nRF5 Series: Developing on Windows with ARM Keil MDK

Getting Started Guide

v1.1

NORDIC® SEMICONDUCTOR

Contents

Revision history

Date	Version	Description
September 2019	1.1	Fixed broken links
July 2018	1.0	First release

1 Introduction

This guide will help you get started with your nRF5 Series *DK (Development Kit)* and developing your application on Windows with the ARM® Keil MDK.

If you have worked with any of Nordic Semiconductor's products before, you are probably familiar with the required software tools. In this case, this guide will mostly provide reference information.

If this is your first time developing software for a Nordic Semiconductor *System on Chip (SoC)*, this guide will help you set up your development toolchain and guide you through all the steps that are necessary to develop, program, test, and debug your application.

This guide describes how to work with the ARM Keil MDK on a Windows operating system. ARM Keil MDK comes with the ARM C/C++ compiler and the µVision *Integrated Development Environment (IDE)*, and all versions of the nRF5 SDK provide ready-to-use Keil projects.

> **Note:** In MDK-Lite, which is free for evaluation and education, code size is restricted to 32 Kbyte. Since most projects require a bigger code size, we recommend using *SEGGER Embedded Studio (SES)* instead, which is free for use with Nordic Semiconductor devices.

There are two Getting Started Guides that show how to work with different *IDE*s:

- nRF5 Series: Developing with SEGGER Embedded Studio
- nRF5 Series: Developing on Windows with ARM Keil MDK (this document)

Check out the Nordic DevZone for additional setup information and help.

2 Minimum requirements

Ensure that you have all the required hardware and that your computer fulfills the software requirements.

Hardware requirements

- One of the following nRF5 Series development kits:
  - nRF52840 *DK*
  - nRF52 *DK*
  - nRF51 *DK*
- Micro-USB 2.0 cable
- Personal computer (PC)
- Smartphone or tablet that supports *Bluetooth*® Low Energy
- Optional: nRF52840 Dongle or nRF51 Dongle

Software requirements

One of the following operating systems:

- Windows 7, Windows 8, or Windows 10

3 Related documentation

This guide is a starting point and gives essential information for getting started with software development on nRF5 Series devices. For more detailed information, check the documentation of the separate components and tools.

Development Kit User Guides

- nRF52840 Development Kit
- nRF52 Development Kit
- nRF51 Development Kit

Dongle User Guides

- nRF52840 Dongle
- nRF51 Dongle

Compatibility Matrices

- nRF52840 Compatibility Matrix
- nRF52832 Compatibility Matrix
- nRF52810 Compatibility Matrix
- nRF51 Series Compatibility Matrix

nRF5 SDK documentation

- nRF5 SDK v15.3.0

Tools User Guides

- nRF Command Line Tools
- nRF Connect Bluetooth Low Energy

4 Development kits, boards, and chips

Each nRF5 Series *DK* contains a board, which in turn contains a chip. Some of Nordic Semiconductor's tools target the chip, while others target the development board or the development kit. Therefore, you must be aware of the relation between the different components.

You can find all compatibility information in the compatibility matrix for each chip. The following table summarizes the information that you must be aware of for programming your development board.

Development kit	Development board	Chip
nRF52840 *DK*	PCA10056	nRF52840
nRF52840 Dongle	PCA10059	nRF52840
nRF52 *DK*	PCA10040	nRF52832/nRF52810
nRF51 *DK*	PCA10028	nRF51422
nRF51 Dongle	PCA10032	nRF51422

Table 1: Relation between development kits, boards, and chips

5 SoftDevices

A *SoftDevice* is a wireless protocol stack that complements an nRF5 Series *SoC*. Nordic Semiconductor provides them as qualified, precompiled binary files. While it is possible to build applications without using a *SoftDevice*, all nRF5 SDK example applications that use Bluetooth Low Energy or ANT™ require a *SoftDevice*.

See the compatibility matrices for detailed information about which *SoftDevice* versions are supported for each chip. The following table summarizes the usage scenarios for each *SoftDevice*.

Protocol	Role	Chip	SoftDevice
Bluetooth Low Energy	Peripheral	• nRF51422 • nRF51822	S110
		• nRF52810 • nRF52832	S112
	Central or Peripheral	• nRF51422 • nRF51822	S120
	Central and Peripheral	• nRF51422 • nRF51822	S130
		• nRF52832	S132
		• nRF52840	S140
ANT		• nRF51422	S210
		• nRF52832	S212
Bluetooth Low Energy and ANT	Peripheral	• nRF51422	S310
	Central and Peripheral	• nRF52832	S332

