



**nRF52840 Preview Development Kit v0.9.x**  
**User Guide**  
**v1.0**

# Contents

- Revision history..... 4**
- Chapter 1: Introduction..... 5**
- Chapter 2: Setting up the development kit..... 7**
- Chapter 3: Software tools..... 8**
- Chapter 4: Start developing..... 9**
- Chapter 5: Interface MCU..... 10**
  - 5.1 IF Boot/Reset button..... 10
  - 5.2 Virtual COM port..... 10
    - 5.2.1 Dynamic HWFC handling..... 11
  - 5.3 Interface MCU firmware..... 11
  - 5.4 MSD..... 12
- Chapter 6: Hardware description..... 13**
  - 6.1 Hardware drawings..... 13
  - 6.2 Block diagram..... 13
  - 6.3 Power supply..... 14
    - 6.3.1 5 V power sources..... 15
    - 6.3.2 VDD power sources..... 16
    - 6.3.3 Interface MCU power..... 18
    - 6.3.4 nRF52840 power source..... 19
    - 6.3.5 nRF52840 direct supply..... 20
  - 6.4 Operating modes..... 21
    - 6.4.1 USB detect..... 21
    - 6.4.2 nRF only mode..... 21
    - 6.4.3 Signal switches..... 22
  - 6.5 External memory..... 24
  - 6.6 Connector interface..... 25
  - 6.7 Buttons and LEDs..... 27
  - 6.8 32.768 kHz crystal..... 28
  - 6.9 Measuring current..... 29
    - 6.9.1 Preparing the development kit board..... 30
    - 6.9.2 Using an oscilloscope for current profile measurement..... 31
    - 6.9.3 Using an ampere-meter for current measurement..... 31
  - 6.10 RF measurements..... 32
  - 6.11 Debug input and Trace..... 33
  - 6.12 Debug output..... 34
  - 6.13 NFC antenna interface..... 35
  - 6.14 Extra op-amp..... 36
  - 6.15 Solder bridge configuration..... 37

**Liability disclaimer.....40**

# Revision history

**Table 1: Revision history**

Date	Version	Description
December 2016	1.0	First release.

---

# Chapter 1

## Introduction

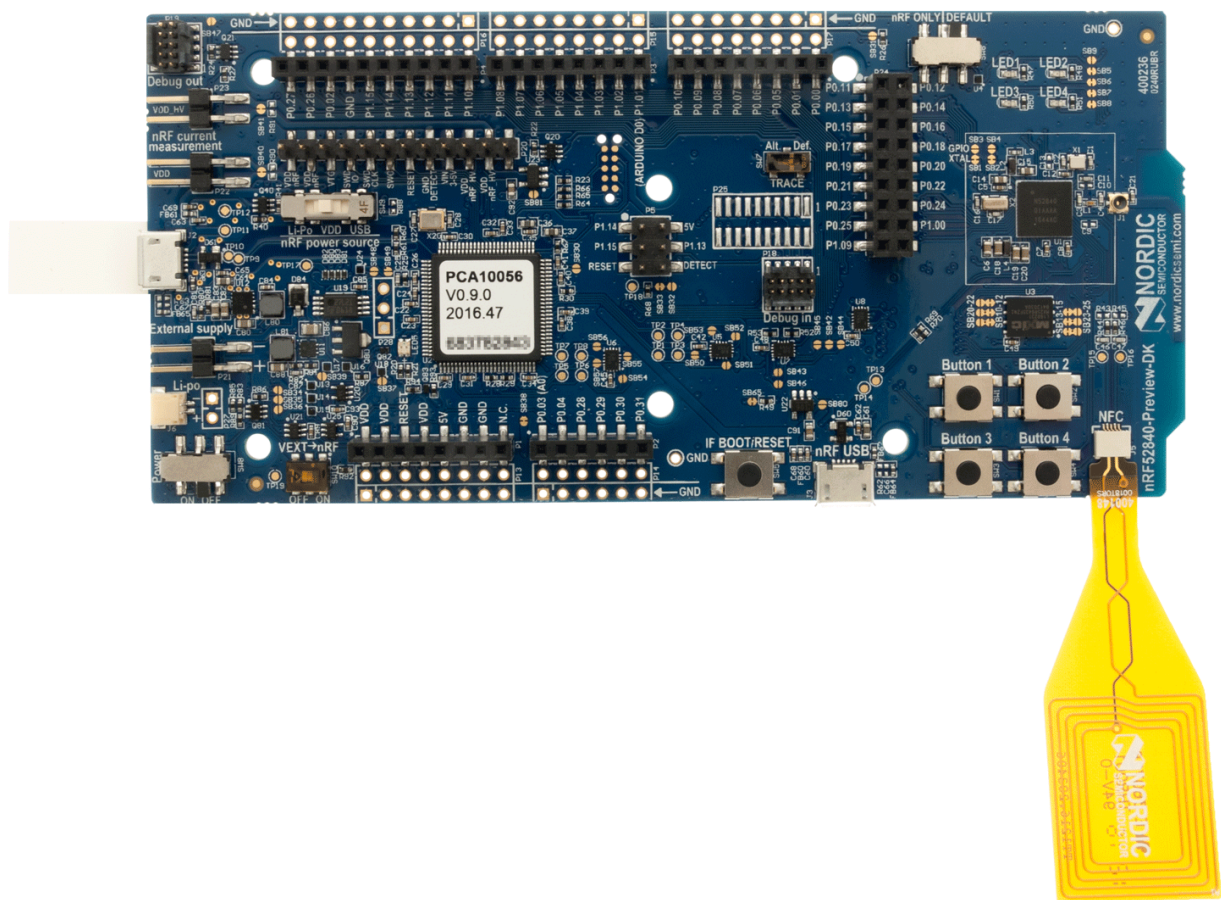
---

The nRF52840 Preview Development Kit (DK) includes hardware, firmware source code, documentation, hardware schematics, and layout files.

