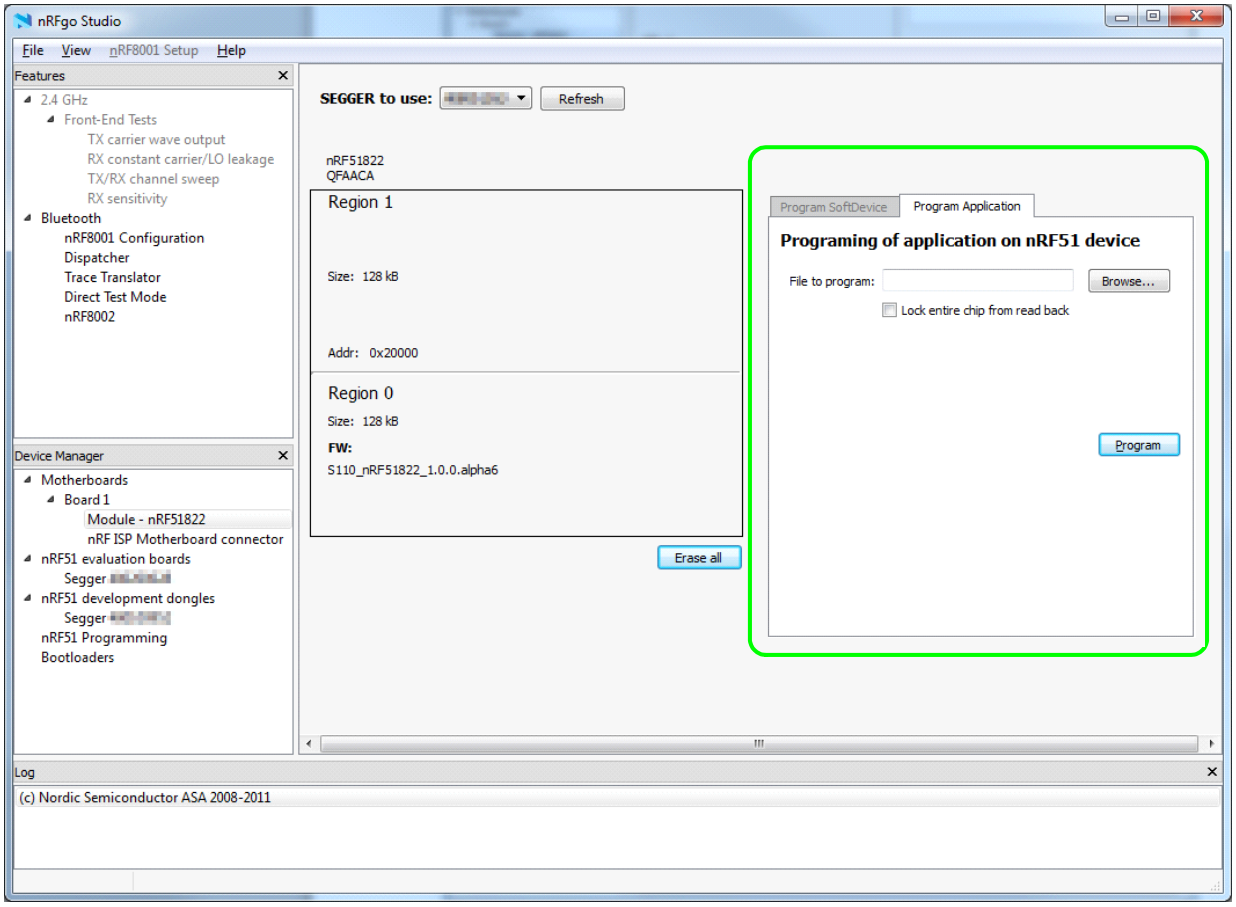
Note: The S110 nRF51822 SoftDevice can be downloaded from www.nordicsemi.com by logging into your MyPage account and entering the product key printed in the Development Kit.

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. If it matches, nRFgo Studio continues with the programming, if not it stops the programming and returns an error message. For example, if an application requires the SoftDevice on the chip, it will check the memory configuration for the SoftDevice before programming the chip.

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



1. Follow the steps in ***Section 6.1.1 "Selecting a board to program"*** on page 26 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 6 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 6 *SoftDevice memory layout*

Note: See the *nRF51822 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 *S110 nRF51822 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**.

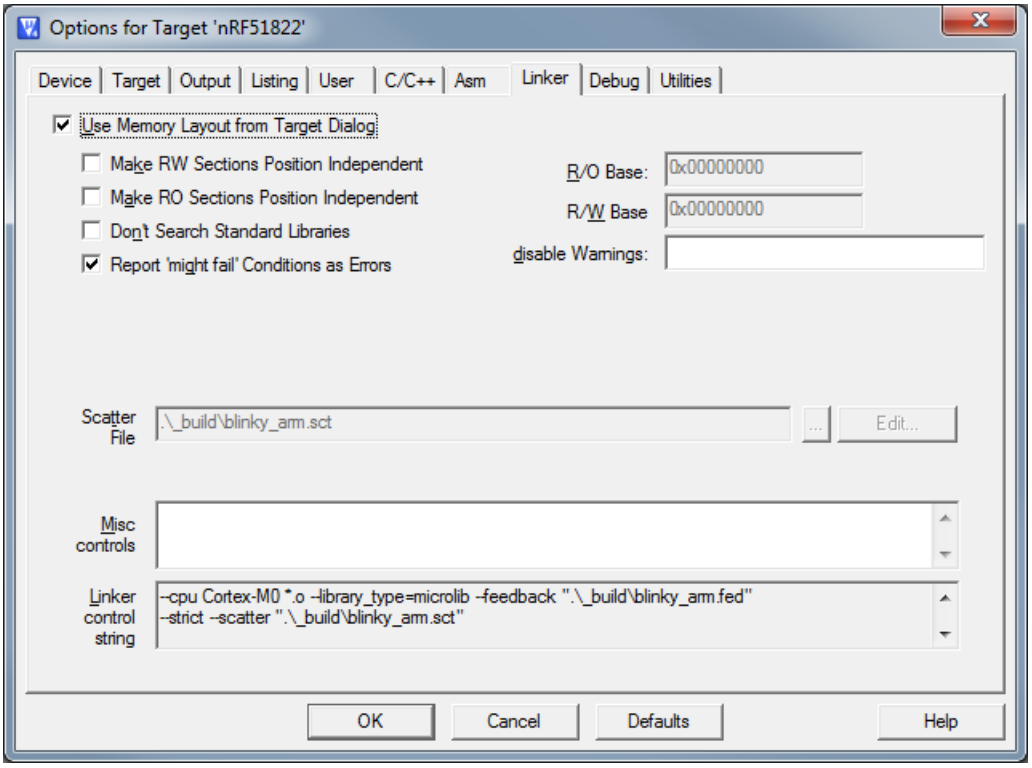


Figure 21 *Keil linker settings*

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 22*** on page 31.
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 6*** on page 29).