Table 2: SoftDevice overview

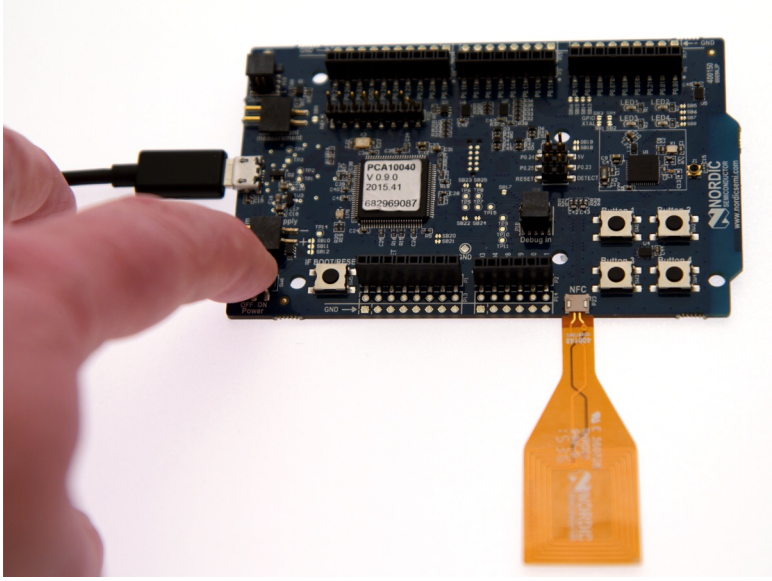
6 Running a first test

Before you start developing, program and run a simple application on your development board to ensure that the board functions as expected and the communication between your computer and your development board works.

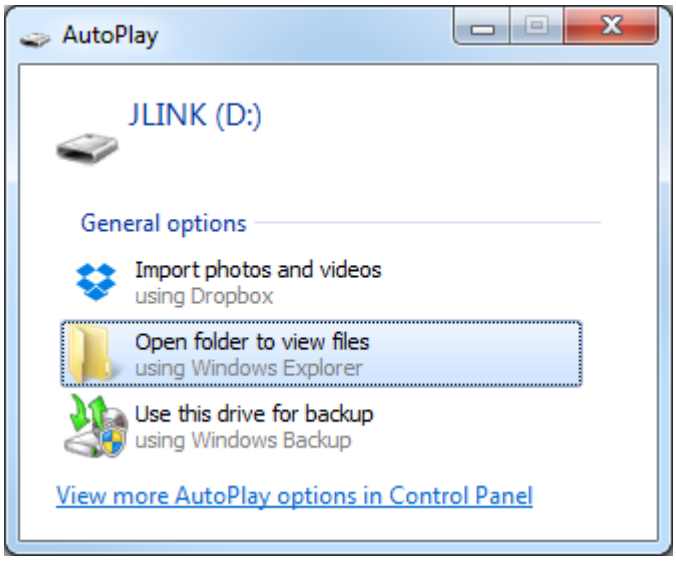
Before you begin, download the latest compatible version of the nRF5 SDK. For information about which SDK supports which IC revisions, check the compatibility matrices.

The nRF5 SDK contains precompiled HEX files of the most common examples. Extract the zip file into a folder of your choosing.

1. Power up the nRF5 Series development board:
   a) Connect one end of a micro-USB 2.0 cable to the USB connector on the board and the other end to one of your PC's USB host ports.
   b) Slide the **power switch** to "ON".



   Observe that **LED1** starts blinking.
2. Open a file explorer and confirm that the development board has appeared as a removable drive named "JLINK".



3. In the folder where you extracted the nRF5 SDK, navigate to `examples\peripheral\blinky\hex`.
4. Select the HEX file that corresponds to your development board and copy it to the JLINK drive. The development board will now restart and run the application.

If the LEDs on the board start blinking, you have successfully programmed your first application! Next, continue to set up your development toolchain and program a more advanced application.

7 Setting up your toolchain

Before you can start developing, you must install the required software. This software includes tools to connect to your development board, an *IDE* for developing your application, and the nRF5 SDK that provides libraries and example applications.

See Nordic tools and downloads on page 11 for an overview of available tools and the links to download the latest versions for your operating system.

The following tools are required:

- nRF5 SDK
- nRF5x Command Line Tools (including nrfjprog)
- Keil MDK-ARM Development Kit

The following tools are optional:

- nRFgo Studio

See the following sections for installation instructions.

7.1 Nordic tools and downloads

This overview lists all available Nordic Semiconductor tools and supported *IDE*s. Not all of these tools are required. To help you pick the *IDE* and tools you want to use, see the following sections for common setup scenarios.

Development IDE

Pick one of the IDEs with a compiler supported by Nordic:

IDE	Windows	Linux	OSX
SEGGER Embedded Studio (SES)	Yes	Yes	Yes
MDK-ARM Keil µVision	Yes	No	No
GNU/GCC	Yes	Yes	Yes
IAR	Yes	No	No

SES is the recommended platform. It is free for use with nRF devices.

Essential tools

You need to download these Nordic tools to develop with our devices.

Tool	Description	Download	Documentation	Protocol
SDK (Software Development Kit)	Application examples, source files, SoftDevices	Windows/Linux	nRF5 SDK v15.3.0 nRF5 SDK for Mesh v3.2.0 nRF5 SDK for Thread and Zigbee v3.1.0	BLE/ANT Bluetooth Mesh Thread and Zigbee
nRF Command Line Tools	Collection of command line tools, like nrfjprog, mergehex	nRF Command Line Tools	nRF Command Line Tools	BLE/ANT

Optional tools

These tools are not essential, but we recommend that you use them.

