



nRF51 Development Kit

Developing with the MDK-ARM Microcontroller Development Kit

User Guide v1.0

1 Introduction

The nRF51 Development Kit combined with the nRF51 SDK forms a complete solution for product development based on nRF51 series chips.

The nRF51 Development Kit is fitted with the nRF51422 chip, which is a powerful, highly flexible multi-protocol System on Chip (SoC) ideally suited for ANT™/ANT+, *Bluetooth*® low energy (BLE), and 2.4 GHz proprietary ultra-low power wireless applications.

And, if you are not using the ANT protocol, you can use the nRF51822 chip in your end product - without any changes needed to your code.

1.1 Key features

The nRF51 DK board has the following key features:

- nRF51422 flash based ANT/ANT+, *Bluetooth* low energy SoC solution
- 2.4 GHz compatible with nRF24L devices
- Buttons and LEDs for user interaction
- I/O interface for Arduino form factor plug-in modules
- SEGGER J-Link OB Debugger with debug out functionality
- Virtual COM Port interface via UART
- Drag and drop Mass Storage Device (MSD) programming
- mbed enabled

1.2 Required tools

Below is a list of hardware and software tools that is required if you plan to explore all the features on this development kit. All the tools may not be required for all use cases.

Nordic Tools	Description
nRFgo Studio	<p>nRFgo Studio is our tool to program and configure devices. It supports the programming of nRF51 SoftDevices, applications, and bootloaders. The different programming modes are available on individual tabs in the nRF51 programming module. Studio is used for the following:</p> <ul style="list-style-type: none"> • <i>Bluetooth</i> Direct Test Mode (DTM) testing • Erasing flash memory • Programming a SoftDevice • Programming an application • Programming the bootloader <p>nRFgo Studio supports programming of SEGGER J-Link based nRF51 devices. For more information, see the help in nRFgo Studio.</p>
nRF51 Tools	<p>nRF51 Tools is a package that contains JLinkARM, JLink CDC, nRFjprog, and mergehex. The nRFjprog is a command line tool for programming nRF51 series chips. It is also useful in a production setup. nRF51 Tools will be installed together with nRFgo Studio.</p>
nRF51 SDK	<p>The nRF51 Software Development Kit (SDK) provides source code of examples and libraries forming the base of your application development. The nRF51 SDK includes:</p> <ul style="list-style-type: none"> • Example code • ANT profile examples • <i>Bluetooth</i> profile examples • Drivers • Libraries <p>For more information, see the documentation packaged with the nRF51 SDK.</p>
Master Control Panel	<p>The Master Control Panel is the software tool that is used with the nRF51 Dongle (PCA10031) to act as a <i>Bluetooth</i> low energy peer device. You can test your application's wireless connection with this tool. The Master Control Panel supports programming of SEGGER J-Link based nRF51 devices. For more information, see the help files in the Master Control Panel.</p>
nRF Master Control Panel for Android 4.3	<p>nRF Master Control Panel for Android 4.3 is a powerful generic tool that allows you to scan and explore your <i>Bluetooth</i> Smart devices and communicate with them on an Android phone. MCP supports a number of <i>Bluetooth</i> SIG adopted profiles including the Device Firmware Update (DFU) profile from Nordic Semiconductor.</p>
S110 SoftDevice	<p><i>Bluetooth</i> low energy Peripheral/Broadcaster protocol stack. For more information, see the <i>S110 nRF51822 SoftDevice Specification</i> and the nRF51 SDK documentation.</p>
S120 SoftDevice	<p><i>Bluetooth</i> low energy Central protocol stack solution supporting up to eight simultaneous Central role connections. For more information, see the <i>S120 nRF51822 SoftDevice Specification</i> and the nRF51 SDK documentation.</p>
S130 SoftDevice	<p><i>Bluetooth</i> Smart concurrent multi-link protocol stack solution supporting simultaneous Central/Peripheral/Broadcaster/Observer role connections. For more information, see the <i>S130 nRF51822 SoftDevice Specification</i> and the nRF51 SDK documentation.</p>
S210 SoftDevice	<p>ANT protocol stack. For more information, see the <i>S210 nRF51422 SoftDevice Specification</i> and the nRF51 SDK documentation.</p>
S310 SoftDevice	<p>ANT and <i>Bluetooth</i> low energy Peripheral controller and host multiprotocol stack. For more information, see the <i>S310 nRF51422 SoftDevice Specification</i> and the nRF51 SDK documentation.</p>

Third party tools	Description
Keil MDK-ARM Development Kit	Keil MDK-ARM Development Kit is a development environment specifically designed for microcontroller applications that lets you develop using the nRF51 SDK application and example files.
SEGGER J-Link Software	The J-Link software is required to debug using the J-Link hardware that is packaged with this development kit.
ANTware II	ANTware is an application used for the control of ANT wireless devices. It is an excellent tool for first time ANT developers to explore the capabilities of ANT as a low power wireless solution, and for experienced users to easily setup and monitor advanced ANT networks. ANTware II improves upon past versions with a slick new interface, streamlined functions, and a variety of new features.

1.3 Documentation

Below is a list of the core documentation for the nRF51 Series and the nRF51x22 chip.

Note: See our website for additional documentation such as Application Notes and White Papers.