- 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).

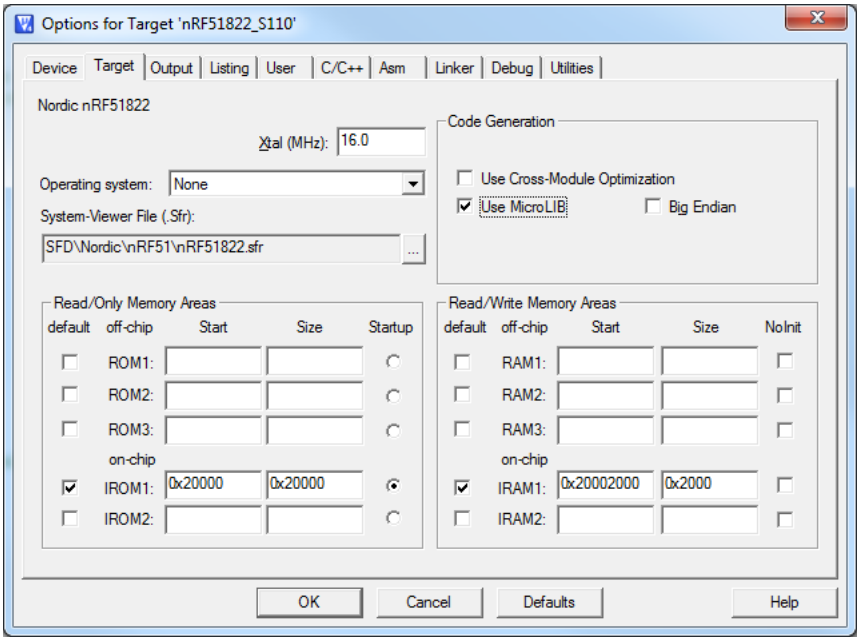


Figure 22 *Memory layout with example SoftDevice*

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 7 *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 *S110 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** (**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

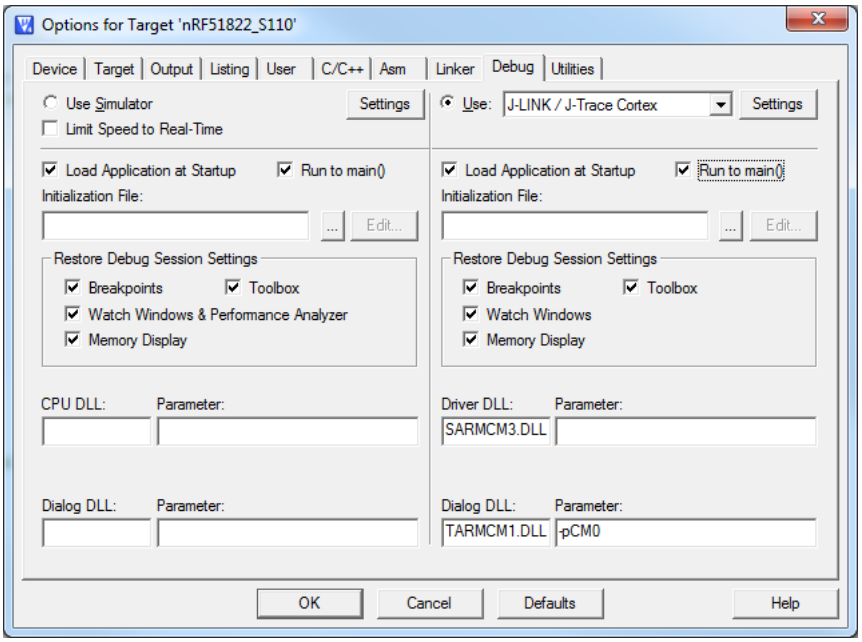


Figure 23 *Debugger options*

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 24*** on page 34.

