Getting started with nRF Connect SDK (nRF53 Series)

Getting Started Guide

NORDIC®
SEMICONDUCTOR

Contents

NORDIC
SEMICONDUCTOR

Revision history

Date	Description
2022-06-02	Updated: • Installing the nRF Connect SDK on page 7 • Programming an application on page 9 • Debugging on page 15
2021-02-11	Removed *Installing SEGGER Embedded Studio Nordic Edition* (part of Installing the nRF Connect SDK on page 7)
2020-12-03	• Updated for nRF5340 DK • Editorial changes
April 2020	• Updated Running a first test on page 6 to use nRF Connect Programmer instead of the command line • Removed *Installing the nRF Command Line Tools* • Updated Installing the nRF Connect SDK on page 7 with information about the Toolchain manager • Updated Programming an application on page 9 • Updated Debugging on page 15 • Editorial changes
November 2019	First release

NORDIC
SEMICONDUCTOR

1 Introduction

This guide will help you get started with your nRF5340 *Development Kit (DK)* and developing your application with the nRF Connect SDK.

The nRF Connect SDK is a *Software Development Kit (SDK)* that contains optimized cellular IoT (LTE-M and NB-IoT), *Bluetooth*® Low Energy, Thread, Zigbee, and Bluetooth Mesh stacks, a range of applications, samples, and reference implementations, as well as a full suite of drivers for Nordic Semiconductor's devices. It includes the Zephyr™ *Real-time operating system (RTOS)*, which is built for connected low power products.

Use this guide to set up your development toolchain so you can develop, program, test, and debug your application.

NORDIC
SEMICONDUCTOR

2 Minimum requirements

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

Hardware requirements

- nRF5340 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
- macOS
- Linux

3 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 Heart Rate Monitor (HRM) demo application from nRF5340 DK Downloads and extract it into a folder of your choice.

   There will be two HEX files in the folder, one for the application core (`peripheral_hr.hex`) and one for the network core (`hci_open_amp.hex`).

2. Install nRF Connect for Desktop and nRF Connect Programmer.

   See Installing the Programmer for instructions.

3. Use nRF Connect Programmer to program the HEX files of the demo application.

   See Programming a Development Kit or the nRF51 Dongle for more information.

4. Press the **reset** button.
   The development kit will now restart and run the application. Note that while restarting, the JLINK drive will be disconnected.

5. Download and install the Nordic nRF Toolbox app from Google Play or App Store.

6. Open nRF Toolbox.

   > **Note:** Enable Bluetooth if prompted.

7. Tap **HRM**.

8. Tap **Connect**.

9. Select **Zephyr Heartrate Sensor**.
   A simulated heart rate and battery charge percentage is displayed.

For more advanced testing scenarios, see Testing the application on page 12.

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

NORDIC
SEMICONDUCTOR

4 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 *Integrated Development Environment (IDE)* for developing your application, and the nRF Connect SDK that provides libraries and example applications.

The following tools are required for this Getting Started Guide:

- nRF Connect SDK

The following tool is optional:

- nRF Connect Programmer

See the following sections for installation instructions.

4.1 Installing the nRF Connect SDK

The nRF Connect SDK includes the libraries and samples that you need to build an application. It also contains the required development tools, including nRF Connect for Visual Studio Code.

The recommended way to install the nRF Connect SDK is through an app in nRF Connect for Desktop.

nRF Connect for Desktop provides different apps to simplify installing the nRF Connect SDK, depending on the operating system that you are using.

> **Note:** If you prefer to install the nRF Connect SDK manually, see Installing the nRF Connect SDK manually for instructions.

Complete the following steps to install the nRF Connect SDK using an nRF Connect for Desktop app:

1. Download and install nRF Connect for Desktop.
2. Install and run one of the following apps:

    - On Windows or macOS, use the Toolchain manager. This app installs the full sandboxed toolchain that you need, including nRF Connect for Visual Studio Code and the nRF Connect SDK source code.
    - On Linux, use nRF Connect for Visual Studio Code. This app assists you in setting up the toolchain and the nRF Connect SDK source code.

    See nRF Connect for Desktop for information about how to install and open apps.
3. Follow the instructions in the app to install the nRF Connect SDK.
4. Make sure that you have the correct version of the nRF Connect SDK source code. Unless instructed otherwise, you should work with the last tagged release of the nRF Connect SDK.

    - If you used the Toolchain manager app, you installed a specific version and no further action is required.
    - If you used nRF Connect for Visual Studio Code, make sure that you checked out the correct version and ran `west update`.

