

*nRF 5 – Software cheatsheet*

Table des matières

[nRF 5 – Software cheatsheet 0](#_Toc42613906)

[1. nRF52840 Development Kit 2](#_Toc42613907)

[1.1. Key features 2](#_Toc42613908)

[2. nRF5 Software 3](#_Toc42613909)

[2.1. IDE 3](#_Toc42613910)

[2.2. Software Development Kit 4](#_Toc42613911)

[SoftDevices + nRF5 SDK and Application-specific SDKs 4](#_Toc42613912)

[nRF Connect SDK (NCS) – V1.2 5](#_Toc42613913)

[2.3. Development tools 6](#_Toc42613914)

[nRF Connect 6](#_Toc42613915)

[nRF Toolbox 6](#_Toc42613916)

[nRF Command Line Tools 6](#_Toc42613917)

[nRF Sniffer for Bluetooth LE 6](#_Toc42613918)

[2.4. CMAKE & GCC 6](#_Toc42613919)

[3. Installation 7](#_Toc42613920)

[3.1. SoftDevices + nRF5 SDK and Application-specific SDKs 7](#_Toc42613921)

[3.2. nRF Connect SDK (NCS 7](#_Toc42613922)

[3.3. Getting started 8](#_Toc42613923)

[3.4. Modifying a sample application 8](#_Toc42613924)

[Open a project 8](#_Toc42613925)

[Adding files 8](#_Toc42613926)

[Softdevice 8](#_Toc42613927)

[3.5. Configuring your application in Debug 8](#_Toc42613928)

[3.6. UART communication 9](#_Toc42613929)

[4. Acronyms 10](#_Toc42613930)

[5. Bluetooth 11](#_Toc42613931)

[5.1. Software architecture 11](#_Toc42613932)

[5.2. Attribute tables 11](#_Toc42613933)

# nRF52840 Development Kit

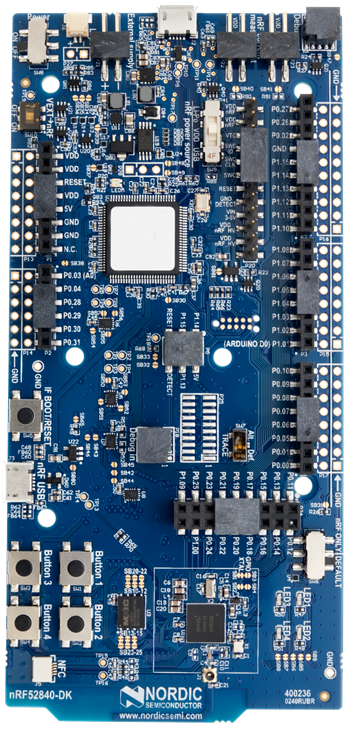


Figure 1: nRF52 DK - pca10056

## Key features

* Supports Bluetooth LE, Bluetooth mesh, NFC, Thread and Zigbee
* User-programmable LEDs(4) and buttons(4)
* 2.4 GHz and NFC antennas
* SWF RF connector for direct RF measurements
* On-board SEGGER J-Link debugger/programmer
* Pins for measuring power consumption
* 1.7-5.0 V supply from USB, external, Li-Po battery or CR2032 coin cell battery
* (Arduino shild compatible 🡪 amelioration, cheap hardware)