Document	Description
<i>nRF51 Series Reference Manual</i>	The <i>nRF51 Series Reference Manual</i> is a functional description of all the modules and peripherals supported by the nRF51 series.
<i>nRF51x22 Product Specification</i>	The <i>nRF51x22 Product Specification</i> contains all specifications specific to the chip.
<i>S110 nRF51822 SoftDevice Specification</i>	The <i>S110 nRF51822 SoftDevice Specification</i> contains information about the SoftDevice features and performance.
<i>S120 nRF51822 SoftDevice Specification</i>	The <i>S120 nRF51822 SoftDevice Specification</i> contains information about the SoftDevice features and performance.
<i>S130 nRF51822 SoftDevice Specification</i>	The <i>S130 nRF51822 SoftDevice Specification</i> contains information about the SoftDevice features and performance.
<i>S210 nRF51422 SoftDevice Specification</i>	The <i>S210 nRF51422 SoftDevice Specification</i> contains information about the SoftDevice features and performance.
<i>S310 nRF51422 SoftDevice Specification</i>	The <i>S310 nRF51422 SoftDevice Specification</i> contains information about the SoftDevice features and performance.
<i>nRF51 SDK Documentation</i>	The nRF51 SDK documentation includes user guides, descriptions, and reference material to help you understand the protocols, examples, and other components of the SDK.
<i>nRF51x22 Product Anomaly Notification</i>	The <i>nRF51x22 Product Anomaly Notification</i> - list of anomalies relevant for the chip.
<i>ANT Message Protocol and Usage</i>	The <i>ANT Message Protocol and Usage</i> document describes the ANT protocol in detail and contains the fundamental knowledge you need in order to develop successfully with ANT.
nRF51 Development Kit Hardware Files	The nRF51 Development Kit includes firmware source code, documentation, hardware schematics, and layout files. Included are the following files: <ul style="list-style-type: none"> • Altium Designer files • Schematics • PCB layout files • Production files <ul style="list-style-type: none"> • Assembly drawings • Drill files • Gerber files • Pick and Place files • Bill of Materials

1.4 Development Kit release notes

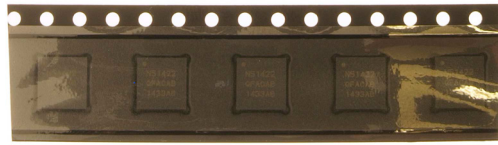
Date	Kit version	Description
October 2014	1.0	<ul style="list-style-type: none">• First release.

2 Kit content

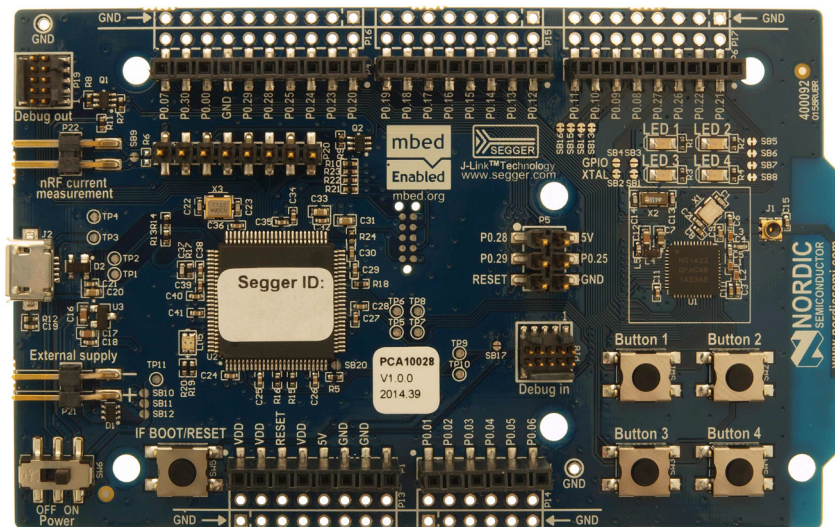
In addition to hardware, the nRF51 Development Kit consists of firmware source code, documentation, hardware schematics, and layout files which are available from www.nordicsemi.com.



1 x 3V CR2032
Lithium battery



5 x nRF51422 samples



1 x nRF51 Development Kit board (PCA10028)

Figure 1 nRF51 Development Kit content

3 Getting started

This section shows you how to get access to the tools, libraries, and documentation.

Connect your nRF51 Development Kit to a computer.

1. Connect your nRF51 DK board to a computer with a USB cable.
2. The status light (**LD5**) will come on, indicating it has power.
3. After a few seconds, the computer will recognize the nRF51 DK board as a standard USB drive.