    If you want to switch to a different tagged release or to the latest version on the main branch, see Updating the repositories for instructions.

4.2 Installing the Programmer

The Programmer is installed as an app for nRF Connect for Desktop.

NORDIC
SEMICONDUCTOR

Before you can install the Programmer, you must download and install nRF Connect for Desktop (version 3.2.0 or later).

To install the Programmer:

1. Open nRF Connect for Desktop.

2. Find the Programmer in the list of apps and click **Install**.

Once the app is installed, you can launch it by clicking **Open**.

For easy access, you can create a desktop shortcut by clicking the **arrow down** button and selecting **Create shortcut**.

If a new version of the app becomes available, an **Update** button is displayed next to the **Open** button. Click this button to install the latest version. To uninstall the app, click the **arrow down** button and select **Uninstall**.

5 Programming an application

Use nRF Connect for Visual Studio Code to build and program an nRF Connect SDK sample. See Building and programming a sample application for detailed instructions.

If you prefer to build and program your application from the command line, see Building on the command line for instructions. You can also use nRF Connect Programmer to program your application.

The following user guides in the nRF Connect SDK documentation explain the concepts that you need to know to work with nRF53 Series devices in the nRF Connect SDK:

- Working with nRF5340
- Bluetooth LE Controller
- Modifying a sample application
- Application Development

You can find documentation for all available samples under Samples, Applications, and Samples and Demos.

The following sections assume that you have programmed the Bluetooth Low Energy Peripheral UART example from the nRF Connect SDK (`ncs/nrf/samples/bluetooth/peripheral_uart`).

6 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 *Universal Asynchronous Receiver/Transmitter (UART)* for communicating with the kit.

Connecting via *UART* is quick and power-efficient, but it requires dedicated use of the *UART* peripheral for logging. Alternatively, you can use an external *UART* to *Universal Serial Bus (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.

6.1 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: 115200 (default baud rate for most samples in the nRF Connect SDK)
- 8 data bits
- 1 stop bit
- No parity
- HW flow control: RTS/CTS

The following instructions show how to configure Termite 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:

NORDIC
SEMICONDUCTOR
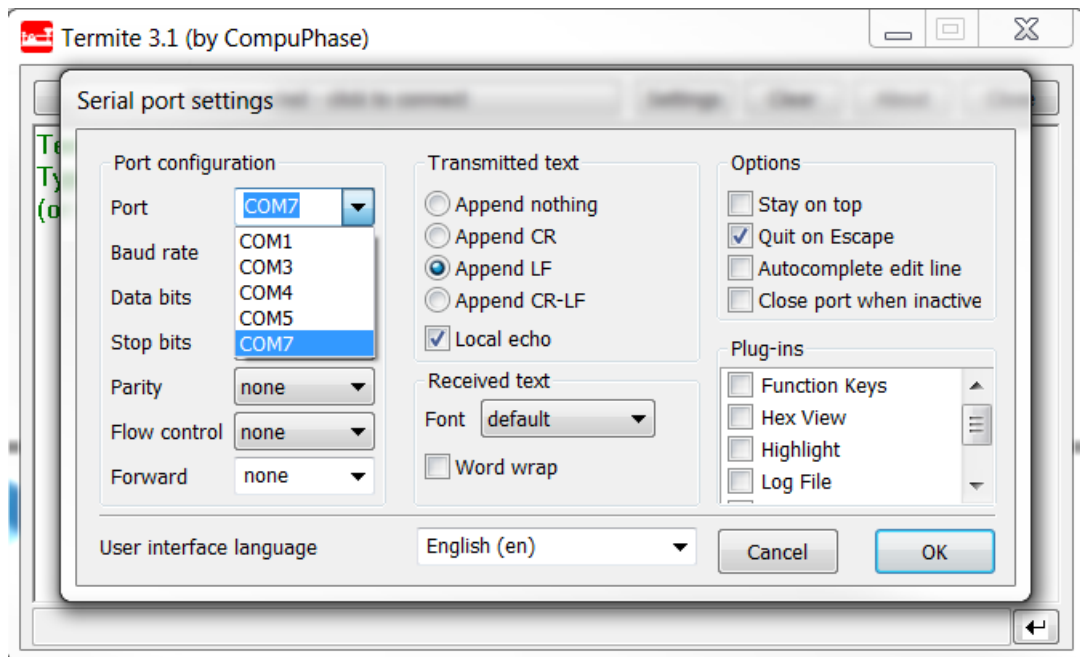
4. Select the correct COM port to connect to the kit.