The key features of the development kit are:

- nRF52840 flash-based ANT/ANT+, *Bluetooth*<sup>®</sup> low energy SoC solution
- 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
- UART interface through virtual COM port
- USB
- Flash memory
- Drag-and-drop mass storage device (MSD) programming
- Supporting NFC-A listen mode
- mbed<sup>™</sup> enabled

For access to firmware source code, hardware schematics, and layout files, see [www.nordicsemi.com](http://www.nordicsemi.com).



**Figure 1: nRF52840 Preview Development Kit board (PCA10056) and NFC antenna**



**Environmental Protection**

Waste electrical products should not be disposed of with household waste.

Please recycle where facilities exist. Check with your local authority or retailer for recycling advice.

---

## Chapter 2

# Setting up the development kit

---

Setting up the development kit is as easy as connecting it to a computer with a USB cable.

1. To set up the hardware, follow the instructions in [Getting started with the nRF52840 Development Kit](#).
2. To set up the software, follow the instructions in [Nordic tools and downloads](#) in nRF5 Getting Started.  
Actual software required depends on your OS and Development IDE.

---

## Chapter 3

# Software tools

---

The extensive range of supporting software tools help you with testing and programming on your chip.

- **S140 SoftDevice:** *Bluetooth*<sup>®</sup> low energy concurrent multi-link protocol stack solution supporting simultaneous Central/Peripheral/Broadcaster/Observer role connections.
- **nRF5 SDK:** The nRF5 Software Development Kit (SDK) provides source code of examples and libraries forming the base of your application development.
- **nRF5x Command Line Tools:** nRF5x Tools is a package that contains JLinkARM, JLink CDC, nRFjprog, and mergehex. The nRFjprog is a command line tool for programming nRF5x Series chips. It is also useful in a production setup. See also [nRF5x Command Line Tools](#).
- **nRF5x-pynrfjprog:** the nRF5x-pynrfjprog utility is a simple Python interface for the nrfjprog DLL. It is useful for scripting, especially in automated tests. See also [nRF5x-pynrfjprg](#).
- **nRFgo Studio:** nRFgo Studio is a graphical user interface for programming nRF5x SoftDevices, applications, and bootloaders.
- **nRF Connect for desktop:** nRF Connect is a desktop application for getting familiar with, developing, and testing *Bluetooth*<sup>®</sup> low energy. nRF Connect allows you to set up a local device, connect it to advertising devices and discover their services, maintain the connection and the connection parameters, pair the devices and change the server setup for your local device. nRF Connect also offers a detailed log for troubleshooting purposes.
- **nRF Connect for mobile:** nRF Connect for mobile is a powerful generic tool that allows you to scan and explore your *Bluetooth*<sup>®</sup> low energy devices and communicate with them on a smartphone. nRF Connect for mobile supports a number of *Bluetooth*<sup>®</sup> SIG adopted profiles together with the Device Firmware Update (DFU) profile from Nordic Semiconductor.

We also recommend some third party software tools that are useful when developing with our products:

- **Keil MDK-ARM Development Kit:** Keil<sup>®</sup> MDK-ARM Development Kit is a development environment specifically designed for microcontroller applications that lets you develop using the nRF5 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 our development kits.



---

## Chapter 4

# Start developing

---

After you have set up the development kit and installed the toolchain, it is time to start developing.

There are several ways to continue from here. For more information, see the following nRF5 SDK instructions :

- [Running precompiled examples](#)

See how you can quickly test a precompiled example without having to use the full toolchain, it is a matter of copying and pasting a precompiled hex file onto your development kit board.

- [Compiling and running a first example](#)

Test that you have set up your toolchain correctly by compiling, programming and running a very simple example.

- [Running examples that use a SoftDevice](#)

Before you can run more advanced examples that use *Bluetooth*<sup>®</sup> or *ANT*<sup>™</sup>, you must first program the SoftDevice on the board.

# Chapter 5

## Interface MCU

The interface MCU on the nRF52840 Preview Development Kit (DK) board can run either SEGGER J-Link OB or mbed™ OB interface firmware and is used to program and debug the firmware of the nRF52840 SoC.

