

Getting started with nRF5 SDK and Keil (nRF51 & nRF52 Series)

Getting Started Guide

v1.2

Contents

Revision history	iii
1 Introduction	4
2 Minimum requirements	5
3 Related documentation	6
4 Development kits, PCA numbers, and chips	7
5 SoftDevices	8
6 Running a first test	9
7 Setting up your toolchain	11
7.1 Nordic tools and downloads	11
7.2 Installing the ARM Keil MDK	14
7.3 Setting up the nRF5 SDK	14
7.4 Installing the nRF Command Line Tools	14
7.5 Installing nRF Connect Programmer	15
8 Programming an application	16
8.1 Erasing the kit	16
8.2 Compiling the application	16
8.3 Programming the SoftDevice	17
8.4 Programming the application	18
9 Communicating with the kit	20
9.1 Connecting via RTT	20
9.2 Connecting via CDC-UART	20
10 Testing the application	22
10.1 Testing with a mobile device	22
10.2 Testing with a computer	23
11 Debugging	25
Glossary	26
Acronyms and abbreviations	27
Legal notices	28

Revision history

Date	Version	Description
April 2020	1.2	<ul style="list-style-type: none">• Changed the document title to clarify that this Getting Started Guide targets the nRF5 SDK• Added Installing nRF Connect Programmer on page 15• Updated Erasing the kit on page 16 and Programming the SoftDevice on page 17 to use nRF Connect Programmer instead of nRFgo Studio• Editorial changes
September 2019	1.1	Fixed broken links
July 2018	1.0	First release

1 Introduction

This guide will help you get started with your nRF51 or nRF52 Series *Development Kit (DK)* and developing your application with the nRF5 SDK 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 *Software Development Kit (SDK)* and the required tools. In this case, this guide will mostly provide reference information.

Use this guide to set up your development toolchain so you can 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 μ Vision *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.

The following Getting Started Guides show how to work with different software development platforms and devices:

- [Getting started with nRF Connect SDK \(nRF53 Series\)](#)
- [Getting started with nRF Connect SDK \(nRF52 Series\)](#)
- [Getting started with nRF5 SDK and SES \(nRF51 & nRF52 Series\)](#)
- [Getting started with nRF5 SDK and Keil \(nRF51 & nRF52 Series\)](#) (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 development kits:
 - nRF52840 DK
 - nRF52833 DK
 - nRF52 DK
 - nRF51 DK
- Micro-USB 2.0 cable
- Personal computer (PC)
- Optional for testing:
 - Smartphone or tablet that supports *Bluetooth*[®] Low Energy
 - nRF52840 Dongle, nRF51 Dongle, or a second Nordic *DK*

Software requirements

One of the following operating systems:

- Windows 8 or Windows 10

3 Related documentation

In addition to the information in this document, you may need to refer to other documents.

Development Kit User Guides

- [nRF52840 DK](#)
- [nRF52833 DK](#)
- [nRF52 DK](#)
- [nRF51 DK](#)

Dongle User Guides

- [nRF52840 Dongle](#)
- [nRF51 Dongle](#)

Compatibility Matrices

- [nRF52840 Compatibility Matrix](#)
- [nRF52833 Compatibility Matrix](#)
- [nRF52832 Compatibility Matrix](#)
- [nRF52811 Compatibility Matrix](#)
- [nRF52810 Compatibility Matrix](#)
- [nRF51 Series Compatibility Matrix](#)

SDK documentation

- [nRF5 SDK v16.0.0](#)

Tools User Guides

- [nRF Connect Bluetooth Low Energy](#)
- [nRF Command Line Tools](#)

4

Development kits, PCA numbers, and chips

Nordic Semiconductor's software tools either target the chip that is soldered onto the kit's development kit, or target the development kit itself.

The following table lists the PCA number and the chip bundled in each kit.