# nRF5 Software

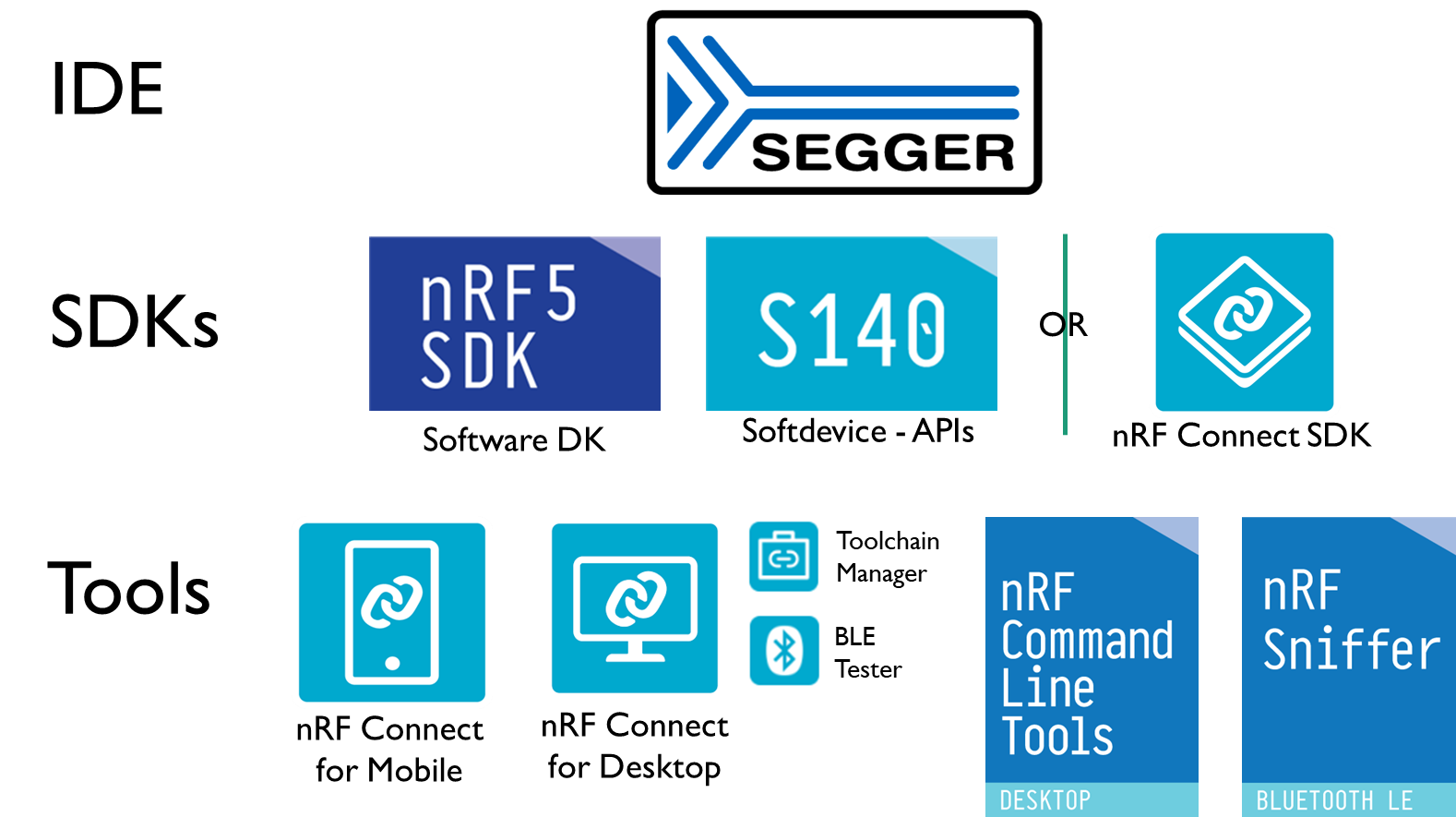


Figure 2 : nRF5 SoftwRE

## IDE

nRF5 SoCs support different IDE, we choose the Segger Embedded Studio because it Free license when working with Nordic Semiconductor. It’s also the default IDE used and develop by Nordic, more example and documentation are available.



Figure 3: Nordic Semiconductor - Supported IDE

Users can choose between the included Clang/LLVM or GCC C/C++ compiler options or use

third party compilers.

For use with nRF Connect SDK, get the SEGGER Embedded Studio Nordic Edition - downloadable from the Toolchain Manager app for nRF Connect for Desktop.