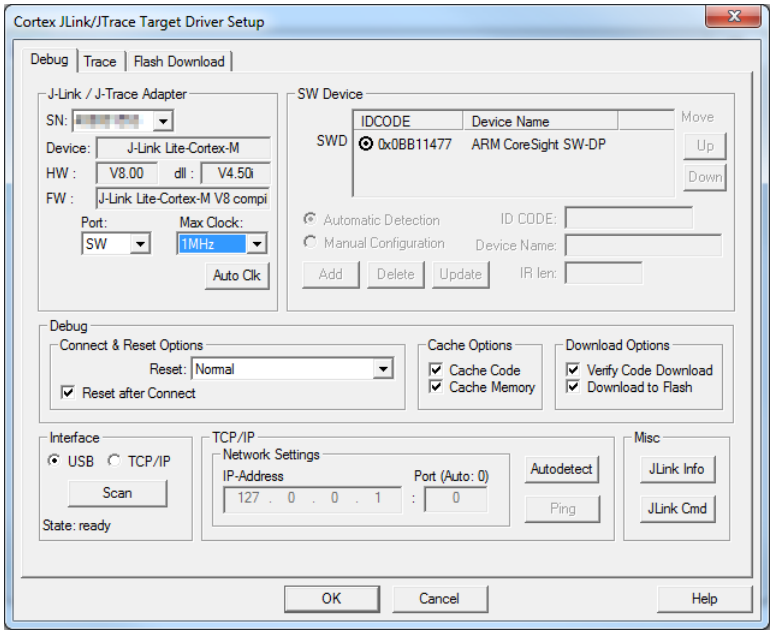


Figure 24 *Target driver setup*

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

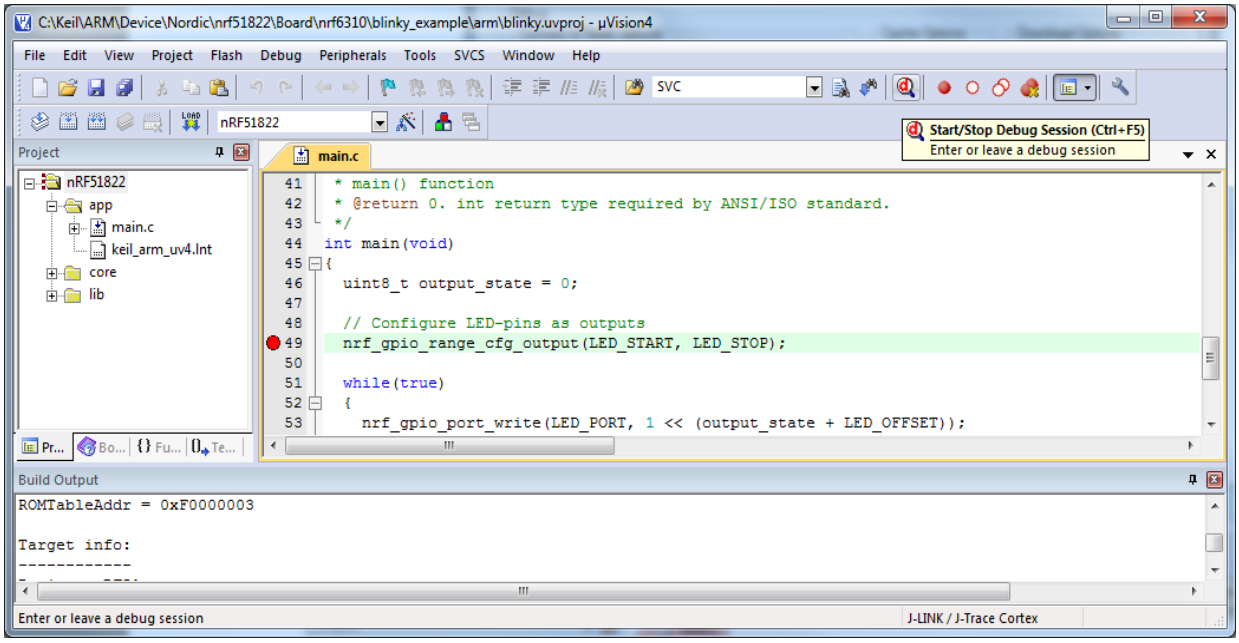


Figure 25 *Debugger initiation*

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. See ***Section 7.1.2*** on page 40 for more information.

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 code memory area occupied by the SoftDevice is write and erase protected. When the SoftDevice is enabled, the Memory Protection Unit implements a write protection to certain peripherals used by the protocol stack. Protected peripherals are described in the *S110 nRF51822 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.

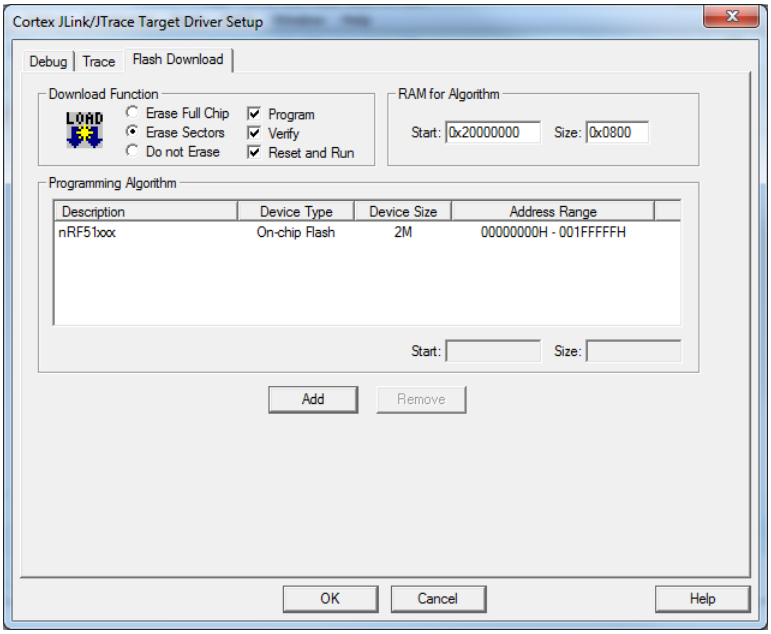


Figure 26 *Selecting J-LINK/J-Trace Cortex*

8. In the **Utilities** tab, select **Use Target Driver for Flash Programming**.
9. Choose the available debugger from the list as shown in ***Figure 27***.

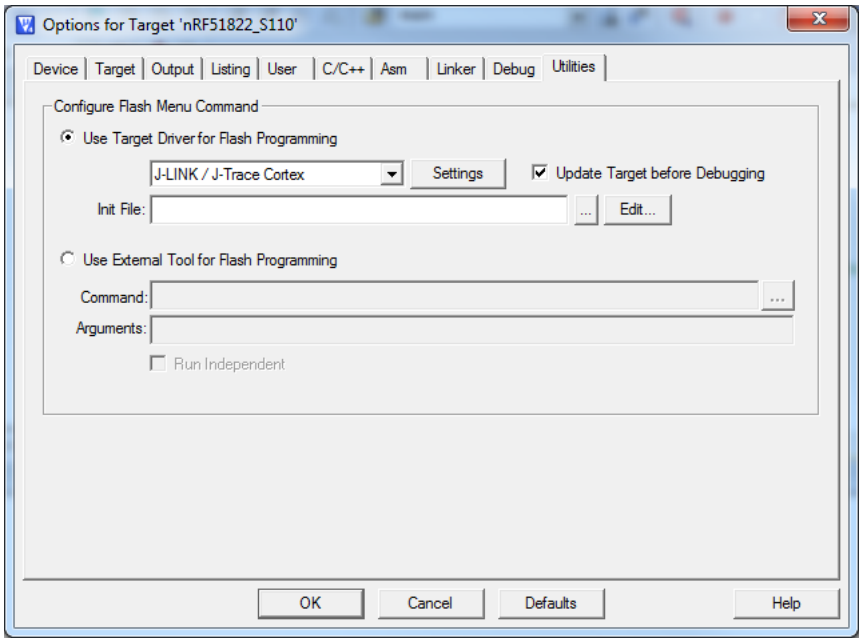


Figure 27 *Debugger selection*

7 Debugging the nRF51822 chip

For debugging with SEGGER J-Link, see ***Appendix A*** on page 45. For general information on how to debug using the Keil µVision 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**.