Development kit	PCA number	Chip
nRF52840 DK	PCA10056	nRF52840
nRF52840 Dongle	PCA10059	nRF52840
nRF52833 DK	PCA10100	nRF52833
nRF52 DK	PCA10040	nRF52832/nRF52810
nRF51 DK	PCA10028	nRF51422
nRF51 Dongle	PCA10032	nRF51422

Table 1: Relation between development kits, PCA numbers, and chips

5 SoftDevices

A *SoftDevice* is a wireless protocol stack that complements an nRF5 Series *System on Chip (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	<ul style="list-style-type: none"> nRF51422 nRF51822 	S110
		<ul style="list-style-type: none"> nRF52810 nRF52832 	S112
		<ul style="list-style-type: none"> nRF52833 	S113
	Central or Peripheral	<ul style="list-style-type: none"> nRF51422 nRF51822 	S120
	Central and Peripheral	<ul style="list-style-type: none"> nRF51422 nRF51822 	S130
		<ul style="list-style-type: none"> nRF52832 	S132
		<ul style="list-style-type: none"> nRF52833 nRF52840 	S140
ANT		<ul style="list-style-type: none"> nRF51422 	S210
		<ul style="list-style-type: none"> nRF52832 	S212
Bluetooth Low Energy and ANT	Peripheral	<ul style="list-style-type: none"> nRF51422 	S310
	Peripheral and Broadcaster	<ul style="list-style-type: none"> nRF52810 	S312
	All roles	<ul style="list-style-type: none"> nRF52832 	S332
		<ul style="list-style-type: none"> nRF52840 	S340

Table 2: SoftDevice overview

6 Running a first test

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

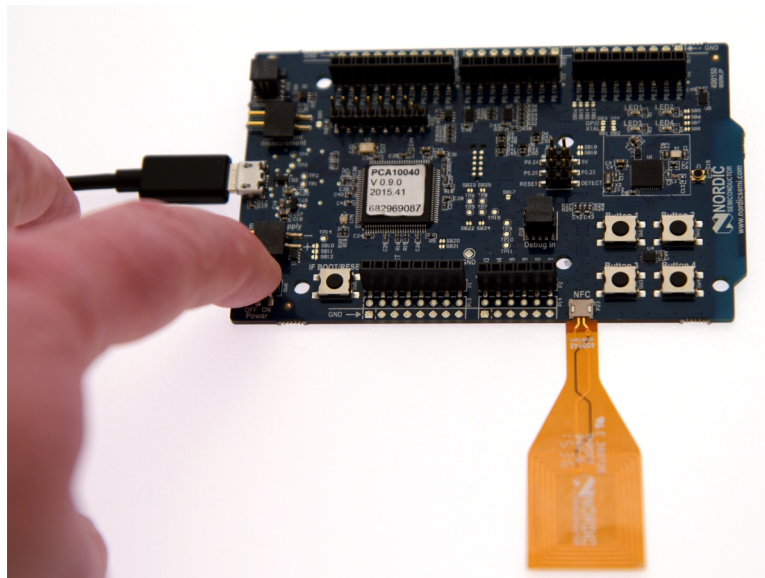
1. Download the latest compatible version of the [nRF5 SDK](#).

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

For information about which SDK supports which IC revisions, check the [compatibility matrices](#).

2. Power up the development kit:

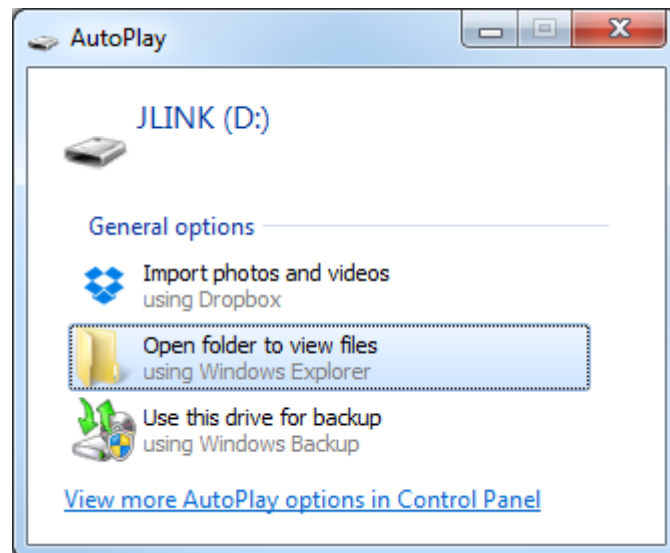
- a) Connect one end of a micro-USB 2.0 cable to the *Universal Serial Bus (USB)* connector on the kit and the other end to one of your PC's *USB* host ports.
- b) Slide the **power switch** to ON.



Observe that **LED1** starts blinking.