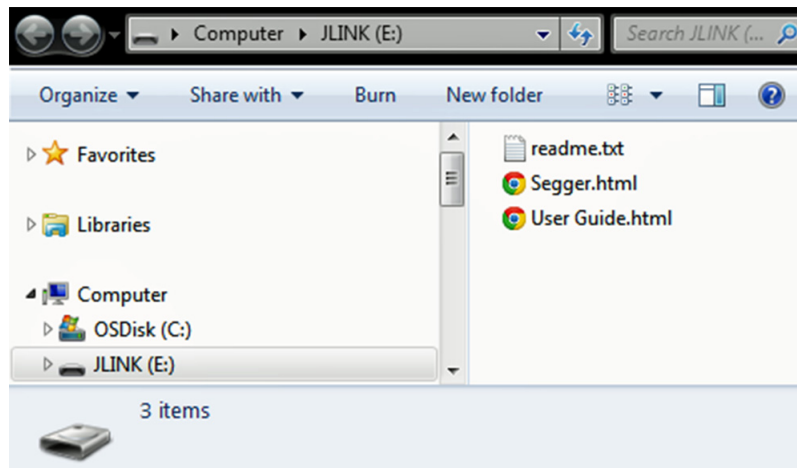


Figure 2 Windows example

Get started with the toolchain and examples.

1. Download and install nRFgo Studio (which includes the nRF Tools package; JLinkARM, JLink CDC, nRFjprog, and mergehex) from www.nordicsemi.com.
2. Download and install the latest Keil MDK-ARM from www.keil.com/arm.
3. Download and install the latest nRF51 SDK found on <http://developer.nordicsemi.com>.
4. Read the information in the Readme.txt file that is installed together with the SDK.

After the installation, the SDK documentation is found here:

Online: http://developer.nordicsemi.com/nRF51_SDK/doc/index.html.

Offline: <keil_location>/ARM/Pack/NordicSemiconductor/nRF_Examples/<version>/documentation

4 Interface MCU

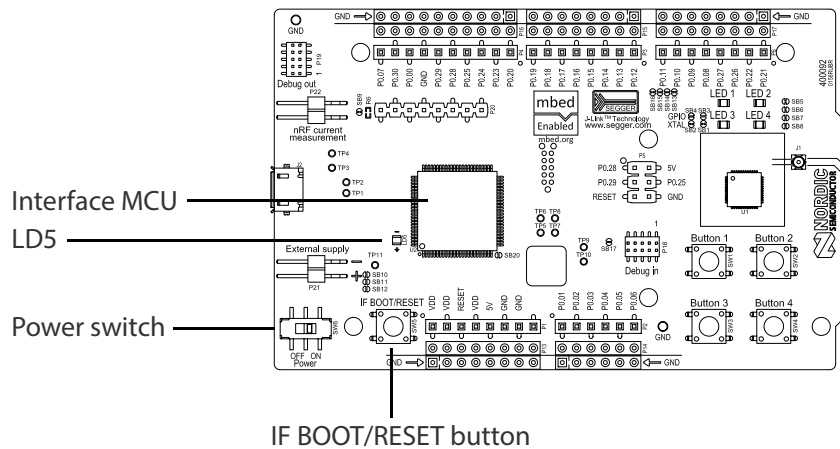


Figure 3 Interface MCU

4.1 IF Boot/Reset button

The nRF51 DK board is equipped with a boot/reset button (**SW5**). This button is connected to the interface MCU on the board and have two functions:

- Reset button for the nRF device.
- Enter boot loader mode of the interface MCU.

During normal operation the button will function as a reset button for the nRF device. To enter boot loader mode of the Interface MCU, the button must be pressed while the board is powered until the LED **LD5** starts to blink. This is done by pressing the reset button and power cycle the board, either by disconnecting and reconnecting the USB cable or toggle the power switch (**SW6**).

4.2 Virtual COM port

The on-board Interface MCU features a Virtual COM port via UART.

- Flexible baudrate setting up to 1 Mbps
- Dynamic Hardware Flow Control (HWFC) handling
- Tri-stated UART lines while no terminal is connected

Table 1 shows an overview of the UART connections on nRF51422 and the interface MCU.

nRF51422		Interface MCU
Default GPIO	UART	UART
P0.08	RTS	CTS
P0.09	TXD	RXD
P0.10	CTS	RTS
P0.11	RXD	TXD

Table 1 Relationship of UART connections on nRF51422 and Interface MCU

The UART signals are routed directly to the Interface MCU. The UART pins connected to the Interface MCU is tri-stated when no terminal is connected to the Virtual COM port on the computer.

Note: The terminal used must send a DTR signal in order to configure the UART Interface MCU pins.

The P0.08 (RTS) and P0.10 (CTS) can be used freely when HWFC is disabled on the nRF51422.

4.3 Interface MCU Firmware

The on board Interface MCU is factory programmed with an mbed compliant bootloader, this feature enables the ability to swap interface FW between the factory preloaded SEGGER J-Link OB and the nRF51 mbed interface FW, see **section 4.1 “IF Boot/Reset button”** on page 8 on how to enter the bootloader.

To swap Interface MCU FW, simply drag the Interface image (.bin) into the mounted bootloader drive on the connected computer and power cycle the board.

Both the nRF51 mbed interface FW and the J-Link OB image can be downloaded from www.nordicsemi.com.

Note: If you have swapped to the mbed image and want to revert back to the J-Link image, download the latest SEGGER J-Link software from www.segger.com and open a debug session to update to the latest J-Link OB firmware version.

Note: The J-Link serial number is linked to the Interface MCU and will not change even when swapping the Interface MCU FW, so it can be useful to write the serial number on a sticker on the board.