## Software Development Kit

For the nRF52 Series SoCs, Nordic offers the choice between using our standard nRF5 software development kit (SDK) together with our Bluetooth Low Energy (LE) protocol stacks (called SoftDevices) or using the nRF Connect SDK, based around the open source Zephyr RTOS.

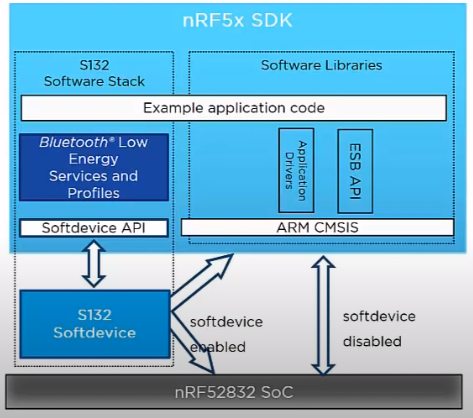
### SoftDevices + nRF5 SDK and Application-specific SDKs

Nordic Semiconductor's most popular solutions today, this solution is very mature and in use on hundreds of millions of products in the market today. This software solution has enormous scope and is qualified to Bluetooth 5.1. There is support for all of the nRF52 Series SoCs. Additionally, there are application-specific SDKs for mesh technologies such as Bluetooth mesh, Thread and Zigbee and HomeKit

#### nRF5 SDK – V15.3

Software development kit, very useful to save time, it includes a lot of varied modules and examples like:

* Bluetooth Low Energy profiles



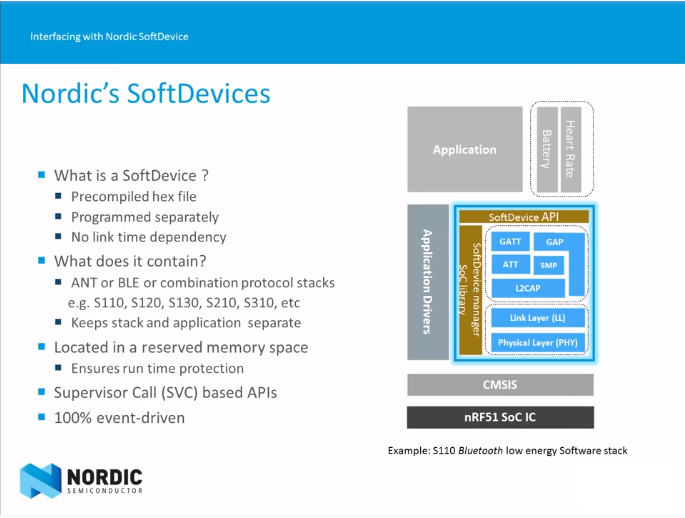
4 - SDK Introduction video

* Device Firmware Upgrade (DFU)
* GATT serializer
* Driver support for all peripherals

#### Softdevice 140

Feature-rich Bluetooth Low Energy protocol stack, Key features:

* Bluetooth 5.1 qualified
* High-throughput 2 Mbps
* Long Range
* Advertising Extensions
* CSA #2
* LE Secure Connections
* Privacy 1.2
* Configurable ATT Table
* Configurable ATT MTU
* Custom UUID
* LE Data Packet Length Extension
* L2CAP connection-oriented channels
* Concurrent multiprotocol support



<https://www.youtube.com/watch?v=tZjlixQPO-Q>

ARM CMSIS: Cortex Microcontroller Software Interface Standard, hardware abstraction layer.

### nRF Connect SDK (NCS) – V1.2

New, open-source and more scalable long-term evolution for development on Nordic devices. From the resource constrained, to the ever more complex high-end solutions. Based around the open source Zephyr RTOS, it uses a specific IDE SEGGER Embedded Studio Nordic Edition.

* Zephyr™ Real-time operating system (RTOS), which is built for connected low power product
* West: swiss-army knife command line tool for Zephyr, West’s built-in commands provide a multiple repository management like Git. Zephyr uses this feature to provide conveniences for building applications, flashing and debugging them, and more.