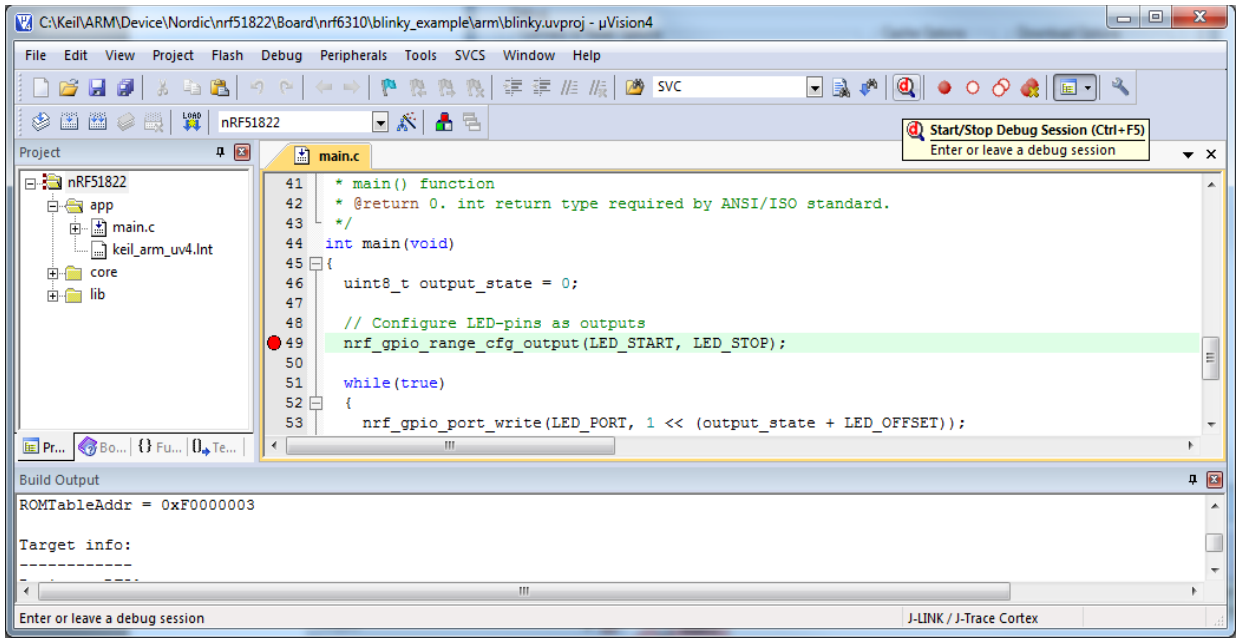


Figure 28 *Start debugging mode*

7.1 nRF51822 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 29***. 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 30*** on page 39. More information on System Viewer can be found at: http://www.keil.com/uvision/db_view_sysview.asp.