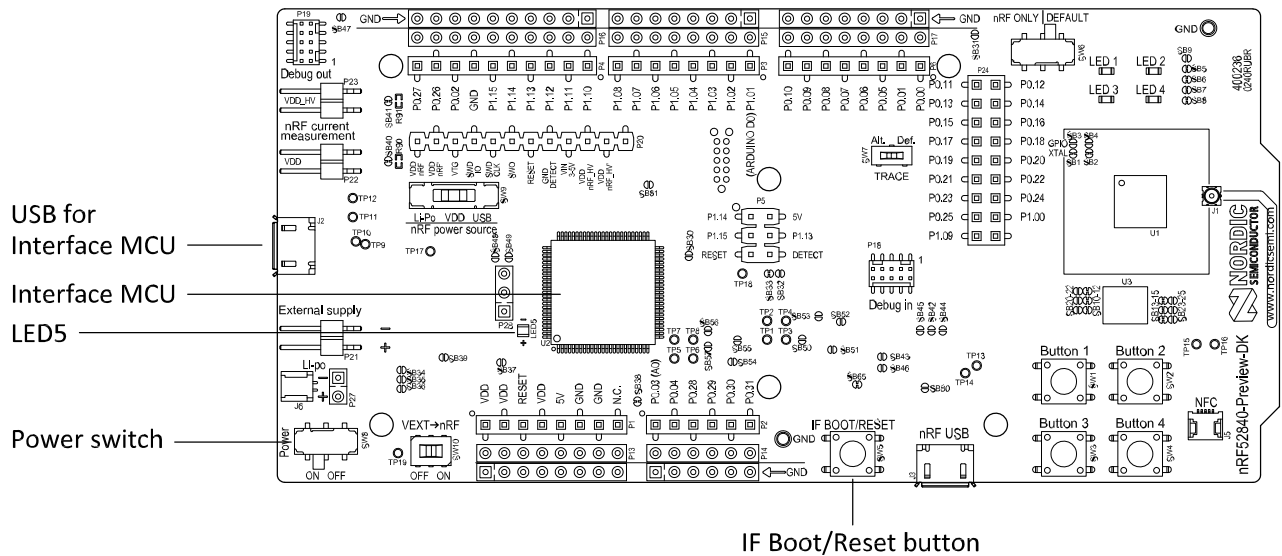


Figure 2: Interface MCU

### 5.1 IF Boot/Reset button

The nRF52840 Preview DK board is equipped with an IF Boot/Reset button (SW5).

This button is connected to the interface MCU on the board and has two functions:

- Resetting the nRF52840 SoC.
- Entering bootloader mode of the interface MCU.

During normal operation the button will function as a reset button for the nRF52840 SoC. For this to work, pin reset on **P0.18** needs to be enabled in the SoC.

The button is also used to enter the bootloader mode of the interface MCU. To enter the bootloader mode, keep the reset button pressed while powering up the board until LED (**LED5**) starts to blink. You can power up the board either by disconnecting and reconnecting the USB cable, or toggling the power switch (**SW8**).

**Important:** Pin reset can be enabled by defining the CONFIG\_GPIO\_AS\_PINRESET variable in the project settings. This can be done, for example, by defining the preprocessor symbol in Keil®, go to: **Project > Options for Target > C/C++ > Preprocessor Symbols > Define**. Here you can add the CONFIG\_GPIO\_AS\_PINRESET variable after NRF52840\_XXAA.

This functionality can be removed by using `nrfjprog.exe` with command `--recover -f NRF52`.

### 5.2 Virtual COM port

The on-board interface MCU features UART interface through virtual COM port.

The virtual COM port has the following features:

- Flexible baudrate setting up to 1 Mbps.<sup>1</sup>
- Dynamic hardware flow control (HWFC) handling.
- Tri-stated UART lines when no terminal is connected.

Table 2: Relationship of UART connections on nRF52840 and interface MCU on page 11 shows an overview of the UART connections on nRF52840 and the interface MCU.

**Table 2: Relationship of UART connections on nRF52840 and interface MCU**

GPIO nRF52840	nRF52840 UART
P0.05	RTS
P0.06	TXD
P0.07	CTS
P0.08	RXD

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

**Important:** The terminal software used must send a DTR signal in order to configure the UART interface MCU pins.

The **P0.05** (RTS) and **P0.07** (CTS) can be used freely when HWFC is disabled on the SoC.

**Important:** mbed™ does not support HWFC through the virtual COM port.

### 5.2.1 Dynamic HWFC handling

When the interface MCU receives a DTR signal from a terminal, it performs automatic hardware flow control (HWFC) detection.

Automatic HWFC detection is done by driving **P0.07** (CTS) from the interface MCU and evaluating the state of **P0.05** (RTS) when the first data is sent or received. If the state of **P0.05** (RTS) is high, HWFC is assumed not to be used. If HWFC is not detected, both CTS and RTS can be used freely by the nRF application.

After a power-on reset of the interface MCU, all UART lines are tri-stated when no terminal is connected to the virtual COM port. Due to the dynamic HWFC handling, if HWFC has been used and detected, **P0.07** (CTS) will be driven by the interface MCU until a power-on reset has been performed or until a new DTR signal is received and the detection is redone.

To ensure that the UART lines are not affected by the interface MCU, the solder bridges for these signals can be cut and later resoldered if needed. This might be necessary if UART without HWFC is needed while **P0.05** (RTS) and **P0.07** (CTS) are used for other purposes.

## 5.3 Interface MCU firmware

The on-board interface MCU is factory programmed with an mbed compliant bootloader. This feature enables the ability to swap the interface firmware between the factory preloaded SEGGER J-Link OB and the mbed™ interface firmware.

See section [IF Boot/Reset button](#) on page 10 on how to enter the bootloader.

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

Both the mbed™ interface firmware and the J-Link OB image can be downloaded from [www.nordicsemi.com](http://www.nordicsemi.com).

<sup>1</sup> Baudrate 921 600 is not supported through the virtual COM port.

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

## 5.4 MSD

The interface MCU features a mass storage device (MSD). This makes the development kit appear as an external drive on your computer.

This drive can be used for drag-and-drop programming. Files cannot be stored on this drive. By copying a HEX file to the drive, the interface MCU will program the file to the nRF52840 device.

**Important:**

- Windows might try to defragment the MSD part of the interface MCU. If this happens, the interface MCU will disconnect and be unresponsive. To return to normal operation, the development kit must be power cycled.
- If the computer is set up to boot from USB, it can try to boot from the development kit if the development kit is connected during boot. This could be avoided by unplugging the development kit before a computer restart, or changing the boot sequence of the computer.

You can also disable the MSD of the kit by using the `msddisable` command in J-Link Commander. To enable, use the `msdenable` command. These commands take effect after a power cycle of the development kit and should stay this way until changed again.

# Chapter 6

## Hardware description

The nRF52840 Preview Development Kit (DK) board PCA10056 can be used as a development platform for the nRF52840 SoC. It features an onboard programming and debugging solution.

In addition to radio communication, the nRF52840 SoC can communicate with a computer through USB and a virtual COM port provided by the interface MCU.