Tool	Description	Download	Documentation	Protocol
SoftDevice	Wireless protocol stack	Click the Download tab on: nRF52840 Product page nRF52832 Product page nRF52810 Product page nRF51822 Product page nRF51422 Product page	nRF51 SoftDevice Specifications nRF52 SoftDevice Specifications	BLE/ANT
nRF Connect for Desktop	Expandable desktop tool with several apps, including: • Peer device emulator • Power Profiler • Programmer • Cloud Gateway	nRF Connect for OS-X nRF Connect for Ubuntu Linux nRF Connect for Windows	nRF Connect Bluetooth Low Energy	BLE
nRF Connect for Mobile	Peer device emulator app for smartphones	Android v4.3 or later IOS v8 or later		BLE
Nordic nRF Toolbox app	App that contains all the Nordic apps	Android v4.3 or later IOS v8 or later Windows Phone v8.1 or later		BLE
nRF pynrfjprog	Simple Python interface for the nrfjprog DLL	nRF pynrfjprog	nRF5x pynrfjprog	BLE/ANT
ANTware II	Peer device emulator for the ANT protocol running on computers	ANTware II		ANT
nRF Sniffer	App for monitoring on-air traffic	nRF Sniffer download	nRF Sniffer	BLE
nRF Thread Topology Monitor	Tool for visualizing Thread mesh network topology in real time	nRF Thread Topology Monitor for Ubuntu Linux 64-bit nRF Thread Topology Monitor for Windows	nRF Thread Topology Monitor	Thread
Thread Border Router	Gateway for connecting Thread	Thread Border Router	Thread Border Router	Thread

Tool	Description	Download	Documentation	Protocol
	network to the Internet			

See also Nordic mobile apps for a list of available Bluetooth Low Energy and Mesh mobile apps for iOS, Android, and Windows Phones.

7.2 Installing the ARM Keil MDK

The ARM Keil MDK includes the ARM C/C++ compiler and the µVision *IDE*.

Follow the instructions in Getting Started with MDK to download and install the ARM Keil MDK.

Note that there are different editions of the ARM Keil MDK depending on the license that you choose. In the MDK-Lite edition, which is free for evaluation and education, code size is restricted to 32 Kbyte.

7.3 Setting up the nRF5 SDK

The nRF5 SDK does not require installation. You only need to download and extract the files.

If you followed the instructions in Running a first test on page 9, you already downloaded and extracted the nRF5 SDK files and are all set up.

To set up your SDK environment:

1. Download the nRF5 SDK zip file.

   If you have an nRF52 device, select the latest version. For nRF51 devices, select the latest version with support for nRF51 (currently, v12.3.0). For information about which SDK supports which IC revisions, check the compatibility matrices.
2. Extract the zip file to the directory that you want to use to work with the SDK.

   This folder will be referred to as *SDK_dir* in the following documentation.

   Note: Compilers tend to run into problems with long path names. Therefore, place the folder as close to the root level of your file system as possible (for example, at `C:/Nordic/SDK`). Also, avoid using spaces in the file path and folder name.

7.4 Installing the nRF5x Command Line Tools

The nRF5x Command Line Tools are used for developing, programming, and debugging of Nordic Semiconductor's nRF5x SoCs (System on Chip).

When installing the nRF5x Command Line Tools on Windows, SEGGER software is automatically installed in addition to the tools.

Complete the following steps to install the nRF5x Command Line Tools:

1. Download the software for your operating system:
   - Windows: nRF5x-Command-Line-Tools for Win32
2. Install the software by running the installer and following the given instructions.
3. Enter the following command in a command line to make sure that nrfjprog is installed correctly:
   `nrfjprog --version`

   If you get an error message that the command cannot be found, nrfjprog must be manually added to the PATH.

a) Go to the Windows **Advanced system settings** and click **Environment Variables**.
b) Select the Path variable and click **Edit**.
c) Add the following text at the end of the variable value: `;C:\Program Files (x86)\Nordic Semiconductor\nrf5x\bin`

   Make sure that you add a semicolon (;) between entries in the PATH values: `path1;path2`
d) Click **OK** twice.

Open a new command prompt and repeat the command. It should now succeed.

7.5 Installing nRFgo Studio

nRFgo Studio is a desktop application that you can use to program and erase chips.

Installing nRFgo Studio is not required. You can use this tool instead of **`nrfjprog`** if you prefer a graphical interface to a command line tool.

To install nRFgo Studio:

1. Download the latest version of the tool: nRF Command Line Tools
2. Double-click the downloaded file and follow the given instructions.

   The installer will also install the SEGGER J-Link software and the nRF5x Command Line tools.

8 Programming an application

After setting up the required toolchain, you are ready to compile your application and program (or "flash") it to your development board.