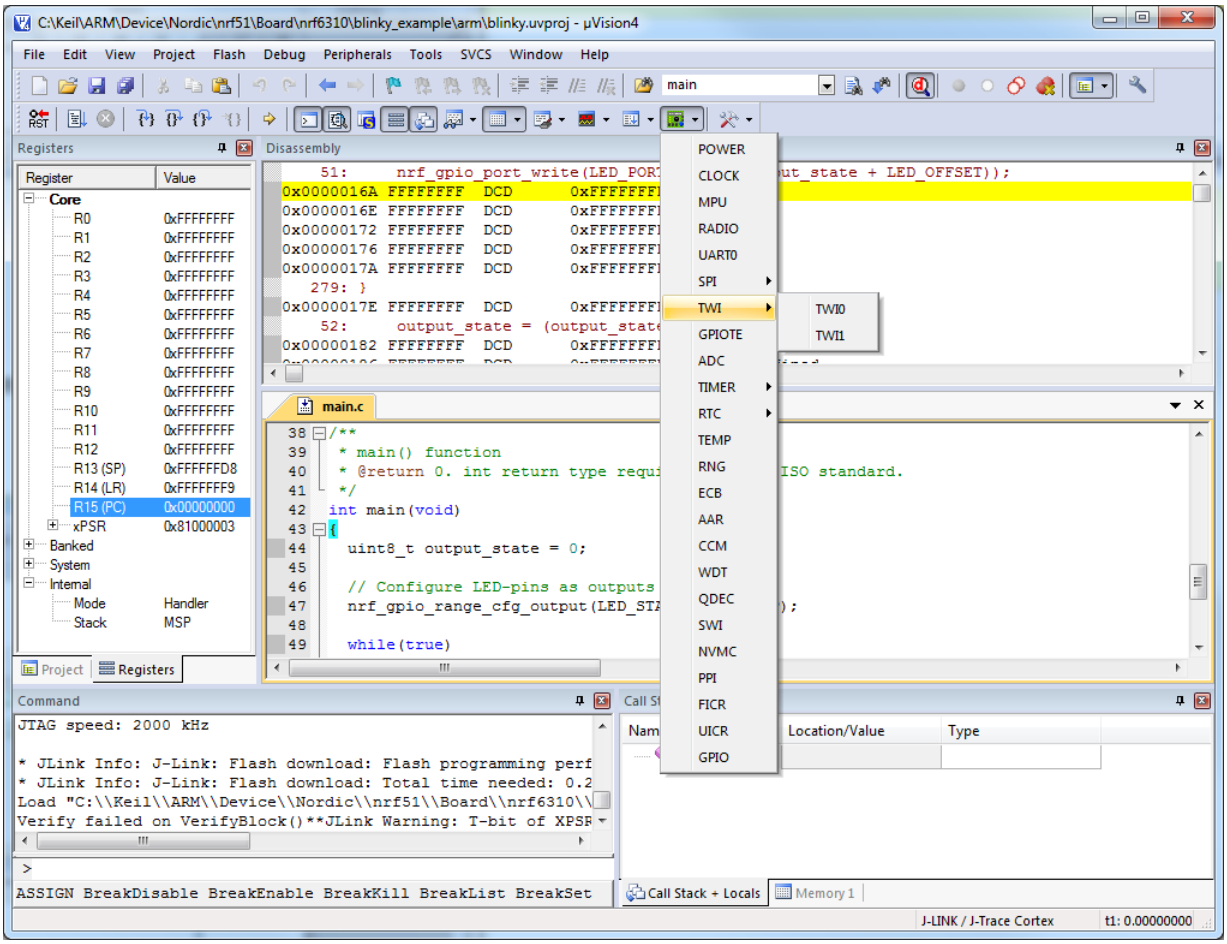


Figure 29 *System Viewer Windows*

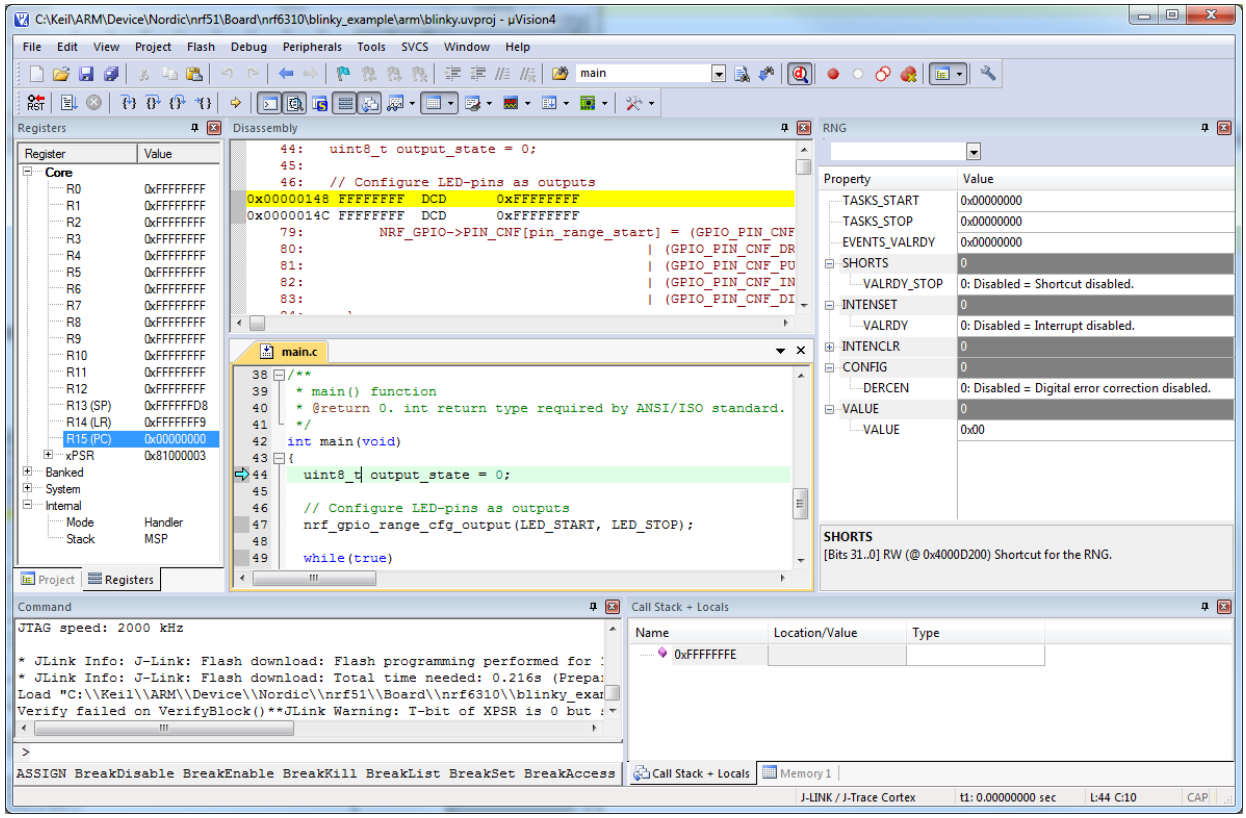


Figure 30 *System viewer window of the RNG peripheral*

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 *S110 nRF51822 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 32*** on page 41. The “step over” function (**F10**) may also be used instead to step over SVC calls to avoid delays when entering the readback protected area.