## Development tools

### nRF Connect

* **nRF Connect for Desktop – V3.4.0**

Contain useful app like:

* + Toolchain Manager: Install and manage tools for nRF Connect SDK (NCS)
  + Bluetooth Low Energy: BLE testing, need one additional nRF DK or Dongue
* **nRF Connect for Mobile – V4.24.1**

Allow to scan and explore your Bluetooth Low Energy devices and communicate with them.

nRF Connect for Mobile supports several Bluetooth **SIG** adopted profiles, as well as the Device Firmware Update profile (**DFU**) from Nordic Semiconductor or Eddystone from Google.

* **nRF Connect for Cloud**

For cellualar IoT application

### nRF Toolbox

The nRF Toolbox is a container app that stores your Nordic Semiconductor apps for Bluetooth® Low Energy in one location.

### nRF Command Line Tools

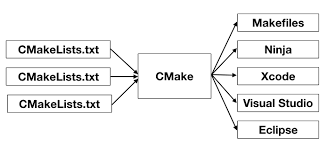
Used for development, programming and debugging of Nordic Semiconductor's nRF52 devices.

* **nrfjprog** executable - tool for programming through SEGGER J-LINK programmers and debuggers
* **mergehex** executable - enables you to combine up to three .HEX files into one single file
* **nrfjprog DLL** - a DLL that exports functions for programming and controlling nRF51, nRF52, nRF53 and nRF91 Series devices and lets developers create their own development tools using the DLLs API
* SEGGER J-Link software and documentation pack (only included in the Windows installer)

### nRF Sniffer for Bluetooth LE

Useful tool for debugging and learning about Bluetooth Low Energy applications, allows near real-time display of Bluetooth LE packets. Equipment needed nRF5x DK or Dongle.

## How to Build a CMake-Based ProjectCMAKE & GCC

Used by the nRF Connect SDK

<https://stackoverflow.com/questions/39761924/understanding-roles-of-cmake-make-and-gcc>

# Installation

Nordic Tools and download: <https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug_getting_started%2FUG%2Fcommon%2Fnordic_tools.html&cp=1_0_1>

## SoftDevices + nRF5 SDK and Application-specific SDKs

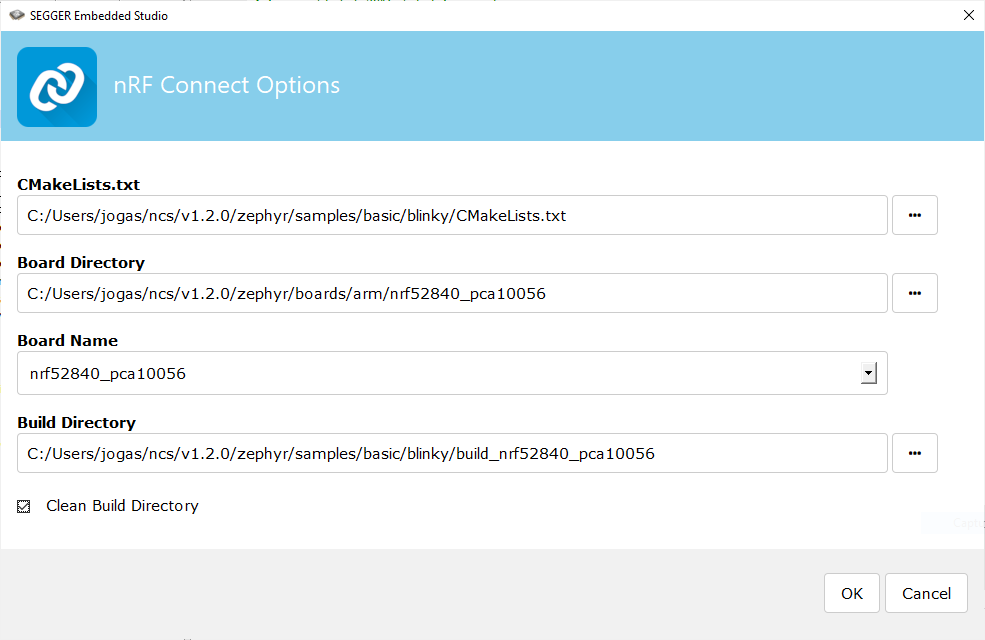
1. Download the nRF5 SDK with all the examples
2. Download the softdevice precompiled (.hex) file
3. Install Segger Embedded Studio **for ARM**
   1. Open any xxx\_pca10056**.emProject** to open one project