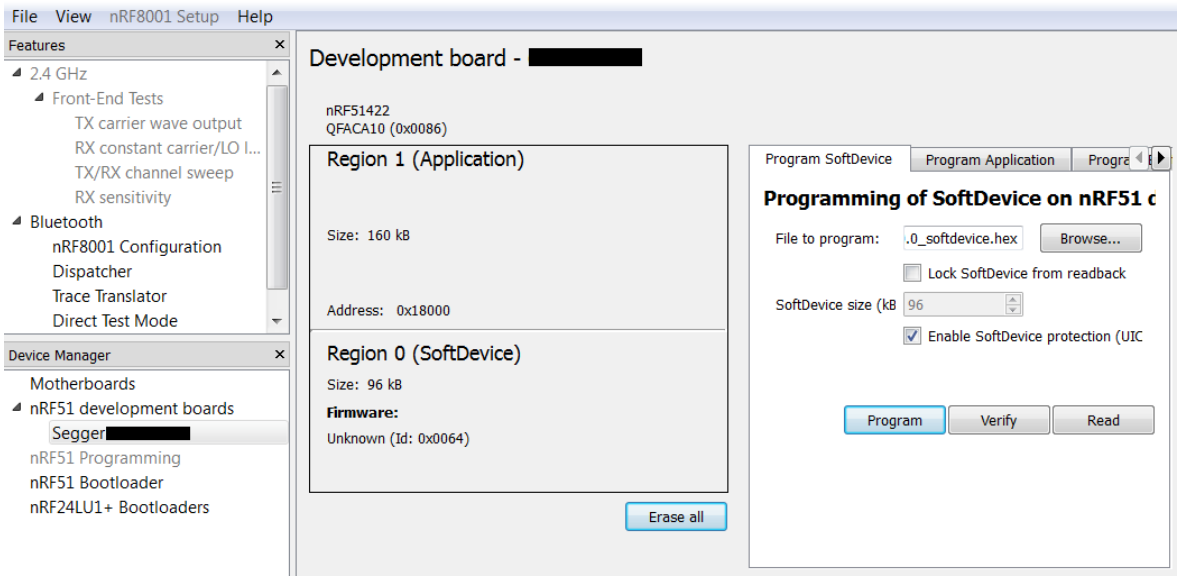
In Windows, you can choose if you want to use a command line tool or a GUI-based tool to program your device.

8.1 Erasing the board

Before you program an example to the development board, you should erase the contents of the board.

There are two ways to erase the board. You can use the command line tool nrfjprog (part of the nRF5x Command Line Tools) or the Windows application nRFgo Studio.

- To erase the contents of the board with nrfjprog, enter the following command:
  - For nRF51 devices: `nrfjprog --family nRF51 --eraseall`
  - For nRF52 devices: `nrfjprog --family nRF52 --eraseall`
- To erase the contents of the board with nRFgo Studio, select your device under nRF51/nRF52 development boards and click **Erase all**.



8.2 Compiling the application

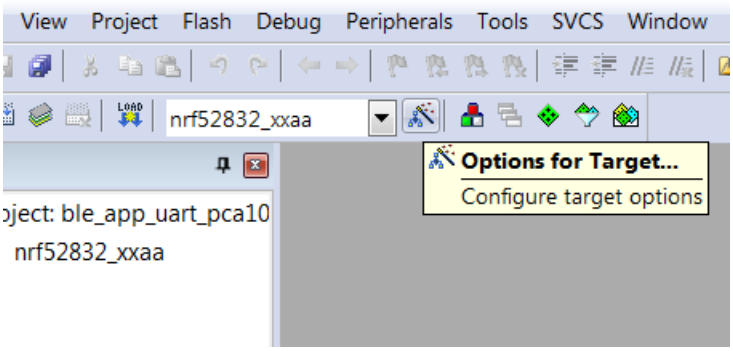
The nRF5 SDK provides example projects for Keil µVision, so compiling the application is very straightforward.

1. In the nRF5 SDK directory, navigate to the folder that contains the example project that you want to run.

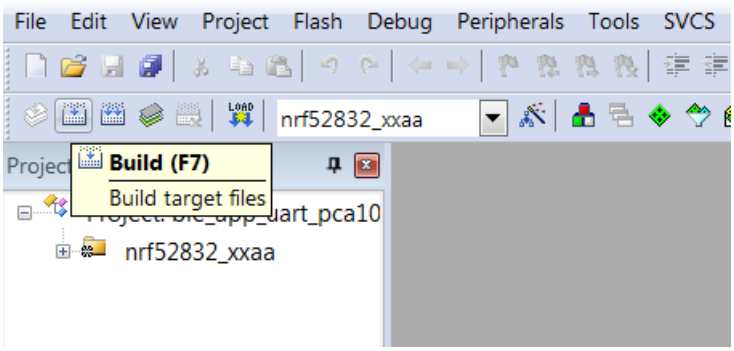
   For a first test, select the `ble_app_uart` example. It is located in the *`SDK_dir`*`\examples\ble_peripheral\ble_app_uart\`*`board`*`\`*`SoftDevice`*`\arm5_no_packs` folder, where *board* is the board and *SoftDevice* is the SoftDevice that you use. For example, for an nRF52 *DK* for nRF52832, go to the following folder:

SDK_dir\examples\ble_peripheral\ble_app_uart\pca10040\s132\arm5_no_packs

2. Double-click the `.uvprojx` file to open the project in Keil.
3. If Keil prompts you to install the nRF_DeviceFamilyPack (also known as nRF5 MDK), accept and install the *Device Family Pack*.
4. If you are not using PCA10028 (nRF51 *DK*) or PCA10040 (nRF52 *DK*), define the correct board in the project.
   a) Go to **Projects** > **Options for Target** or click the shortcut.



   b) In the C/C++ tab, change the `BOARD_PCA10028` define to the board that you are using.
   See the `components\boards\boards.h` file for a list of supported boards.
5. Compile all files by clicking the build button.