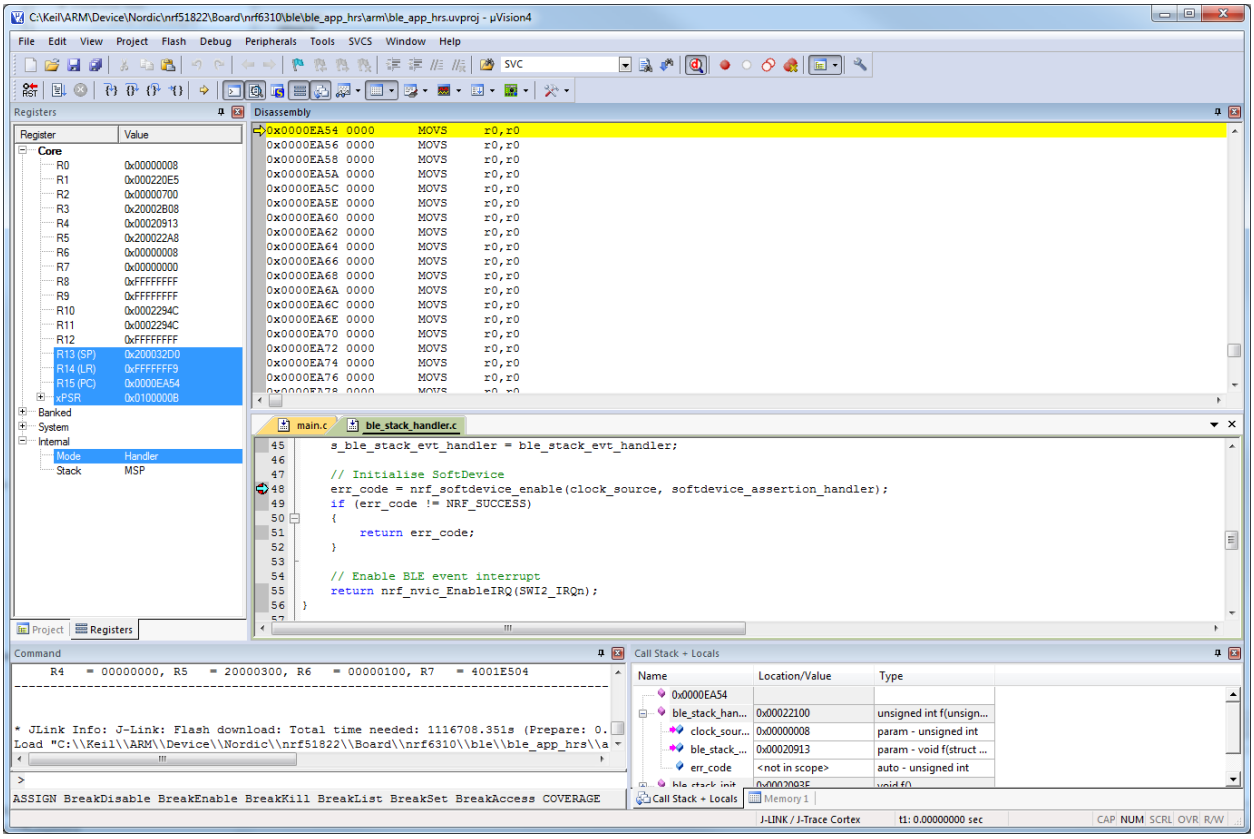


Figure 31 *Debugger information for a setup with a SoftDevice enabled in the protected area*

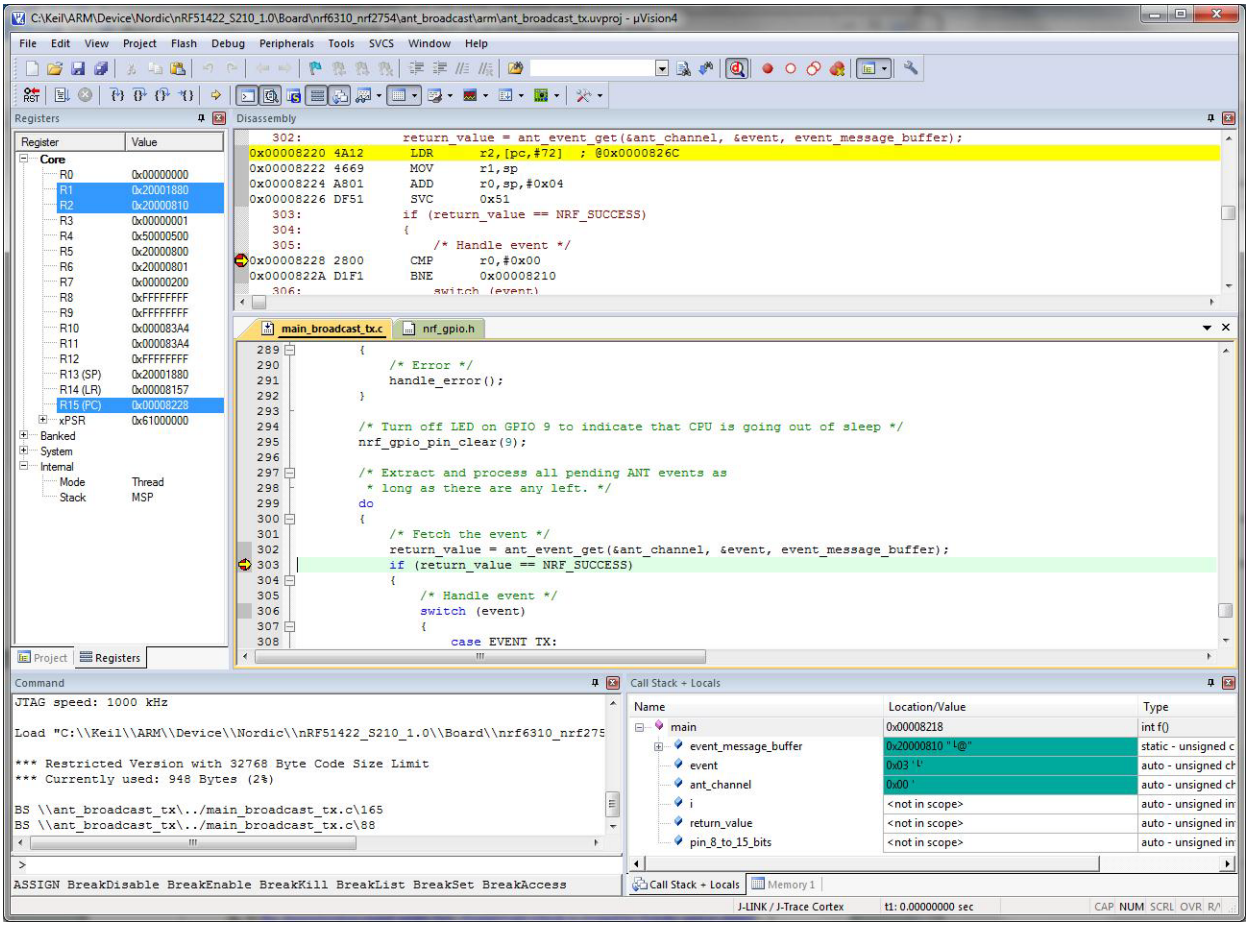


Figure 32 *Setup with a breakpoint after an SVC call*

8 Software Development Kit

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

- *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 nRF51 SDK

The nRF51 SDK is a part of the downloadable content available from your My Page account, see ***Section 2.2 "Downloadable content"*** 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, Master Control Panel, and/or Custom install. The following describes each installation option:

- **Keil MDK support** - installs Keil µVision 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.
- **mergehex** - installs a command line tool to combine two hex files.

nrfjprog and mergehex are installed by default at
C:\Program Files (x86)\Nordic Semiconductor\nrf51\bin.

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 hardware v1.0 of 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 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.

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 "Programming and erasing flash using nRFgo Studio"*** on page 25), 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 "Debugging an application when a readback protected SoftDevice is present"*** on page 40 for more details.

The debugger does not halt on breakpoints.

Some Keil projects in the SDK have **Optimization level 3 (-03)** 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.

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 "Kit content"*** on page 4 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 "Configuring memory layout"*** on page 29 on how to set up memory configuration.

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 nRF51SDK 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 34*** on page 46.
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.