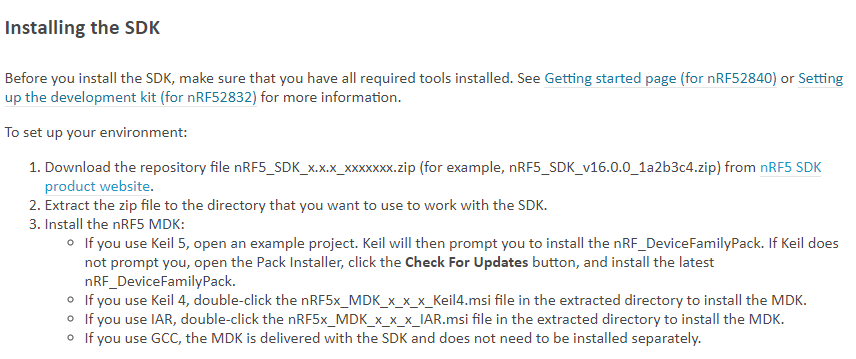
## nRF Connect SDK (NCS

1. Install nRF Connect for Desctop
2. Install the Toolchain Manager app

🡪 Install nRF Connect SDK and Segger Embedded Studio **Nordic Edition**

1. Checkbuild configuration: File>>Open nRF Connect SDK Project





## Getting started

<https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug_getting_started%2FUG%2Fgs%2Fdevelop_sw.html&cp=1_0_2>

Sample and application: <http://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/examples.html>

Drag and drop a **file.HEX** on the usb J-LINK peripheral to Flash the MCU.

## Debug

<https://www.youtube.com/watch?v=uP8RYgYGRvI&hd=1&feature=youtu.be&t=254>

### Custom Services error

1. For newer SDKs, go to sdk\_config.h and set "#define NRF\_SDH\_BLE\_VS\_UUID\_COUNT 1"

2. You need to adjust the RAM size and start address. Simply add 16 bytes to the start and subtract 16 from the size per each new UUID you add.

For example if:

* RAM\_START=0x20002218
* RAM\_SIZE=0xdde8

the new setting would be

* RAM\_START=0x20002218 + 0x10 =0x20002228
* RAM\_SIZE=0xdde8 -0x10 =0xDDD8

## CMSIS Configuration Wizard

<https://www.youtube.com/watch?v=b0MxWaAjMco>

<https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.sdk5.v14.1.0%2Fsdk_config.html>

## Modifying a sample application

<https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/gs_modifying.html>

### Open a project

Search the file: softdevice/ses/xxx\_pca10056.emProject

* F7: Build the project -> create the zypher.elf file
* F5: Debug/Go

Open it directly form the Toolchain Manager

### Adding files

* Modify *CMakeList.txt* directly with Segger
* Use tags:

*# NORDIC SDK APP START*

target\_sources(app PRIVATE src/main.c)

*# NORDIC SDK APP END*

### Softdevice

Library for Bluetooth and ANT, located in folder components\softdevice\SoftDevice\hex (precompiled)

## Configuring your application in Debug

<https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug_getting_started%2FUG%2Fgs%2Fdevelop_sw.html&cp=1_0_2>

set CONFIG\_DEBUG\_OPTIMIZATIONS to y

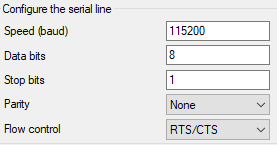
Default configuration for:

* A Library: Kconfig (& prj.conf permanently changes)
* A Board: \*\_defconfig & Kconfig.defconfig

(Write your devicetree n the Zephyr documentation for more information)

<https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/zephyr/application/index.html#application-kconfig>

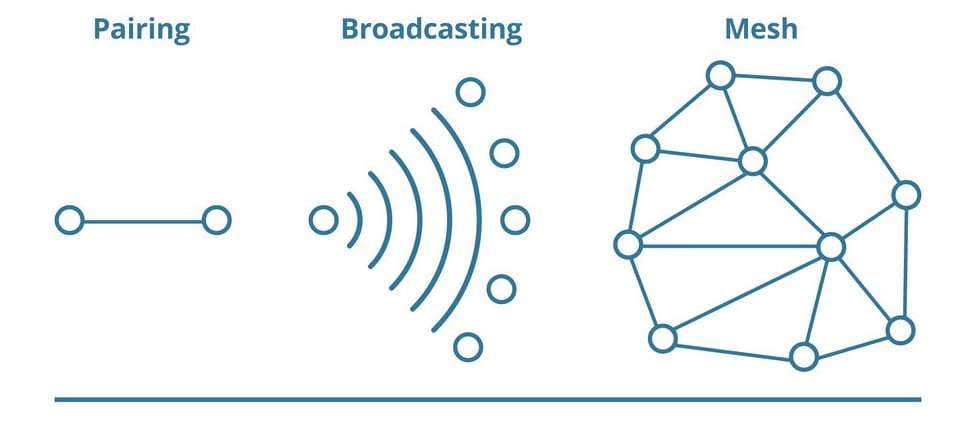
## UART communication



# Acronyms

* MCU: microcontroller unit
* SoC: System on Chips = MCU
* SPI : Serial Peripheral Interface
* API: application programming interface
* IDE: Integrated Development Environment
* SDK: Software Development Kits (SDKs) are your starting point for software development on the nRF51 and nRF52 Series. They contain source code libraries and example applications covering wireless functions, libraries for all peripherals, bootloaders, Wired and OTA firmware upgrades, RTOS examples, serialization libraries and more. Some of the specialized SDKs are installed on top of the generic nRF5 SDK and extend its functionality.
* Nordic Semiconductor provides Software Development Kits to facilitate firmware development for different devices and applications. The SDKs contain examples that are tailored to run on Nordic Semiconductor's Development Kits.
* MDK Microcontroller Development Kit
* GNU: **G**NU’s **N**ot **U**NIX, operating system
* NCS: nRF Connect SDK
* DLL: Dynamic-link library, Modules used by a program are loaded from individual shared library into memory at load time or runtime, rather than being copied by a linker when it creates a single monolithic executable file for the program.
* CMSIS: Cortex Microcontroller Software Interface Standard, hardware abstraction layer.
* <https://infocenter.nordicsemi.com/index.jsp?topic=%2Fug_getting_started%2FUG%2Fgs%2Fdevelop_sw.html&cp=1_0_2>

# Bluetooth

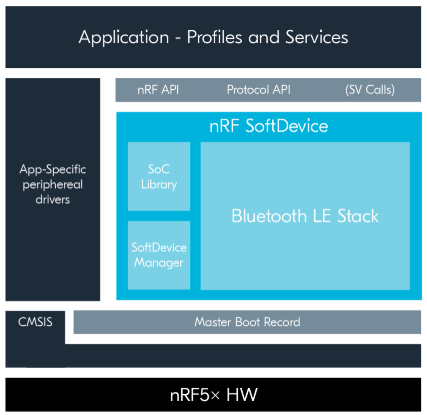


Develop with Bluetooth: <https://www.bluetooth.com/develop-with-bluetooth/>

Specifications: <https://www.bluetooth.com/specifications/>

## Software architecture

https://www.nordicsemi.com/Software-and-tools/Software/Bluetooth-Software



## Attribute tables