   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.

NORDIC
SEMICONDUCTOR

7 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.

7.1 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.
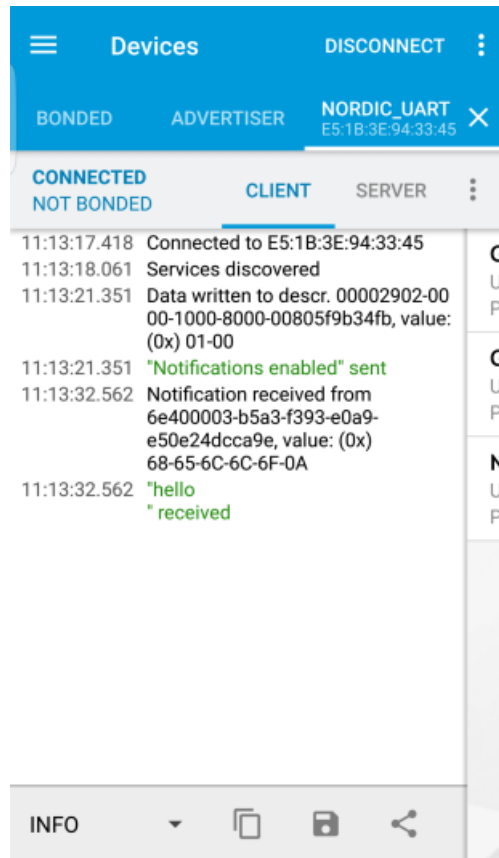
The following procedure assumes that you have programmed the Bluetooth Low Energy Peripheral UART example from the nRF Connect SDK (`ncs/nrf/samples/bluetooth/peripheral_uart`) on your development kit. Steps for testing other examples are similar. See the testing instructions for each example in the nRF Connect 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 `peripheral_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_Service".

6. When connected, tap the options button below the device name and select **Enable services**.
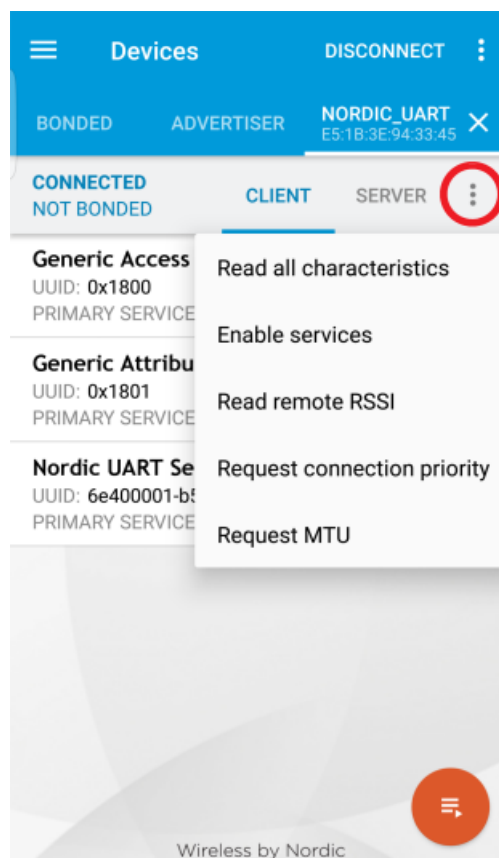
This example communicates over Bluetooth Low Energy using the Nordic UART Service (NUS).

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:

7.2 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 nRF Connect SDK (`ncs/nrf/samples/bluetooth/peripheral_uart`) on your development kit. Steps for testing other examples are similar. See the testing instructions for each example in the nRF Connect 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_Service".

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`.

NORDIC
SEMICONDUCTOR

8 Debugging

To actually see what is happening on the development kit while the application is running, enable debugging in nRF Connect for Visual Studio Code. See Debugging an application for more information.

NORDIC
SEMICONDUCTOR

Glossary

Development Kit (DK)

A hardware development platform used for application development.

Integrated Development Environment (IDE)

A software application that provides facilities for software development.

Real-time operating system (RTOS)

An operating system that reacts to input within a specific period of time. A real-time deadline can be so small that system reaction appears instantaneous.

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.

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.

Recommended reading

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

Development Kit User Guides

nRF5340 DK

Compatibility Matrices

nRF5340 Compatibility Matrix

SDK documentation

nRF Connect SDK documentation

Tools User Guides

nRF Connect extensions for VS Code
nRF Connect Bluetooth Low Energy
nRF Connect Programmer
nRF Command Line Tools

NORDIC
SEMICONDUCTOR

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