3. Open a file explorer and confirm that the development kit has appeared as a removable drive named **JLINK**.

On Windows, you should see a pop-up window similar to this:



4. In the folder where you extracted the nRF5 SDK, navigate to `examples\ble_peripheral\ble_app_hrs\hex`.
5. Select the HEX file that corresponds to your development kit and copy it to the JLINK drive. The development kit will now restart and run the application. Note that while restarting, the JLINK drive will be disconnected.
6. Download and install the [Nordic nRF Toolbox app](#) from Google Play or App Store.
7. Open nRF Toolbox.

Note: Enable Bluetooth if prompted.

8. Tap **HRM**.
9. Tap **Connect**.
10. Select **Nordic_HRS**.
A simulated heart rate and battery charge percentage is displayed.

For more advanced testing scenarios, see [Testing the application](#) on page 22.

Next, continue to set up your development toolchain and build and program an application from the source code.

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 kit, 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 for this Getting Started Guide:

- [Keil MDK-ARM Development Kit](#)
- [nRF5 SDK](#)
- [nRF Command Line Tools](#) (including nrfjprog)

The following tool is optional:

- [nRF Connect Programmer](#)

See the following sections for installation instructions.

7.1 Nordic tools and downloads

This overview lists all available Nordic Semiconductor tools and supported *IDEs*. 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 v16.0.0 nRF5 SDK for Mesh v4.1.0 nRF5 SDK for Thread and Zigbee v4.0.0	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	Compatible downloads for nRF52840 Compatible downloads for nRF52833 Compatible downloads for nRF52832 Compatible downloads for nRF52811 Compatible downloads for nRF52810 Compatible downloads for nRF51822 Compatible downloads for nRF51422	nRF51 SoftDevice Specifications nRF52 SoftDevice Specifications	BLE/ANT
nRF Connect for Desktop	Expandable desktop tool with several apps, including: <ul style="list-style-type: none"> Peer device emulator Power Profiler Programmer Cloud Gateway 	nRF Connect for Desktop	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 for Bluetooth LE	BLE
nRF Thread Topology Monitor	Tool for visualizing Thread mesh network topology in real time	nRF Thread Topology Monitor	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 μ Vision *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.

Complete the following steps 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 nRF Command Line Tools

The nRF Command Line Tools are used for developing, programming, and debugging Nordic Semiconductor's SoCs.

Complete the following steps to install the nRF Command Line Tools and verify the installation:

1. Follow the instructions in [Installing the nRF Command Line Tools](#) to download and install the nRF Command Line Tools.

2. 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 nRF Connect Programmer

[nRF Connect Programmer](#) is an app for [nRF Connect for Desktop](#) that you can use to program and erase chips.

Installing nRF Connect Programmer is not required. You can use this tool instead of `nrfjprog` if you prefer a graphical interface to a command line tool.

Follow the instructions in [Installing the Programmer](#) to install nRF Connect for Desktop and the Programmer app.

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 kit.