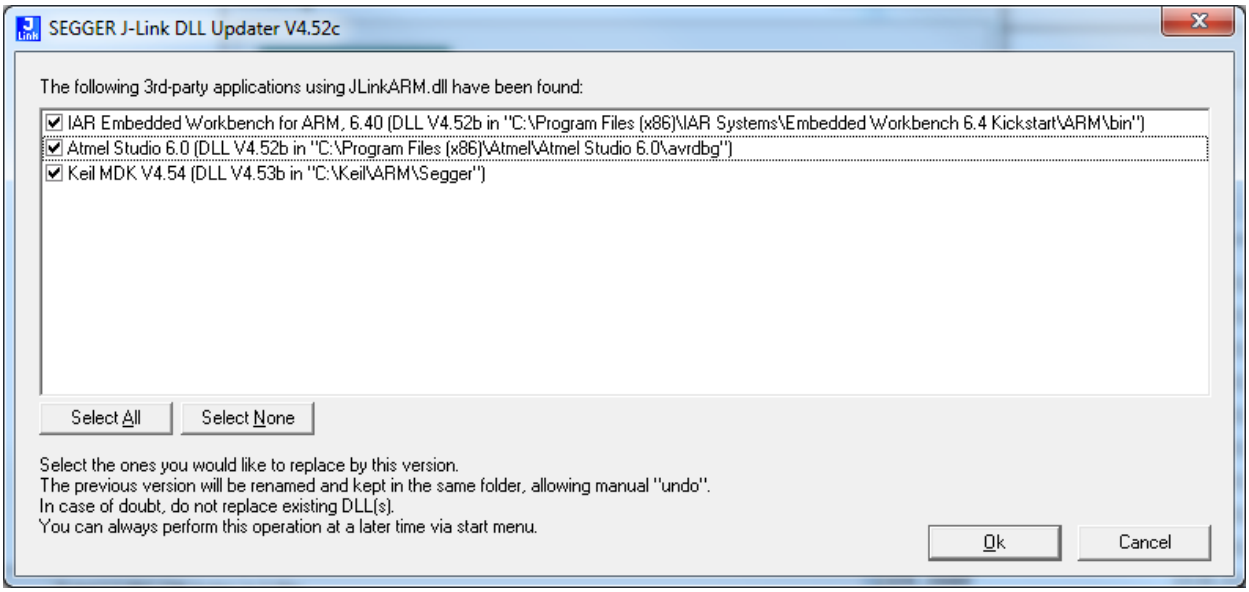


Figure 33 *IDEs selected for updating to the latest SEGGER DLLs*

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 (PCA10001) into your computer with a USB cable. The LD3 LED will blink while the driver installation occurs. Wait until the LED is continuously lit

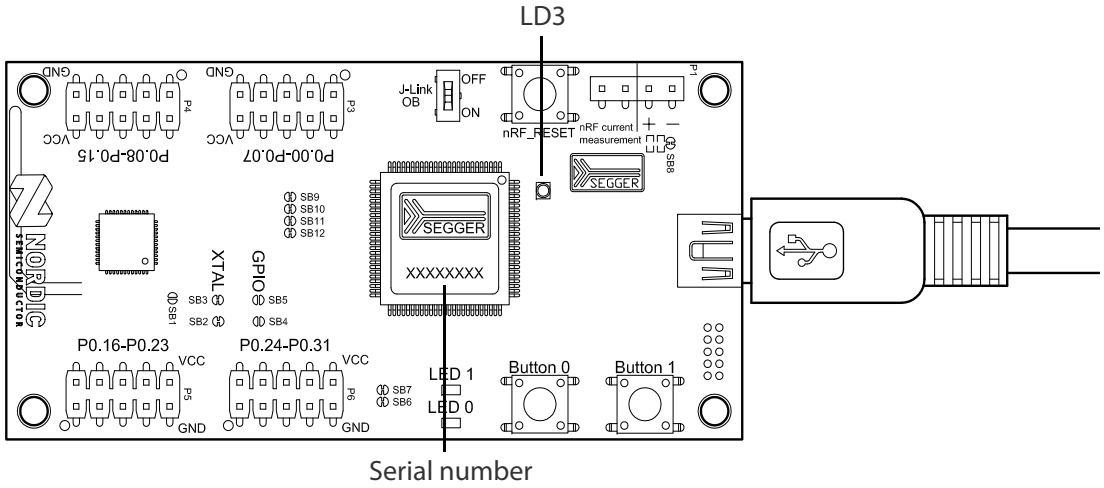


Figure 34 *J-Link Lite CortexM-9 serial number location*

Configuring KEIL projects for the SEGGER debugger for first time use

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

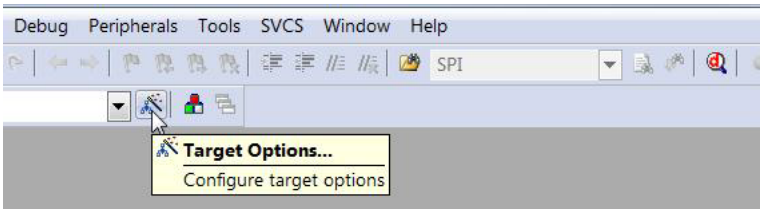


Figure 35 *Keil Target configuration*

3. Under the **Debug** tab in the Use list, select **J-LINK / J-Trace Cortex** option as shown in ***Figure 36***.
4. Click **Settings** as shown in ***Figure 36.*** Both the SEGGER Control Panel and the Keil Target Driver Setup will open.

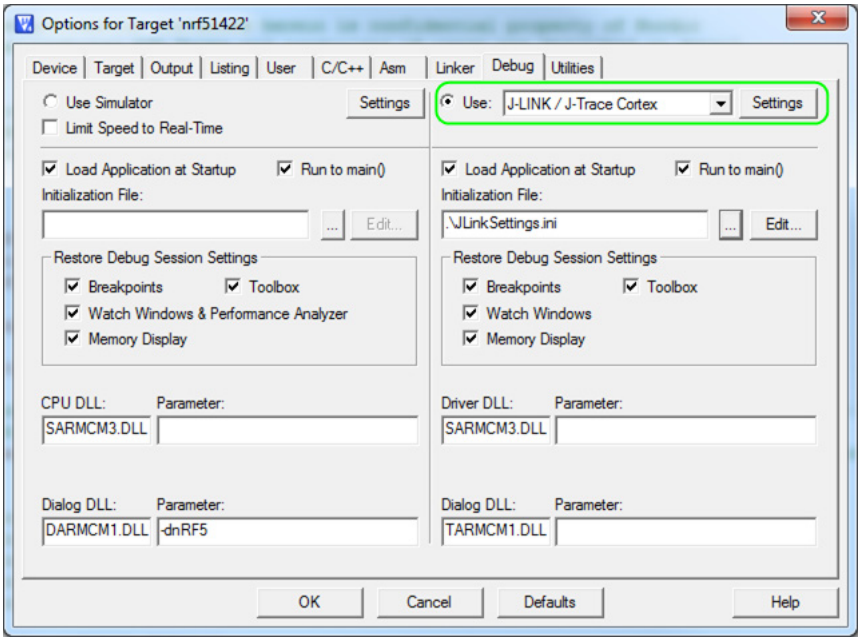


Figure 36 *Selecting JLink debugger in Keil*

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**.