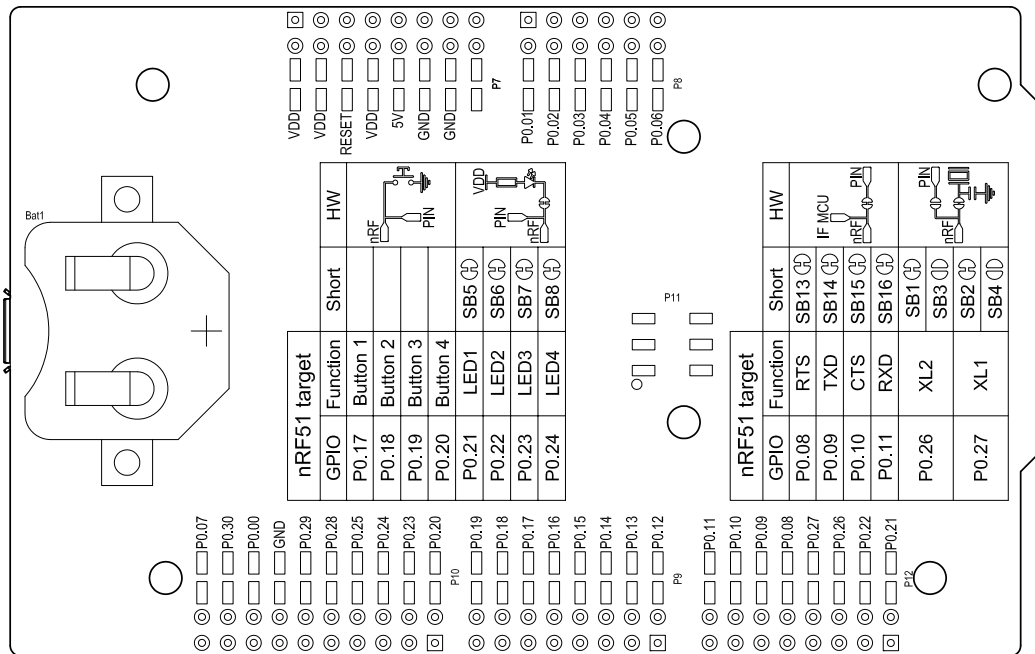


Figure 5 nRF51 DK board bottom

5.2 Block diagram

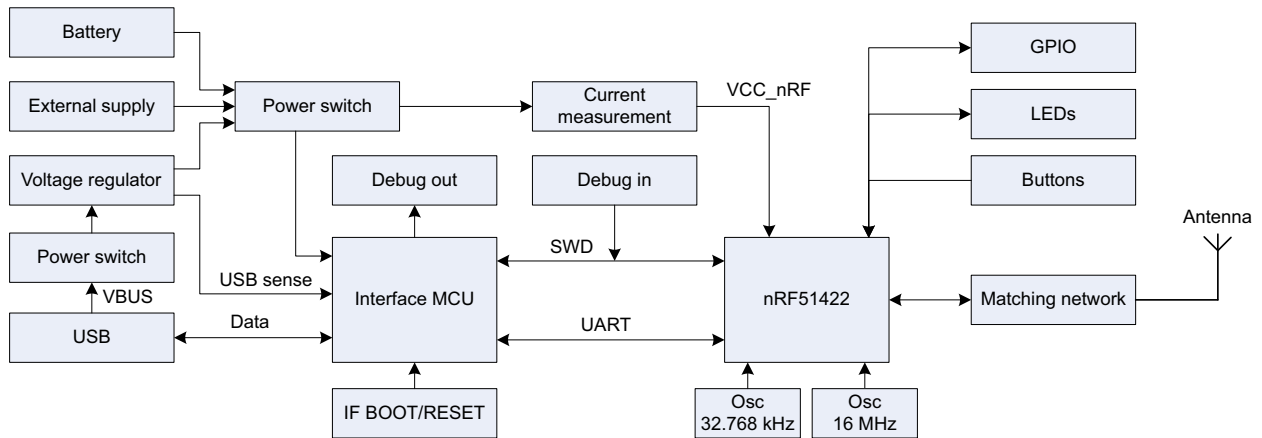


Figure 6 nRF51 DK board block diagram

5.3 Power supply

The nRF51 DK board has several power options:

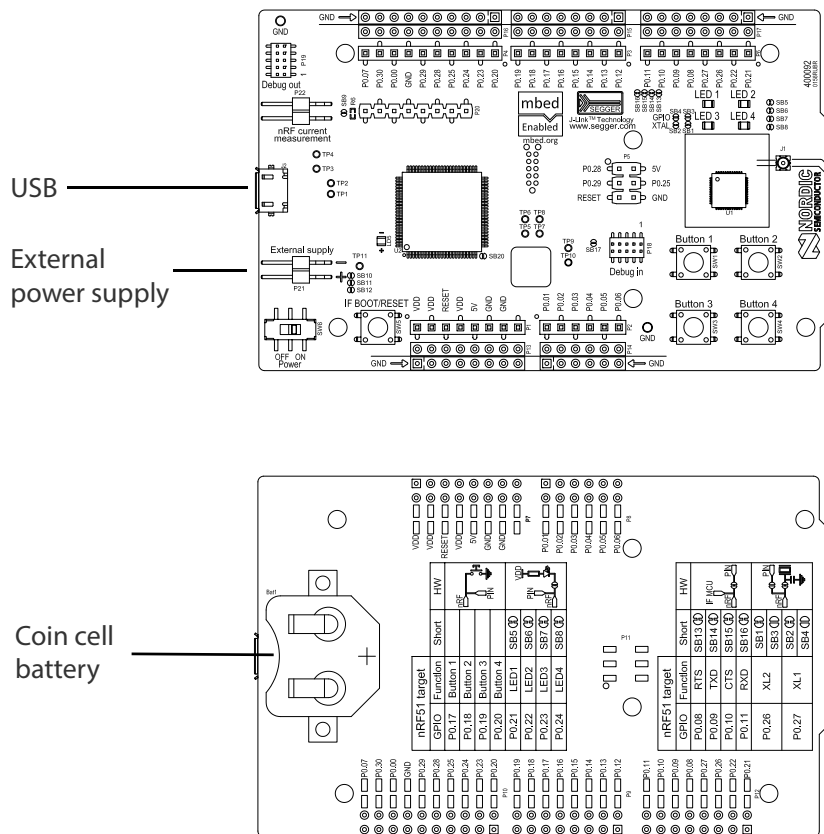


Figure 7 Power supply options

The 5 V from the USB is regulated down to 3.3 V through an on-board voltage regulator. The battery and external power supply are not regulated. The power sources are routed through a set of diodes (D1A, D1B, and D1C) for reverse voltage protection, where the circuit is supplied from the source with the highest voltage.

Note: When not USB powered, the Interface MCU is in dormant state and will draw an additional current of $\sim 20 \mu\text{A}$ in order to maintain the reset button functionality. This will affect board current consumption, but will not affect the nRF51 current measurements as described in *Section 5.7 “Measuring current”* on page 18.



Power source	Protection bypass	Voltage level
USB	SB10	3.3 V
Coin-cell battery	SB11	Battery
External supply	SB12	1.8 V - 3.6 V

Table 2 Protection diode bypass solder bridges



Page 13

5.4 Connector interface

Access to the nRF51422 GPIOs is available at connectors **P2**, **P3**, **P4**, **P5**, and **P6** on the nRF51 DK board. In addition there is access to ground and power on the **P1** connector.

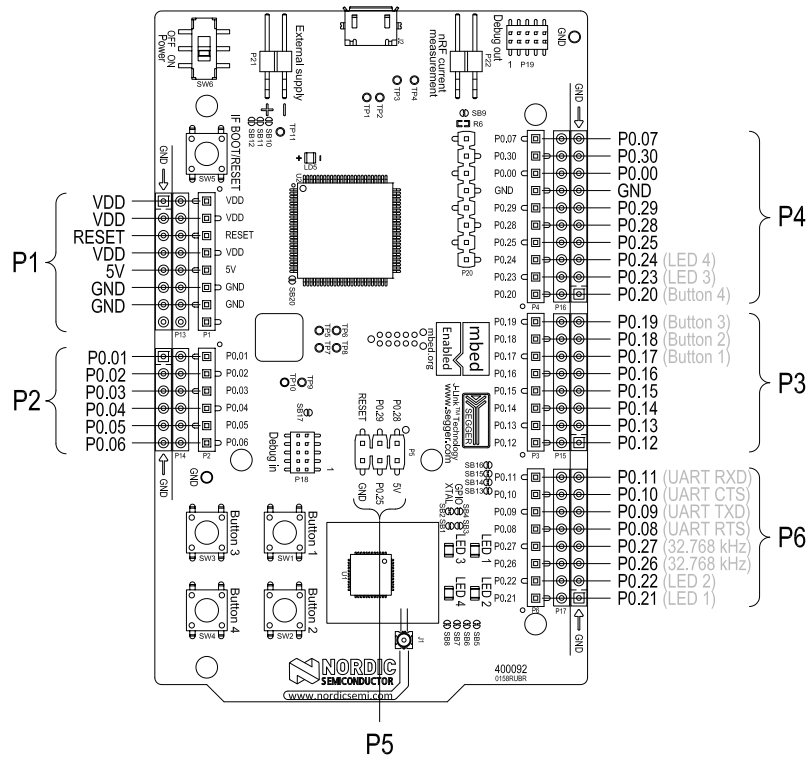


Figure 10 nRF51 DK board connectors

The signals are also available on bottom side connectors **P7**, **P8**, **P9**, **P10**, **P11**, and **P12**. By mounting pin lists on the connector footprints, the nRF51 DK board can be used as a shield for Arduino motherboards or other boards that follows the Arduino standard.

For easy access to GPIO, power, and ground, the signals can also be found on the through-hole connectors **P13-P17**.

Note: Some pins have default settings.

- P0.08, P0.09, P0.10, and P0.11 are by default used by the UART connected to the Interface MCU. See **Section 4.2 “Virtual COM port”** on page 9 for more information.
- P0.17 - P0.24 are by default connected to the buttons and LED. See **Section 5.5 “Buttons and LEDs”** on page 16 for more information.
- P0.26 and P0.27 are by default used for the 32 kHz crystal and are not available on the connectors. See **Section 5.6 “32.768 kHz crystal”** on page 17 for more information.