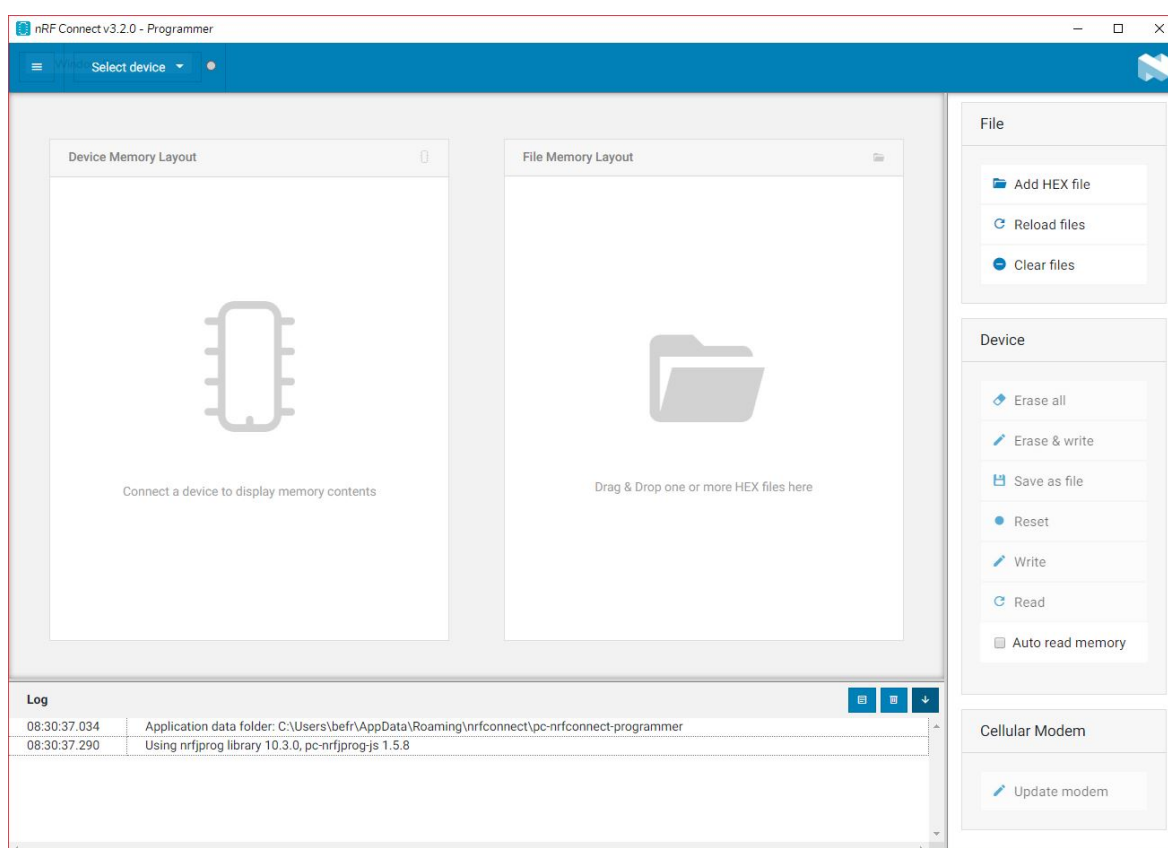
In Windows, you can choose if you want to use a command line tool or a GUI-based tool to program your device.

8.1 Erasing the kit

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

There are two ways to erase the kit. You can use the command line tool `nrfjprog` (part of the nRF Command Line Tools) or the [nRF Connect Programmer](#).

- To erase the contents of the kit 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 kit with the nRF Connect Programmer, select your device in the navigation bar. Then click **Erase all** in the **Device** section.



8.2 Compiling the application

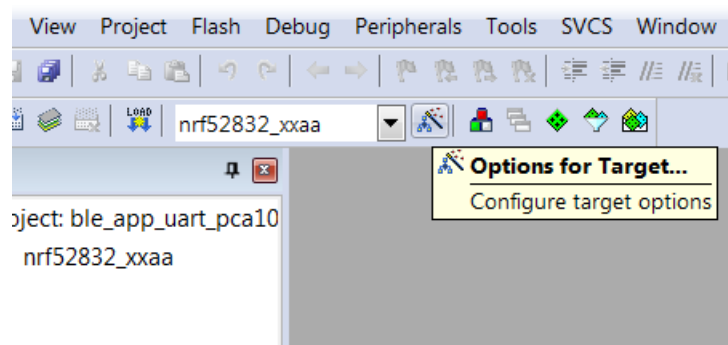
The nRF5 SDK provides example projects for Keil μ Vision, so compiling the application is very straightforward.

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\pca_number\SoftDevice\arm5_no_packs` folder, where `pca_number` is the [PCA number](#) and `SoftDevice` is the [SoftDevice](#) that you use. For example, for an nRF52 DK for nRF52832, go to the following folder:

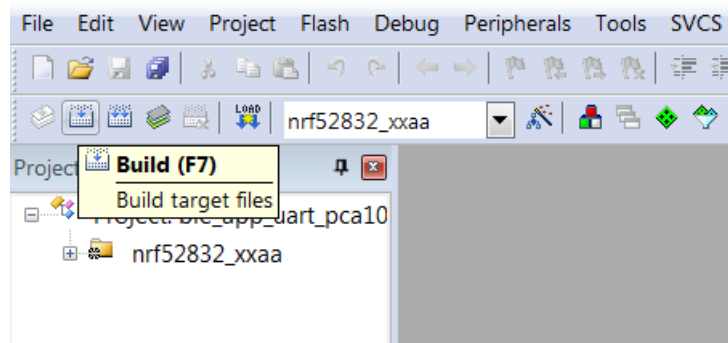
```
SDK_dir\examples\ble_peripheral\ble_app_uart
\pca10040\s132\arm5_no_packs
```

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 kit 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 kit that you are using. See the `components\boards\boards.h` file for a list of supported kits.