### 6.1 Hardware drawings

nRF52840 Preview DK hardware drawings show both sides of the PCA10056 board.

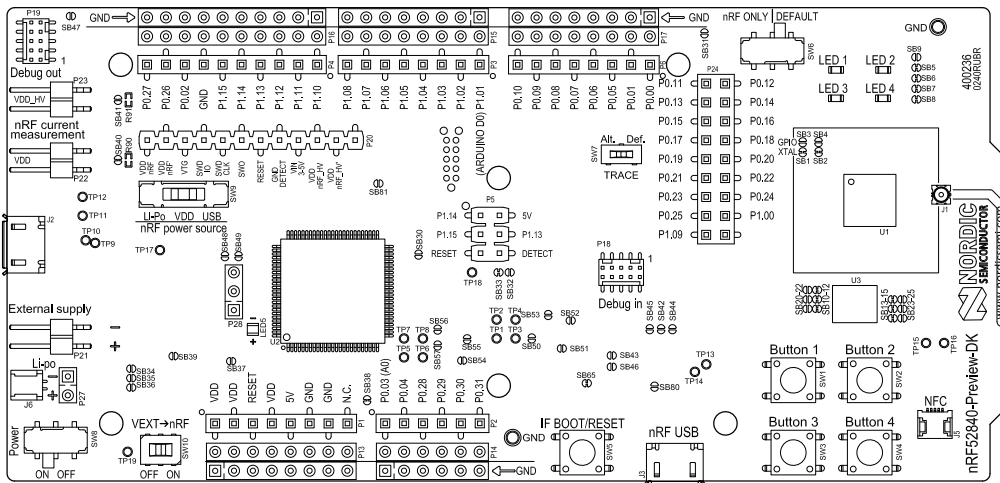


Figure 3: nRF52840 Preview DK board front view

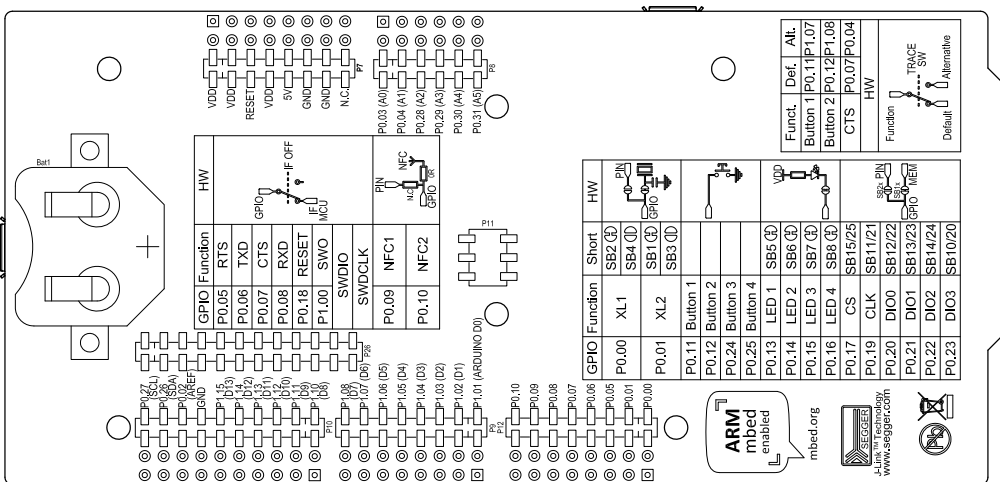
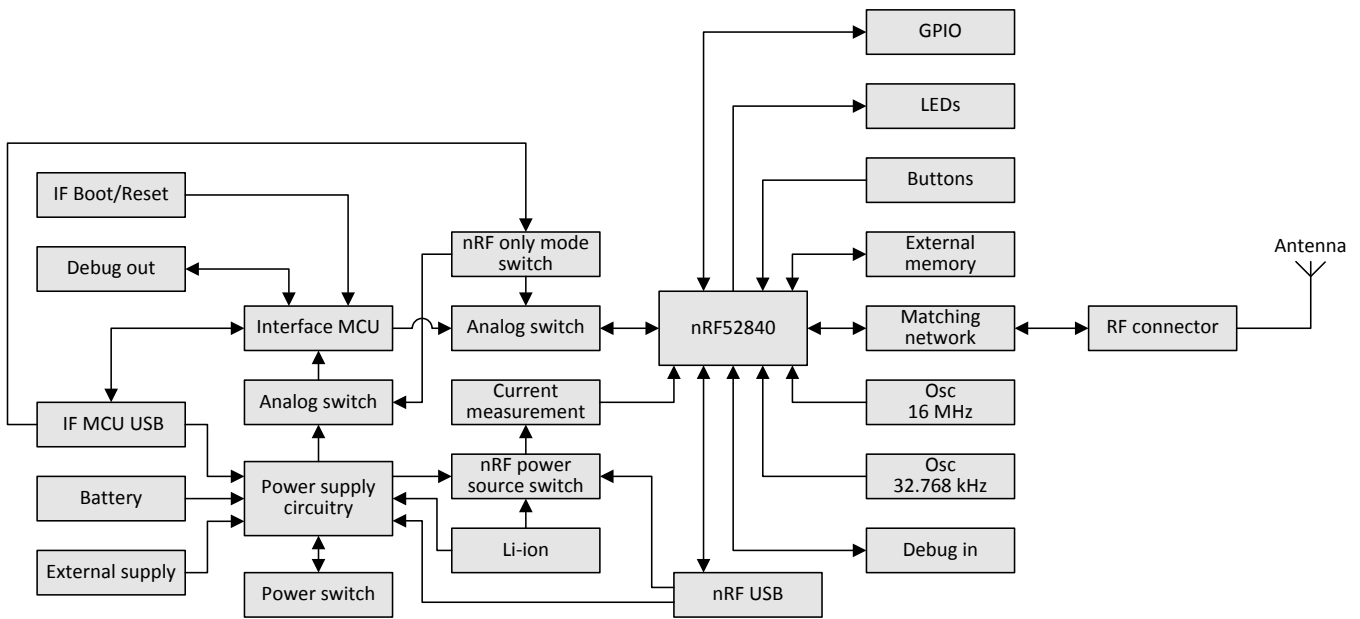


Figure 4: nRF52840 Preview DK board back view

### 6.2 Block diagram

The nRF52840 Preview DK block diagram shows the connections between the different blocks.