When the nRF51 DK board is used as a shield together with an Arduino standard motherboard, the Arduino signals is routed like shown in **Figure 11**.

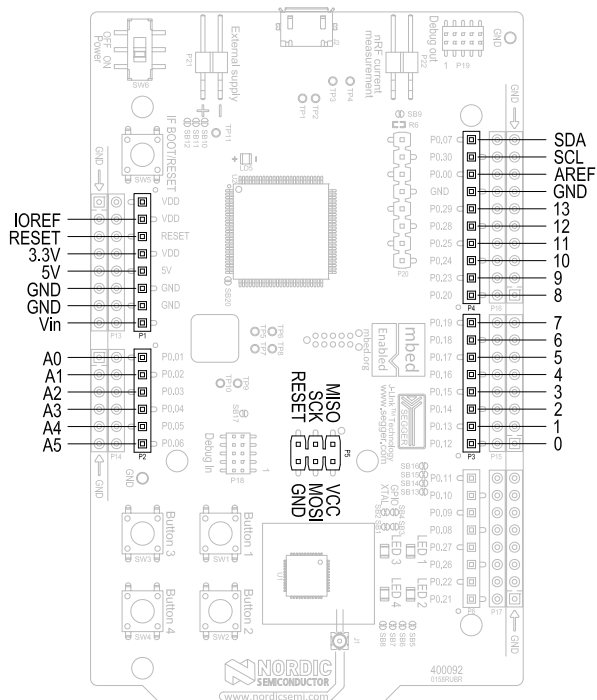


Figure 11 Arduino signals routing on the nRF51 DK board

5.5 Buttons and LEDs

The four buttons and four LEDs on nRF51 DK board are connected to dedicated I/Os on the nRF51422 chip. The connections are shown in **Table 3**.

Part	GPIO	Short
Button 1	P0.17	-
Button 2	P0.18	-
Button 3	P0.19	-
Button 4	P0.20	-
LED 1	P0.21	SB5
LED 2	P0.22	SB6
LED 3	P0.23	SB7
LED 4	P0.24	SB8

Table 3 Button and LED connection

If GPIO P0.21 - P0.24 are needed elsewhere, the LEDs can be disconnected by cutting the short on SB5-SB8, see **Figure 12**.

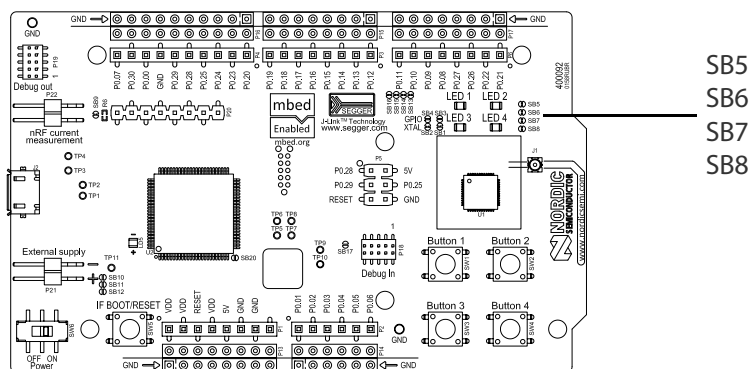


Figure 12 Disconnecting the LEDs

The buttons are active low meaning the input will be connected to ground when the button is activated. The buttons have no external pull-up resistor, so to use the buttons the P0.17 - P0.20 pins must be configured as an input with internal pull-up resistor.

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

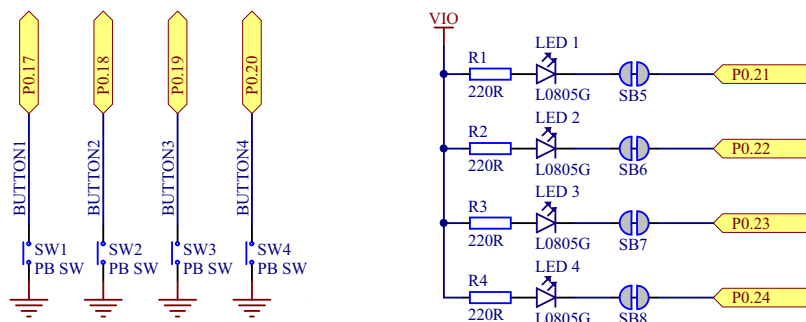


Figure 13 Button and LED configuration

5.6 32.768 kHz crystal

nRF51422 can use an optional 32.768 kHz crystal (X2) for higher accuracy and lower average power consumption. On the DK board, P0.26 and P0.27 are by default used for the 32.768 kHz crystal and are not available as a GPIO on the connectors.

Note: When using ANT/ANT+, the 32.768 kHz crystal (X2) is required for correct operation.

If P0.26 and P0.27 are needed as normal I/Os the 32.768 kHz crystal can be disconnected and the GPIO routed to the connectors. Cut the shorting track on SB1 and SB2, and solder SB3 and SB4. See **Figure 14** for reference.