5. Compile all files by clicking the build button.



8.3 Programming the SoftDevice

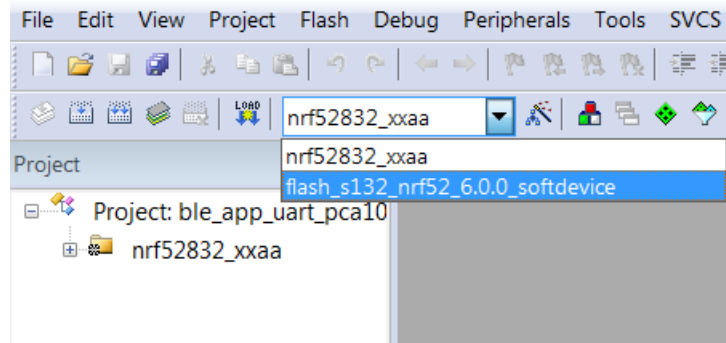
If your application uses Bluetooth or ANT, you must program a *SoftDevice* in addition to the application.

There are several ways to program the *SoftDevice* to the kit. You can program the *SoftDevice* directly from Keil, use the command line tool `nrfjprog` (part of the nRF Command Line Tools), or use nRF Connect Programmer.

The Bluetooth *SoftDevices* 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 kit.

Note: The nRF5 SDK does not contain ANT *SoftDevices*. You can download them from thisisant.com. The nRF5 SDK does not provide *Targets* to program ANT *SoftDevices*. Therefore, you must use nrfjprog or nRF Connect Programmer to program them.

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



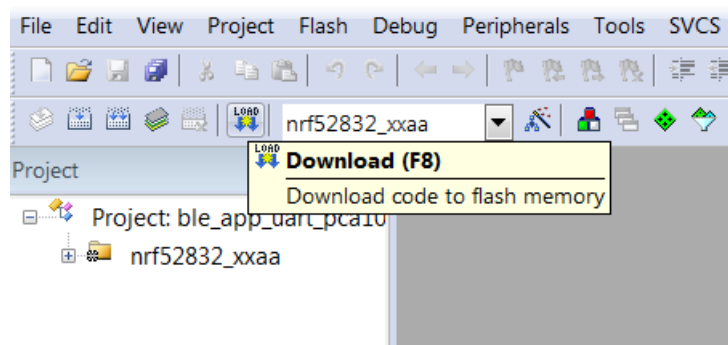
- Click **Options for Target**.
 - Select the Debug pane and click the **Settings** button for the J-Link / J-TRACE Cortex.
 - Select the J-Link / J-Trace Adapter corresponding to the serial number that is printed on your device.
 - Click **OK** to close the dialogs.
 - In the main window, click the **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 nRF Connect Programmer, follow the instructions in [Programming a Development Kit or the nRF51 Dongle](#).

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.

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



- 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 kit approximately every 2 seconds.

If you have more than one kit connected to your computer, you will get an error message. In this case, configure which kit 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 kit from the **SN** list in the J-Link / J-Trace Adapter area.
5. Confirm, then click the **Load** button again.

9 Communicating with the kit

If your application outputs logging information or needs console input, you should connect the kit to your computer to interact with a console. You can use *Real Time Transfer (RTT)* or *Universal Asynchronous Receiver/Transmitter (UART)* for communicating with the kit.

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 DKs 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 kit via *USB* and run the J-Link RTT Viewer.

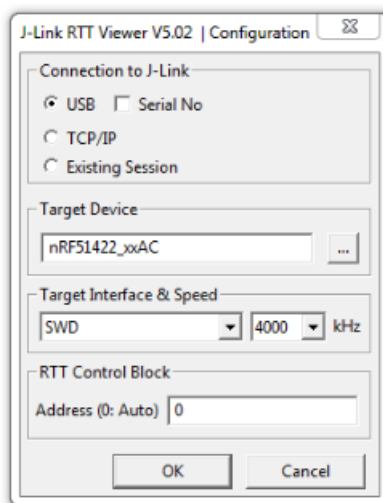
The J-Link RTT Viewer is installed as part of the nRF Command Line Tools.