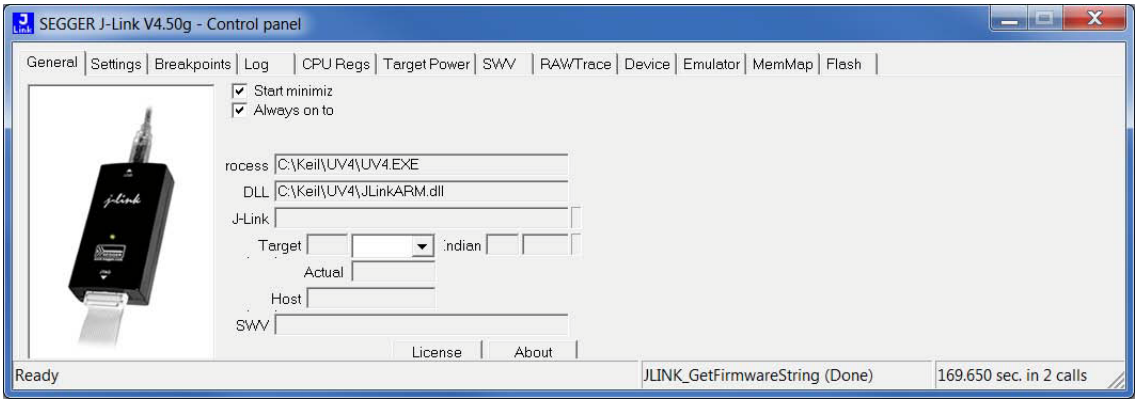


Figure 37 *Segger control panel*

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

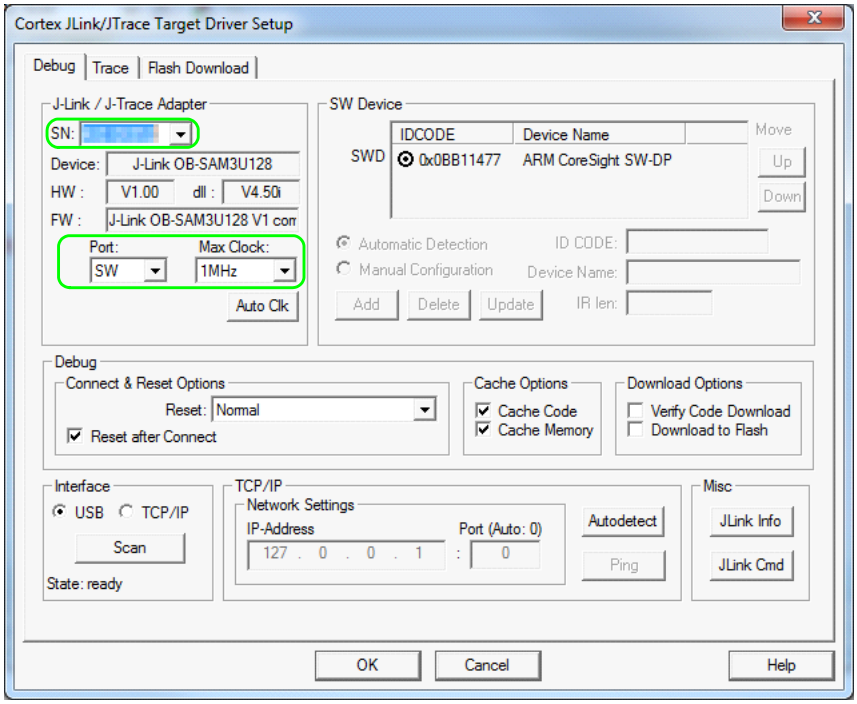


Figure 38 *Debug settings*

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

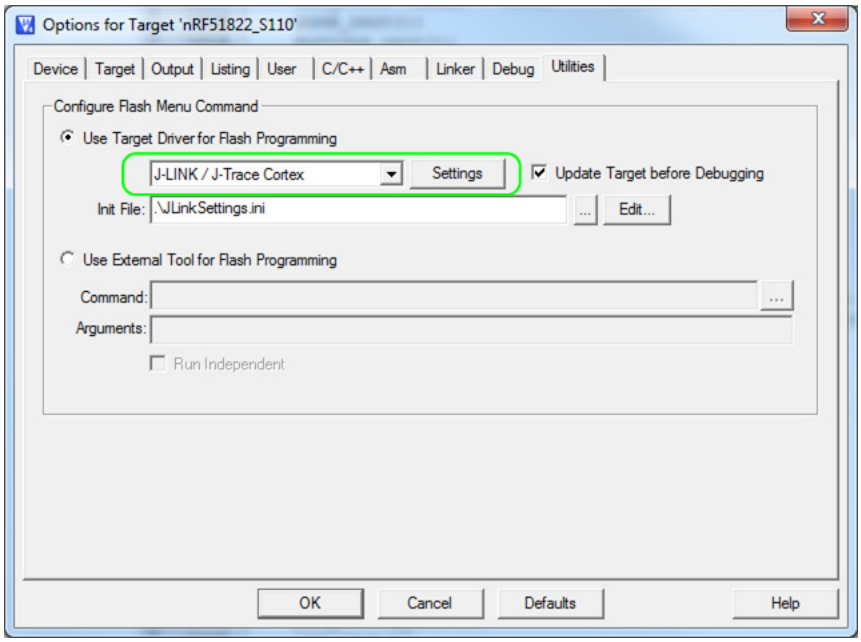


Figure 39 *Flash settings*

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.

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 distributor, please visit 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
August 2013	1.2	Updated content to match v2.1 of the hardware. Updated ***Section 3*** on page 6.
February 2013	1.1	Updated content to match v2.0 of the hardware. Updated hardware information in ***Section 5*** on page 14.
September 2012	1.0	First release

ARM statement