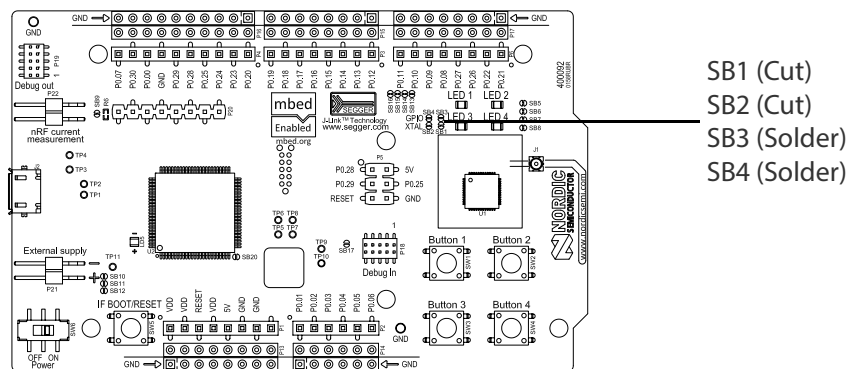


Figure 14 Disconnecting 32.768 kHz crystal and connecting P0.26 and P0.27 to the connectors

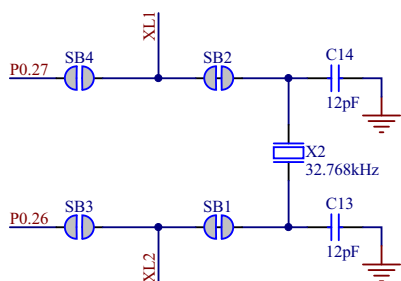


Figure 15 32.768 kHz crystal and SB1 to SB4 schematic

5.7 Measuring current

The current drawn by the nRF51422 device can be monitored on the nRF51 DK board. To measure the current, you must first prepare the board by cutting the shorting of solder bridge SB9.

There are two ways of measuring the current consumption:

1. Connect an ampere-meter between the pins of connector P22. This will monitor the current directly.

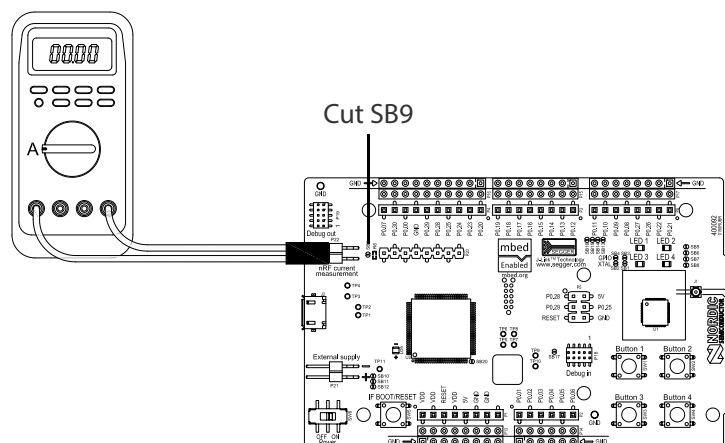


Figure 16 Current measurement with ampere-meter

2. Mount a resistor on the footprint for R6. The resistor should not be larger than $10\ \Omega$. Connect an oscilloscope in differential mode or similar with two probes on the pins of the P1 connector and measure the voltage drop. The voltage drop will be proportional with the current consumption. For example, if a $10\ \Omega$ resistor is chosen, $10\ \text{mV}$ equals $1\ \text{mA}$.

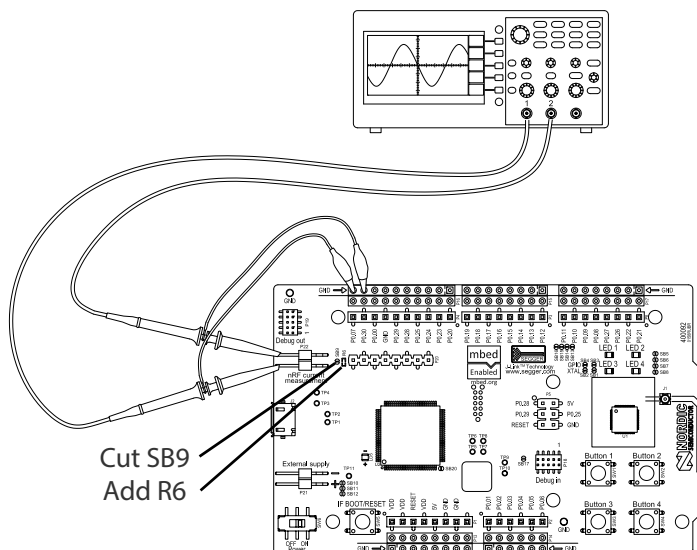


Figure 17 Current measurement with oscilloscope

Note: The current measurements will become unreliable when a Serial terminal is connected to the Virtual COM port.

5.8 RF measurements

The nRF51 DK board is equipped with a small size coaxial connector for conducted measurements of the RF signal (**J1**). The connector is of SWF type from Murata (part no. MM8130-2600) with an internal switch. By default, when there is no cable attached, the RF signal is routed to the on-board PCB trace antenna.

A test probe is available from Murata, part no. MXHS83QE3000, which has a standard SMA connection on the other end for connecting instruments. When connecting the test probe, the internal switch in the SWF connector will disconnect the PCB antenna and connect the RF signal from the nRF51 device to the test probe.