8.3 Programming the SoftDevice

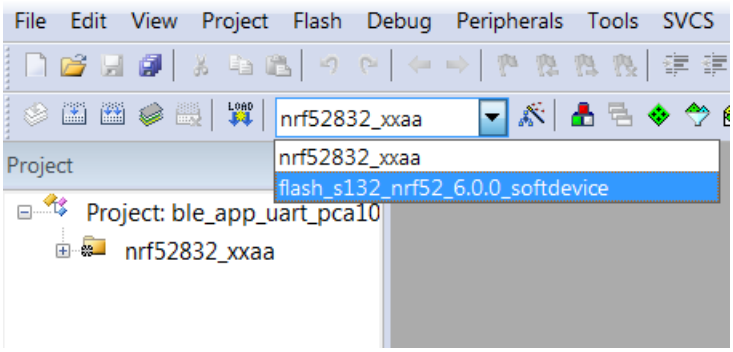
If your application uses Bluetooth or ANT, you must program a *SoftDevice* in addition to the application.

There are several ways to program the *SoftDevice* to the board. You can program the *SoftDevice* directly from Keil, use the command line tool nrfjprog (part of the nRF5x Command Line Tools), or use the Windows application nRFgo Studio.

The Bluetooth *SoftDevice*s are included in the nRF5 SDK and can be found in the *SDK_dir*\components\softdevice\ folder. Check SoftDevices on page 8 for information about which *SoftDevice* is compatible with your development board.

Note: The nRF5 SDK does not contain ANT *SoftDevice*s. You can download them from thisisant.com. The nRF5 SDK does not provide *Target*s to program ANT *SoftDevice*s. Therefore, you must use nrfjprog or nRFgo Studio to program them.

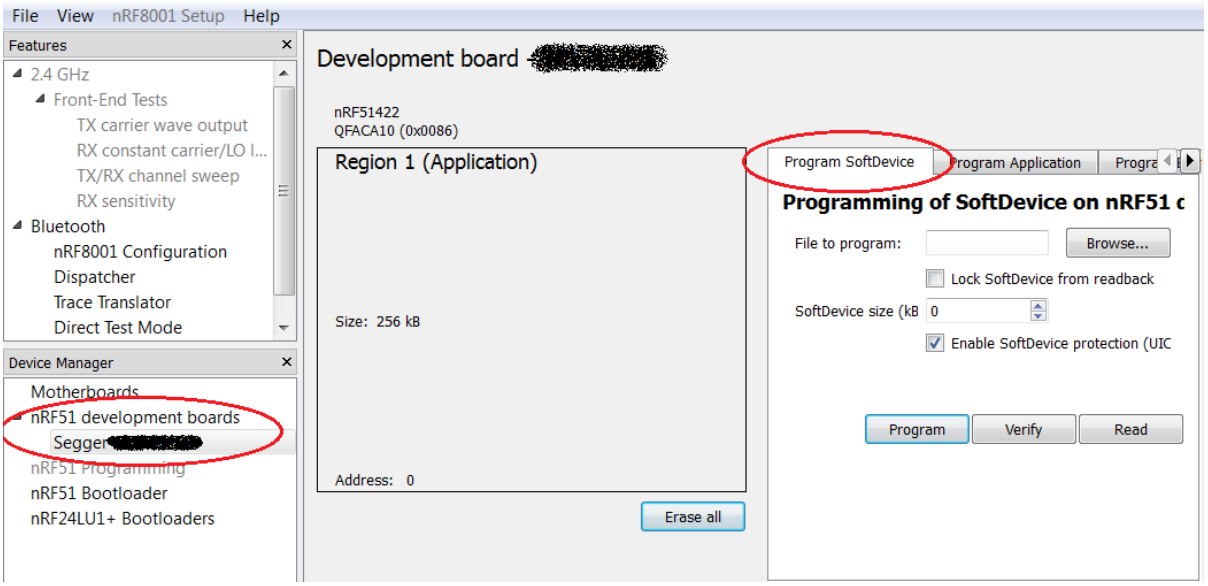
- To program the *SoftDevice* with Keil, complete the following steps:
  a) Instead of the default *Target*, select the *Target* to flash the *SoftDevice*, for example, **flash_s132_nrf52_6.0.0_softdevice**.



b) Click **Options for Target**.
c) Select the Debug pane and click the **Settings** button for the J-Link / J-TRACE Cortex.
d) Select the J-Link / J-Trace Adapter corresponding to the serial number that is printed on your device.
e) Click **OK** to close the dialogs.
f) In the main window, click the **Load** button to program the *SoftDevice*.

- To program the *SoftDevice* with nrfjprog, enter the following command, where *HEX* is the path and file name of the *SoftDevice* HEX file:
  - For nRF51 devices: `nrfjprog --family nRF51 --program HEX`
  - For nRF52 devices: `nrfjprog --family nRF52 --program HEX`
- To program the *SoftDevice* with nRFgo Studio, complete the following steps:

a) Select your device under nRF51/nRF52 development boards.
b) Select the **Program SoftDevice** tab.
c) Click **Browse** and locate the *SoftDevice* HEX file.
d) Click **Program**.

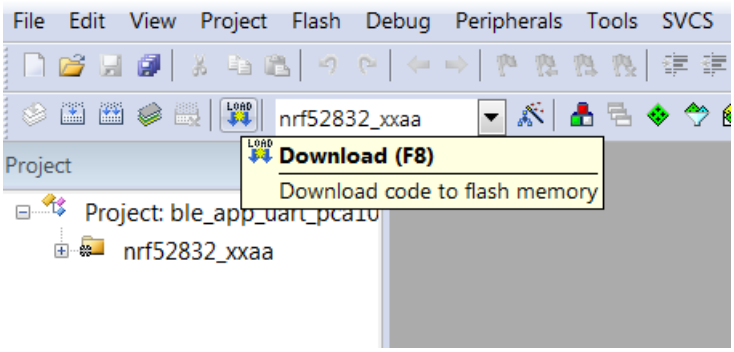


8.4 Programming the application

After compiling the application and programming the *SoftDevice*, you are ready to program the application.