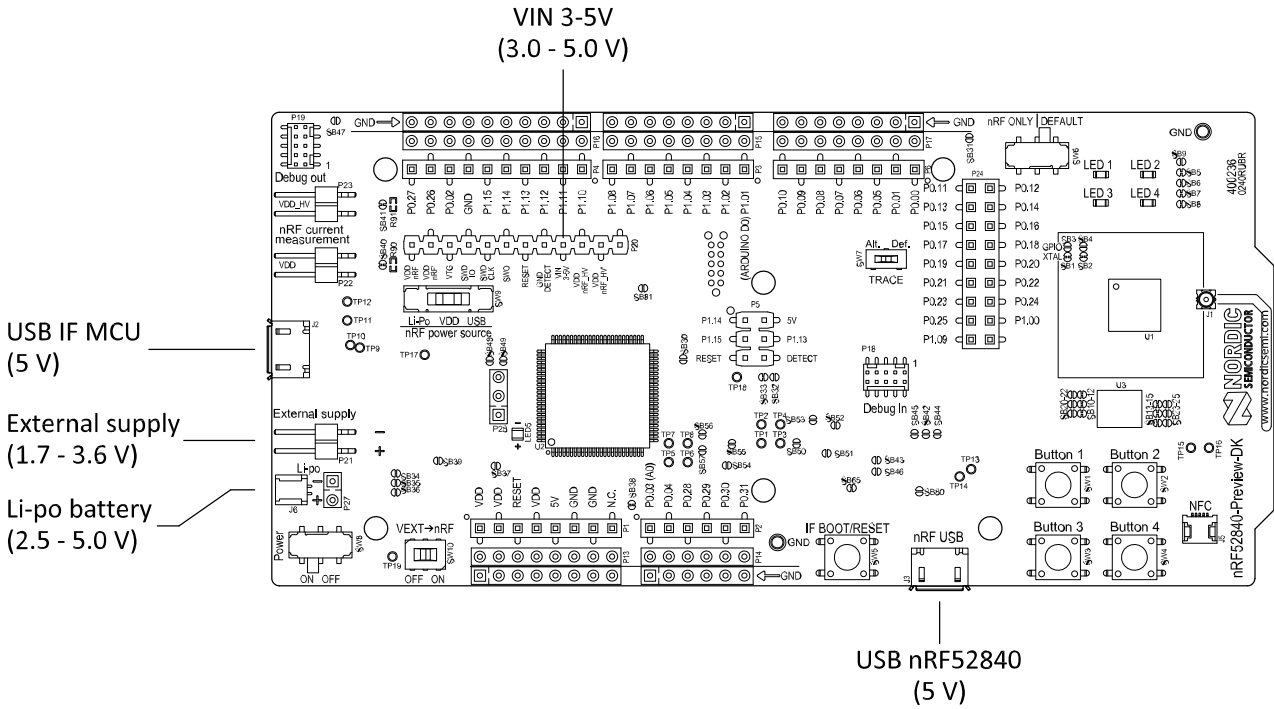
**Figure 5: nRF52840 Preview DK block diagram**

## 6.3 Power supply

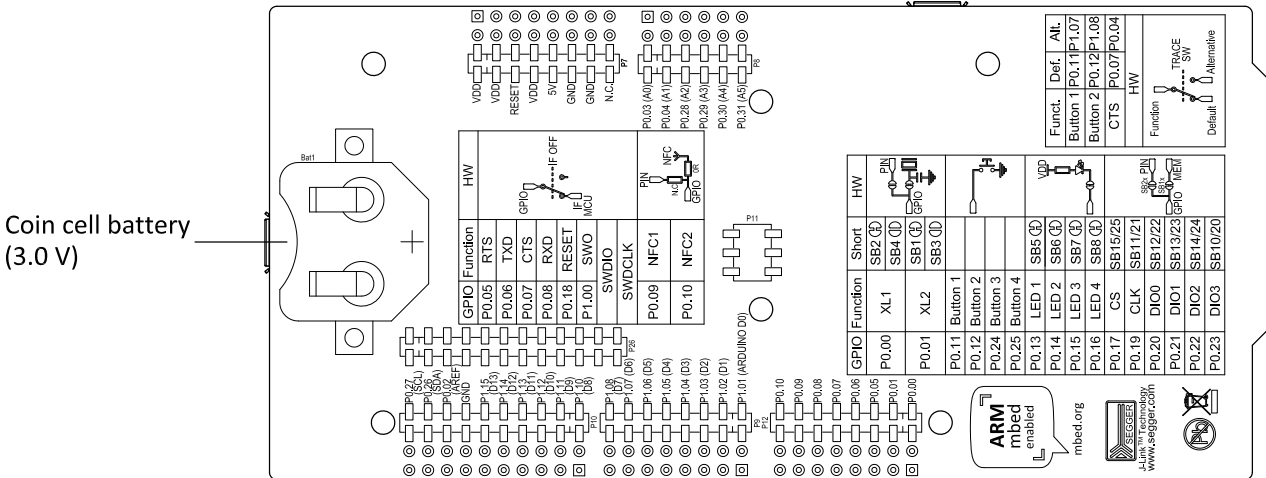
The nRF52840 Preview DK board has multiple power options.