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 kit.

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, Termit (GUI-based, Windows only) or PuTTY (GUI-based, available for multiple operating systems).

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

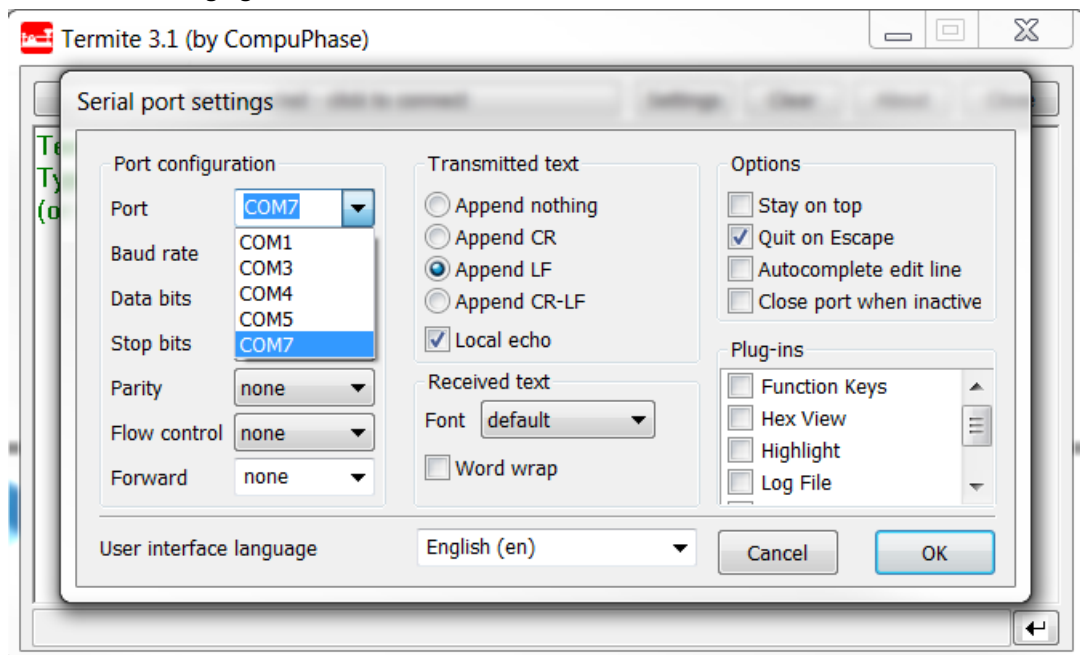
- Baud rate: 115200 (default baud rate for most examples in the nRF5 SDK)

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

The following instructions show how to configure Terminate on Windows. Other GUI-based terminal emulators can be set up in a similar way.

1. Download and install the latest version of [Termite](#).
2. Connect the development kit 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 kit.
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 kit:
 - 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.

10 Testing the application

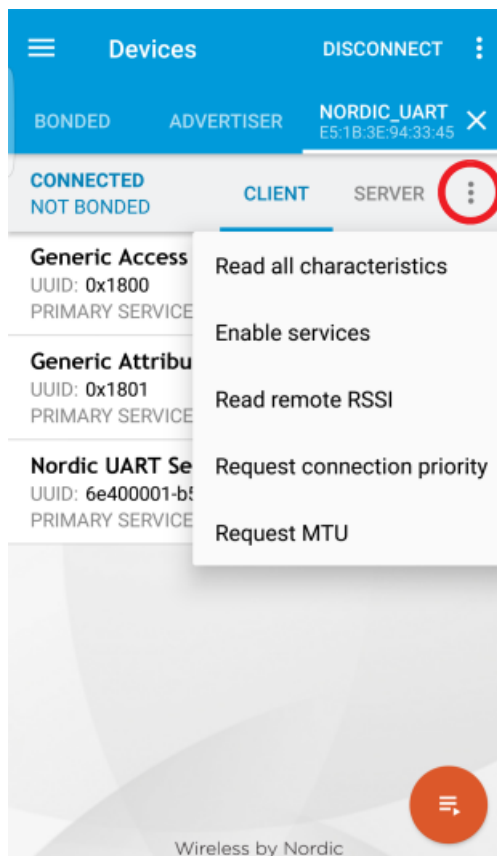
The next step after compiling and programming your application is to test it. With the nRF Connect platform, Nordic Semiconductor provides its own testing apps, which are available both for mobile and for desktop.