The easiest way to program the application is to do it directly from Keil.

1. Click the **Load** button in Keil.



2. If Keil prompts you to update to the latest firmware version, select **Yes**.

In this example, we compiled and programmed the ble_app_uart application. If everything worked as expected, you should see LED1 blinking on the board approximately every 2 seconds.

If you have more than one board connected to your computer, you will get an error message. In this case, configure which board Keil should use:

1. Select **Project** > **Options for Target 'XXX'**.
2. Switch to the **Debug** tab.
3. Click the **Settings** button next to the selected "J-LINK / J-TRACE Cortex" option.
4. Choose the serial number of your board from the **SN** list in the J-Link / J-Trace Adapter area.
5. Confirm, then click the **Load** button again.

9 Communicating with the board

Unless you programmed a very simple application, you probably want to connect to the board from your computer to display logging information or send input. You can use *Real Time Transfer (RTT)* or *Universal Asynchronous Receiver/Transmitter (UART)* for communicating with the board.

SEGGER Real Time Transfer (RTT) is a proprietary technology for bidirectional communication that supports J-Link devices and ARM-based microcontrollers. The advantage of using *RTT* is that it is very efficient and does not require any other peripheral than the J-Link debugging interface.

Connecting via *UART* is quick and power-efficient, but it requires dedicated use of the *UART* peripheral for logging. The nRF5 *DK*s and the nRF51 Dongle include a *UART* to USB CDC ACM bridge, which is needed to connect to the *UART*. Alternatively, you can use an external *UART* to USB bridge. We use the term CDC-UART to refer to *UART* communication through the *UART* to USB CDC ACM bridge, to distinguish it from communication through the Nordic UART Service (NUS) over Bluetooth Low Energy.

9.1 Connecting via RTT

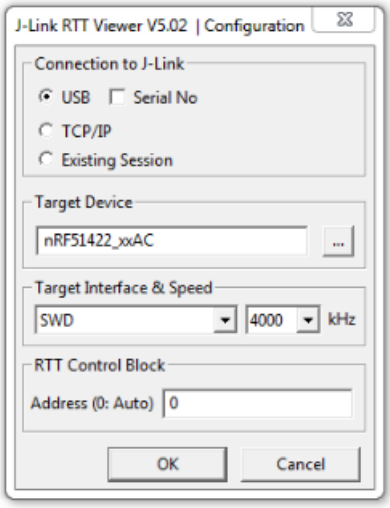
To communicate via *RTT*, connect your development board via USB and run the J-Link RTT Viewer.

The J-Link RTT Viewer is installed as part of the nRF5x Command Line Tools.

To run the J-Link RTT Viewer on Windows, complete the following steps:

1. Select the correct target device.

   The target device is represented by the ID of your development board.
2. Select **SWD** as the target interface.



9.2 Connecting via CDC-UART

To connect via CDC-UART, start a terminal emulator and connect to the used COM port.

There is a wide variety of terminal emulators that you can use, for example, Termite (GUI-based, Windows only) or PuTTY (GUI-based, available for multiple operating systems).

When configuring the connection, use the following *UART* settings:

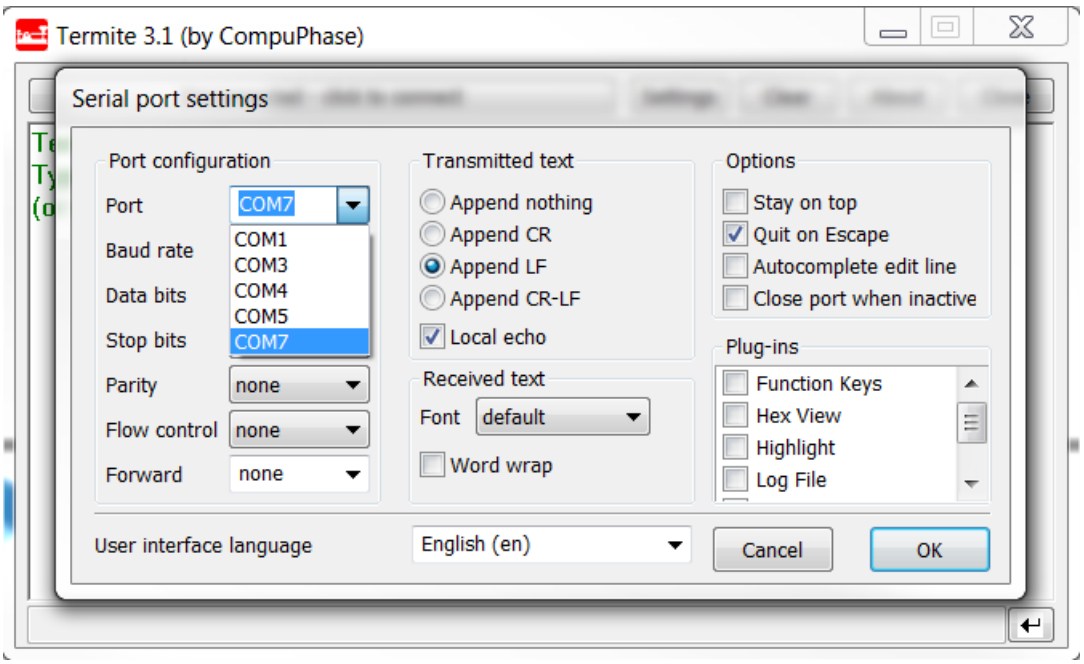
- Baud rate: 115.200

- 8 data bits
- 1 stop bit
- No parity
- HW flow control: RTS/CTS

The following instructions show how to configure Termite correctly. Other terminal emulators can be set up in a similar way.