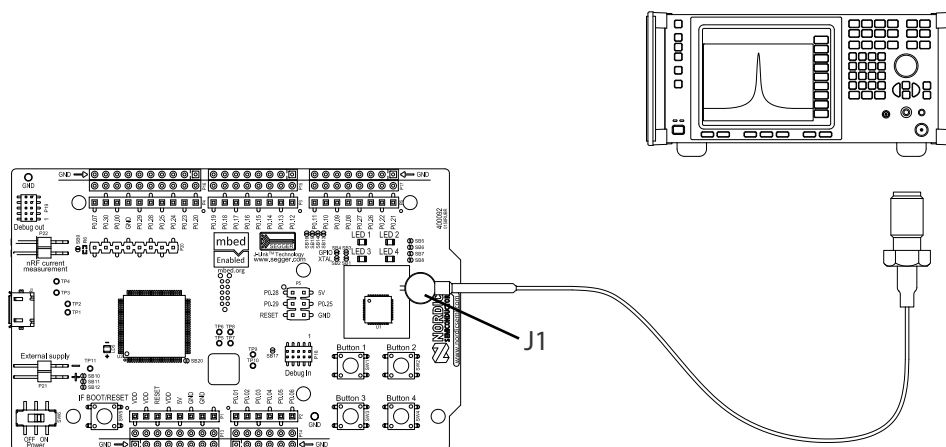


Figure 18 Connecting a spectrum analyzer

The connector and test probe will add loss to the RF signal which should be taken into account when doing measurements, see **Table 4**.

Frequency (MHz)	Loss (dB)
2440	1.0
4880	1.7
7320	2.6

Table 4 Typically loss in connector and test probe

5.9 Debug input

The debug input, Debug in (**P18**) connector, enables connection of external debuggers in order to debug while running on battery or external power supply.

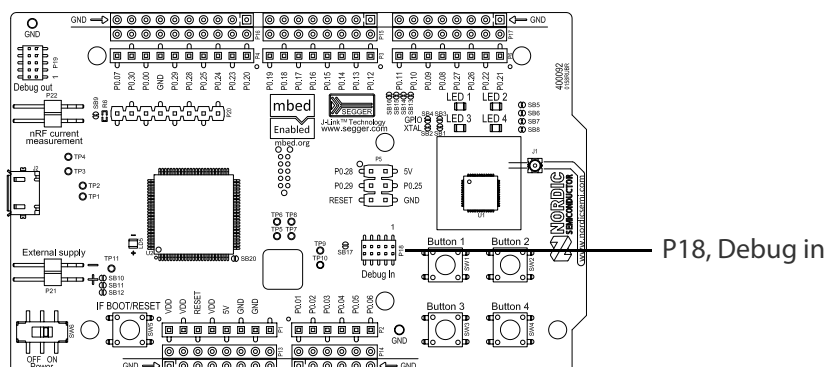


Figure 19 Debug input connector

5.10 Debug output

The nRF51 DK board supports programming and debugging of external boards. To debug an external board, connect to the Debug out (**P19**) connector with a 10 pin cable.

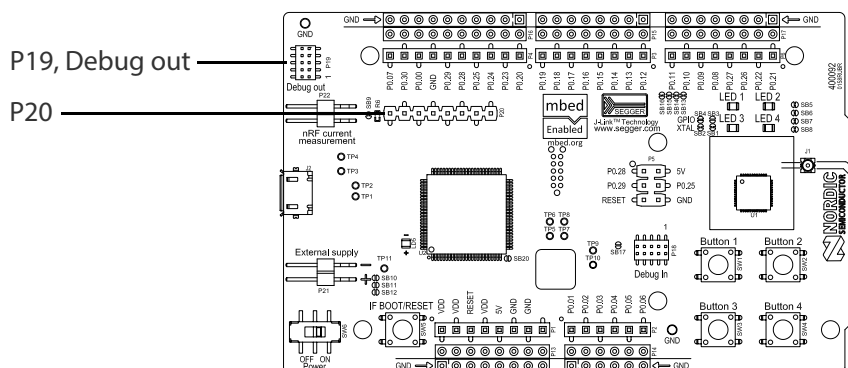


Figure 20 Debug output connector

When the external board is powered, the interface MCU will detect the supply voltage of the board and program/debug the target chip on external board instead of the on-board nRF51422.

Note: The voltage supported by external debugging/programming is 3.0 V.

The **P20** also features a debug out connection in order to program shield mounted targets, as for the Debug out **P19** header the Interface MCU will detect the supply voltage on the mounted shield and program/debug the shield target.

If the Interface MCU detects target power on both **P19** and **P20** it will default to program/debug the target connected to **P19**

Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

Life support applications

Nordic Semiconductor's products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

Contact details

For your nearest distributor, please visit <http://www.nordicsemi.com>.

Information regarding product updates, downloads, and technical support can be accessed through your My Page account on our homepage.

Main office: Otto Nielsens veg 12
7052 Trondheim
Norway

Phone: +47 72 89 89 00

Fax: +47 72 89 89 89

Mailing address: Nordic Semiconductor
P.O. Box 2336
7004 Trondheim
Norway



Revision history

Date	Version	Description
October 2014	1.0	• First release.

ARM statement

Keil, µVision, and Cortex are trademarks of ARM Limited. All other brands or product names are the property of their respective holders.