The power options are:

- USB connector **J2** for the interface MCU (5 V)
- USB connector **J3** for the nRF52840 (5 V)
- Lithium polymer (Li-Po) battery connectors **J6** or **P27** (2.5–5.0 V)
- VIN 3–5 pin on **P20** (3.0–5.0 V)
- External supply on **P21** (1.7–3.6 V)
- Coin cell battery



**Figure 6: Power supply options (board front)**



**Figure 7: Power supply options (board back)**

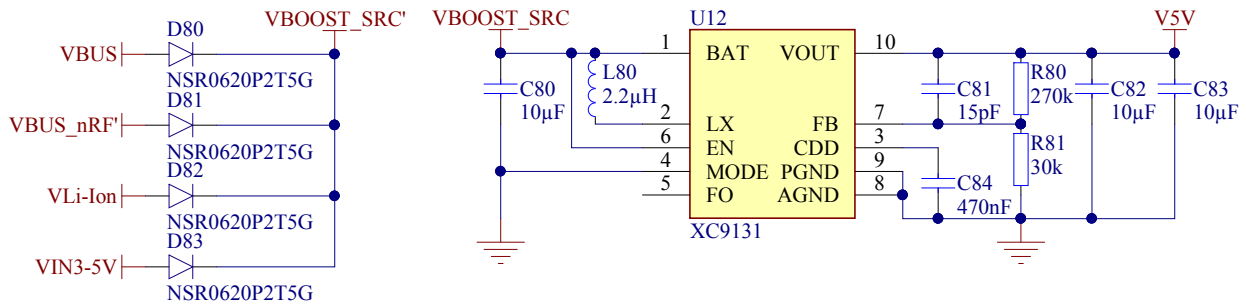
### 6.3.1 5 V power sources

The nRF52840 Preview DK board has a 5 V boost regulator.

It gives a stable 5 V output from four possible sources:

- USB connector **J2** for the interface MCU
- USB connector **J3** for the nRF52840
- Li-Po polymer battery connectors (**J6** or **P27**)
- VIN 3–5 V pin on **P20**

Each of these sources has a reverse protection diode to prevent current flowing in the wrong direction if multiple sources are connected at the same time.



**Figure 8: 5 V regulator and protecting diodes**

### 6.3.2 VDD power sources

The main board supply (VDD) can be sourced from the 5 V domain, external power supply, and coin cell battery.

For the 5 V domain, there are two regulators, one fixed 3 V buck regulator and one voltage follower regulator that follows the VDD\_nRF voltage. The coin cell battery and external power supply are not regulated.

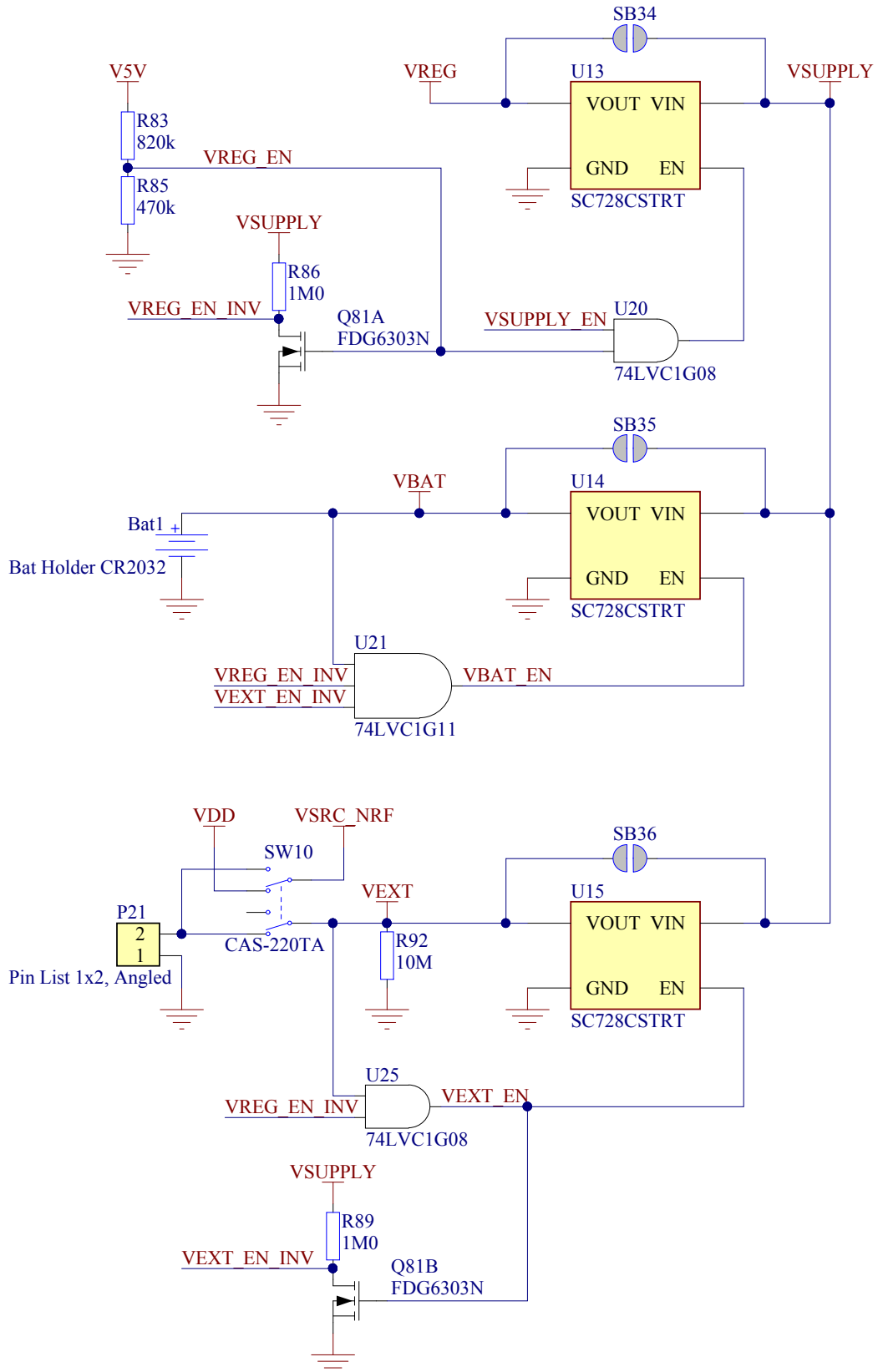
- 5 V domain:
  - Fixed 3 V buck regulator
  - VDD\_nRF voltage follower
- External power supply
- Coin cell battery