1. Download and install the latest version of Termite.
2. Connect the development board to your computer.
3. Open Termite and click **Settings**.

   Depending on what devices you have connected to your computer, you might have several choices, as shown in the following figure:



4. Select the correct COM port to connect to the board.

   To find the correct port, follow these steps:

   a) Go to the start menu in Windows and type `devmgmt.msc` to open the Device Manager.
   b) Scroll down and expand **Ports (COM & LPT)**.
   c) Find the port named **JLink CDC UART Port** and note down the number in parentheses.
   d) If you have more than one J-Link UART port, unplug the one that you want to use, plug it back in, and observe which one appeared last.

5. Configure the baud rate and the flow control. Use the default values for the rest of the settings (8 data bits, 1 stop bit, no parity).

   By default, the SDK uses a baud rate of 115200 and RTS/CTS flow control.

6. Make sure that **Append LF** is selected.

   This option appends a newline character to any text that is sent.

7. Configure the terminal to send an RTS (Ready To Send) signal to the development board:

   a) Go to **Settings** > **Plug Ins**.
   b) Enable **Status LEDs** and click **OK**.
   c) Click on the dark green rectangle above RTS to set this signal high.
      The text `Start...` is displayed in Termite.

   Alternatively, you can turn off hardware flow control in your application.

10 Testing the application

The next step after compiling and programming your application is to test it. Nordic Semiconductor provides its own testing tool, nRF Connect, which is available both for mobile and for desktop.

10.1 Testing with a mobile device

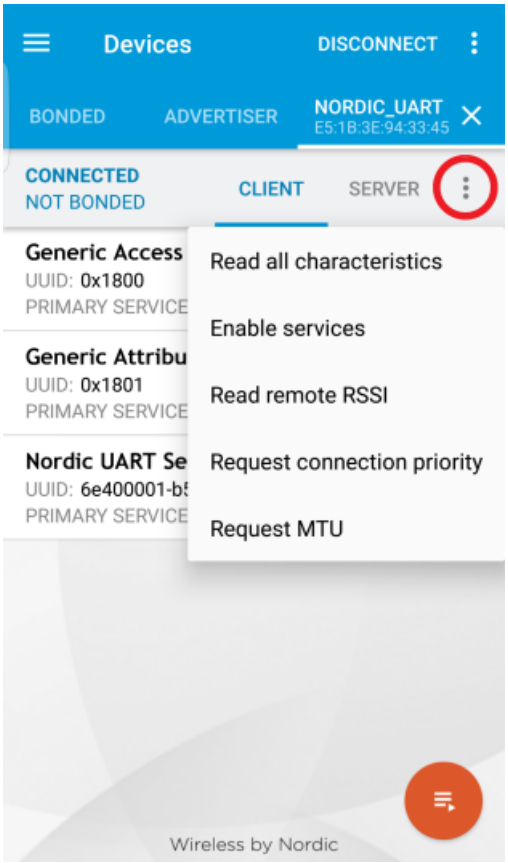
If you have a mobile device that supports Bluetooth Low Energy, download the nRF Connect app from Google Play or App Store to test your application.

The following procedure assumes that you have programmed the ble_app_uart example from the nRF5 SDK. Steps for testing other examples are similar. See the testing instructions for each example in the nRF5 SDK documentation for more information.

1. Download and install nRF Connect from Google Play or App Store.
2. Open nRF Connect.
3. Make sure that the *DK* is running the ble_app_uart example.

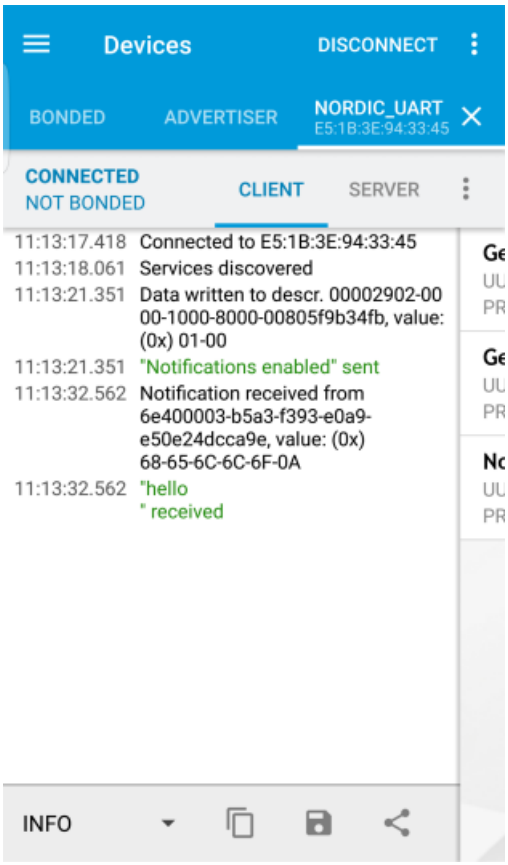
   LED1 should be blinking every 2 seconds, indicating that it is advertising.
4. Tap **Scan**.
5. Find the device and tap **Connect**.