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.

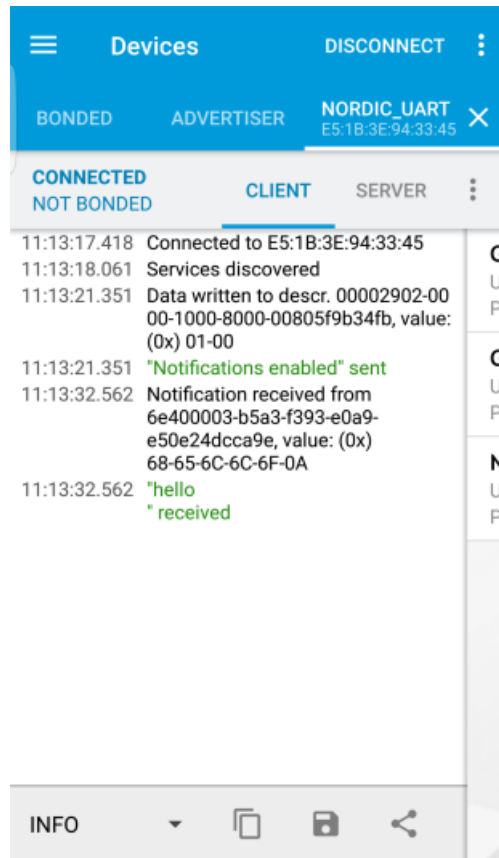
The following procedure assumes that you have programmed the Bluetooth Low Energy Peripheral UART example from the nRF5 SDK (`examples/ble_peripheral/ble_app_uart`) on your development kit. Steps for testing other examples are similar. See the testing instructions for each example in the [nRF5 SDK documentation](#) for more information.

1. Download and install nRF Connect from Google Play or App Store.
2. Open nRF Connect.
3. Make sure that the development kit 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 Peripheral 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 development kit. The text is sent through the development kit to your mobile device over a Bluetooth Low Energy link. The mobile device will then display it in the nRF Connect log:



10.2 Testing with a computer

If you have a dongle or a second Nordic *DK*, you can test your application with the Bluetooth Low Energy app in nRF Connect for Desktop. nRF Connect for Desktop is available for Windows, Linux, and macOS.

Note: This method requires a dongle or a second Nordic *DK* with Bluetooth Low Energy support to be connected to your computer.

The following procedure assumes that you have programmed the Bluetooth Low Energy Peripheral UART example from the nRF5 SDK ([examples/ble_peripheral/ble_app_uart](#)) on your development kit. 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 development kit to a *USB* port of your computer.
3. [Connect to the kit that runs the ble_app_uart example via CDC-UART](#).
4. Open nRF Connect for Desktop and install the Bluetooth Low Energy app.
See [nRF Connect for Desktop](#) for information about how to install and open apps.
5. Launch the Bluetooth Low Energy app.
6. Select the serial port for the dongle or the development kit that is connected to your computer (not the kit that runs the Peripheral 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 must 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 Peripheral 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 data is transmitted over Bluetooth Low Energy from the app to the development kit that runs the Peripheral UART example. The text "0123456789" is displayed in the terminal that is connected to the kit that runs the Peripheral UART example 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, the value for `Hello` is 48 65 6C 6C 6F.

11 Debugging

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

See [\$\mu\$ Vision User's Guide: Debugging](#) for detailed information about how to use the debugger.

Glossary

Development Kit (DK)

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.

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.

Software Development Kit (SDK)

A set of tools used for developing applications for a specific device or operating system.

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.

Universal Serial Bus (USB)

An industry standard that establishes specifications for cables and connectors and protocols for connection, communication, and power supply between computers, peripheral devices, and other computers.

Acronyms and abbreviations

These acronyms and abbreviations are used in this document.

DK

Development Kit

IDE

Integrated Development Environment

RTT

SEGGER Real Time Transfer

SDK

Software Development Kit

SES

SEGGER Embedded Studio

SoC

System on Chip

UART

Universal Asynchronous Receiver/Transmitter

USB

Universal Serial Bus

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

© 2020 Nordic Semiconductor ASA. All rights are reserved. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

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