For more information about power sources, see section [nRF52840 power source](#) on page 19.

The power sources are routed through a set of load switches, which is controlled by logic to give the correct priority of the power sources.

This means that if the high voltage regulator of the nRF52840 is used, the board will be supplied from the VDD\_nRF voltage follower regardless of the state of the other power sources.





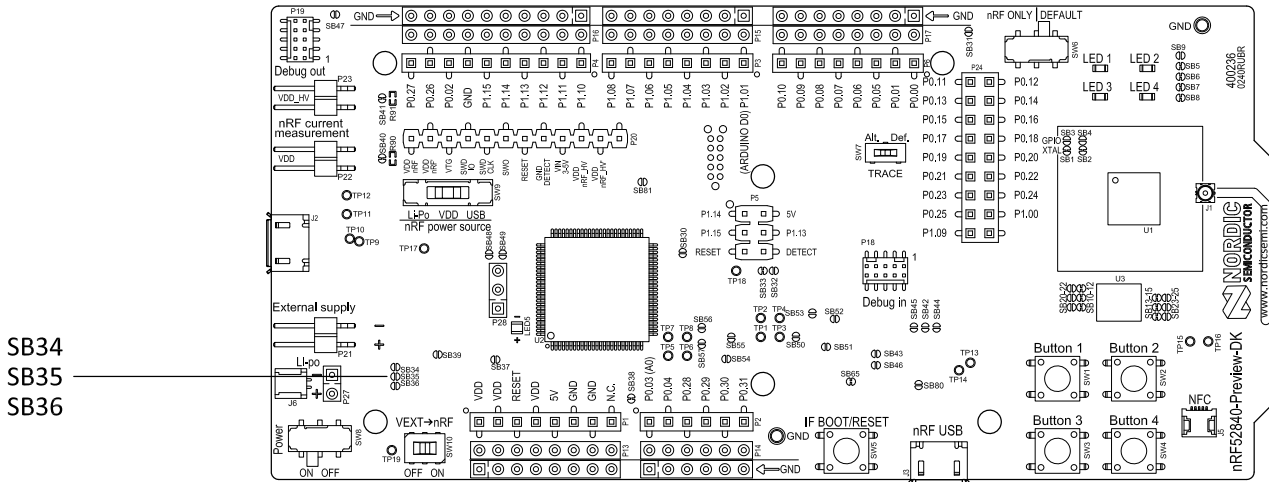
**Figure 9: Power supply circuitry**

The power switches work in the way that the body diode of the internal transistor powers the VSUPPLY net, which supplies the gates controlling the enable signal of the switches. If 5 V is present, the switches for external supply and battery are disabled. If external supply is present, the switch for the battery is disabled.

The power switches can be bypassed by shorting one or more solder bridges.

**Table 3: Power switch bypass solder bridges**

Power source	Power switch bypass	Voltage level
Regulator	SB34	3.0 V
Coin cell battery	SB35	Battery
External supply	SB36	1.7 V– 3.6 V

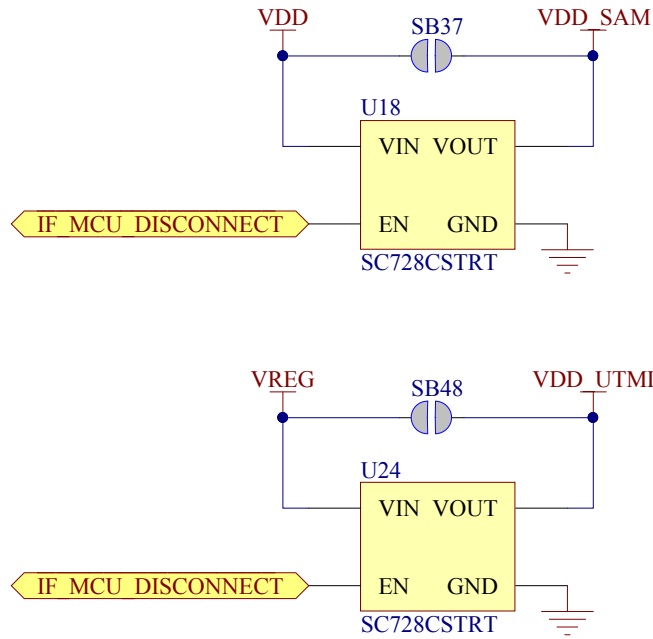


**Figure 10: Power switch bypass solder bridges**

**Important:** Connect only one power source at a time. Shorting the solder bridges removes the reverse voltage protection.

### 6.3.3 Interface MCU power

The power for the interface MCU is routed through two load switches, one for the VDD supply and one for the USB supply.