   The default device name for the ble_app_uart example is "Nordic_UART".
6. When connected, tap the options button below the device name and select **Enable services**.



   This example communicates over Bluetooth Low Energy using the Nordic UART Service (NUS).
7. Tap the options button and select **Show log**.

8. In a terminal connected via CDC-UART, enter `hello` and send it to the *DK*.
   The text is sent through the *DK* to your device, which will display it in the nRF Connect log:



10.2 Testing with a computer

If you have an nRF51 Dongle or a second *DK*, you can test your application with nRF Connect for Desktop. nRF Connect for Desktop is available for Windows, Linux, and macOS.

> **Note:** This method requires an nRF5 *DK* or dongle to be connected to your computer.

The following procedure assumes that you have programmed the ble_app_uart example from the nRF5 SDK. Steps for testing other examples are similar. See the testing instructions for each example in the nRF5 SDK documentation for more information.

1. Download and install nRF Connect for Desktop.
2. Connect the dongle or the second *DK* to a USB port of your computer.
3. Connect to the board that runs the ble_app_uart example via CDC-UART.
4. Open nRF Connect for Desktop and add the Bluetooth Low Energy app.
5. Launch the Bluetooth Low Energy app.
6. Select the serial port for the dongle or the *DK* that is connected to your computer (not the board that runs the ble_app_uart example).

   If the device has not been used with the nRF Connect Bluetooth Low Energy app before, you may be asked to update the J-Link firmware and connectivity firmware for the device. You need to have the correct connectivity firmware on the nRF SoC to continue. When the nRF SoC has been programmed with the correct firmware, the nRF Connect Bluetooth Low Energy app proceeds to connect to it over USB. When the connection is established, the device appears in the main view.
7. Click **Start scan**.

8. Find the device and click **Connect**.

   The default device name for the ble_app_uart example is "Nordic_UART".

9. Select the *UART* RX characteristic value.
10. Write `30 31 32 33 34 35 36 37 38 39` (the hexadecimal value for the string "0123456789") and click **write**.
    The text "0123456789" is displayed in the terminal that is connected to the board via *UART*.
11. Enter any text, for example, `Hello`, in the terminal.
    In nRF Connect, the *UART* TX characteristic value changes to the corresponding ASCII value. For example, for `Hello`, the value is `48 65 6C 6C 6F`.

11 Debugging

To actually see what is happening on the development board while the application is running, you must set up a J-Link debugging session. Keil µVision includes a debugger that you can use to step through your application.

See µVision User's Guide: Debugging for detailed information about how to use the debugger.

Glossary

DK (Development Kit)

A development platform used for application development.

Device Family Pack

A software pack that provides hardware descriptions and startup files for nRF5 Series devices. It is also called nRF5 MDK.

GNU Compiler Collection (GCC)

A compiler system that supports various programming languages, maintained by the GNU Project.

Integrated Development Environment (IDE)

A software application that provides facilities for software development.

Real Time Transfer (RTT)

A proprietary technology for bidirectional communication that supports J-Link devices and ARM-based microcontrollers, developed by SEGGER Microcontroller.

SEGGER Embedded Studio (SES)

A cross-platform *IDE* for embedded C/C++ programming with support for Nordic Semiconductor devices, produced by SEGGER Microcontroller.

SoftDevice

A wireless protocol stack that complements the nRF5 Series SoCs. Nordic Semiconductor provides these stacks as qualified, precompiled binary files.

System on Chip (SoC)

A microchip that integrates all the necessary electronic circuits and components of a computer or other electronic systems on a single integrated circuit.

Target

The goal of an operation, for example, programming a specific image on a device, compiling a specific set of files, or removing previously generated files.

Universal Asynchronous Receiver/Transmitter (UART)

A hardware device for asynchronous serial communication between devices.

Acronyms and abbreviations

These acronyms and abbreviations are used in this document.

DK
Development Kit

GCC
GNU Compiler Collection

IDE
Integrated Development Environment

RTT
SEGGER Real Time Transfer

SES
SEGGER Embedded Studio

SoC
System on Chip

UART
Universal Asynchronous Receiver/Transmitter

Legal notices

By using this documentation you agree to our terms and conditions of use. Nordic Semiconductor may change these terms and conditions at any time without notice.

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.

Nordic Semiconductor ASA does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. If there are any discrepancies, ambiguities or conflicts in Nordic Semiconductor's documentation, the Product Specification prevails.

Nordic Semiconductor ASA reserves the right to make corrections, enhancements, and other changes to this document without notice.

Life support applications

Nordic Semiconductor 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.

RoHS and REACH statement

Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website www.nordicsemi.com.

Trademarks

All trademarks, service marks, trade names, product names, and logos appearing in this documentation are the property of their respective owners.

Copyright notice

COMPANY WITH
QUALITY SYSTEM
CERTIFIED BY DNV GL
= ISO 9001 =