**Figure 11: Interface MCU power switch**

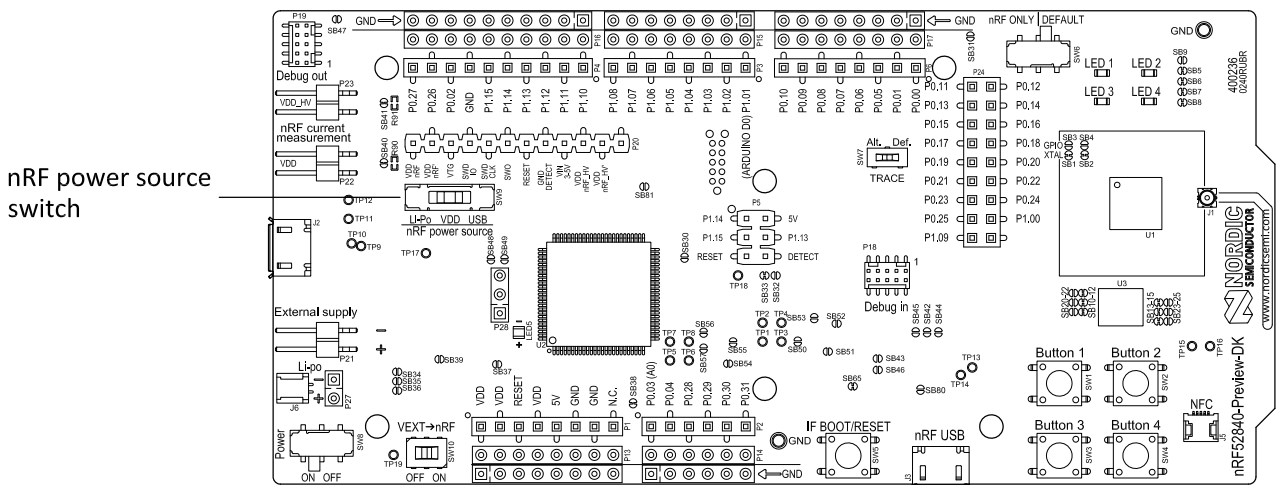
These switches are controlled by the presence of a USB connected to the interface MCU USB connector (**J2**), and the state of the nRF only switch (**SW6**). See section [Operating modes](#) on page 21 for more information. This makes it possible to disconnect the interface MCU from the power domain when not in use.

**6.3.4 nRF52840 power source**

The nRF52840 Preview DK board has a power source switch (**SW9**) for selecting between three power sources for the nRF52840 SoC.

The three positions of the switch are:

- VDD (default)
- Li-Po
- USB

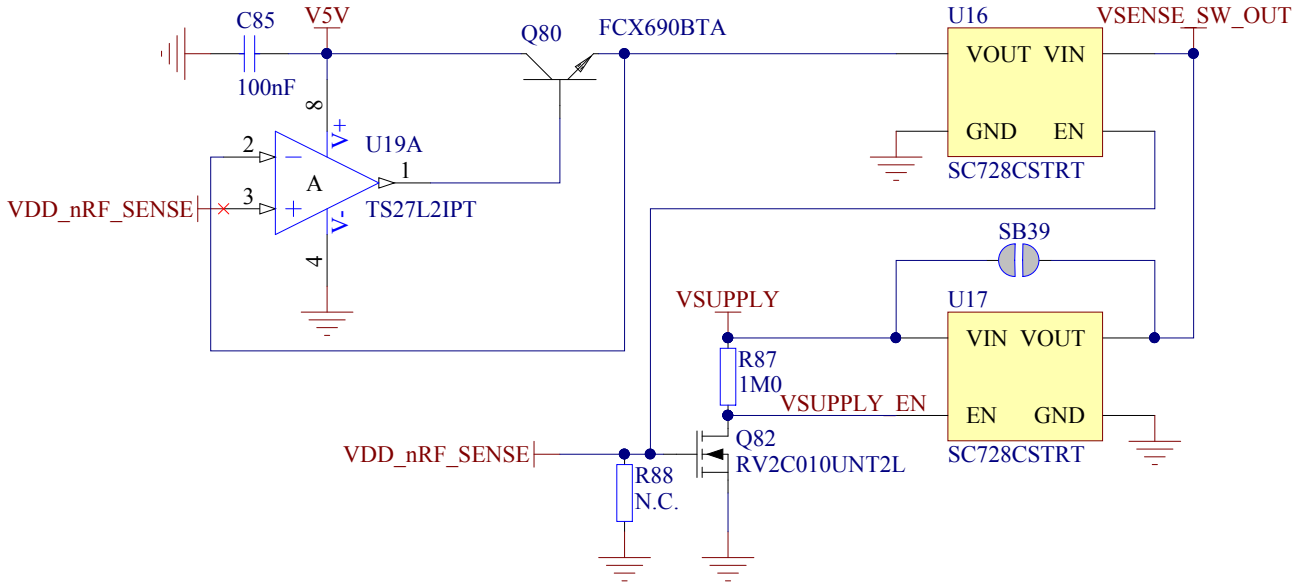


**Figure 12: nRF52840 power source switch**

The nRF52840 SoC has a high voltage buck regulator that can support up to 5 V input. In the VDD position, the SoC is powered either from the on-board buck regulator, coin cell battery, or external supply (**P21**). In the Li-

Po position, the high voltage regulator of the SoC is supplied directly from the Li-Po battery connectors (**J6** or **P27**). In the USB position, the USB high voltage regulator gets power from the nRF52840 USB connector (**J3**).

When the high voltage regulator is used, the VDD\_nRF voltage can be set by the firmware of the SoC. To make sure the rest of the board has the same voltage level, the VDD of the board is sourced by a regulator following the VDD\_nRF voltage when the high voltage regulator is used.



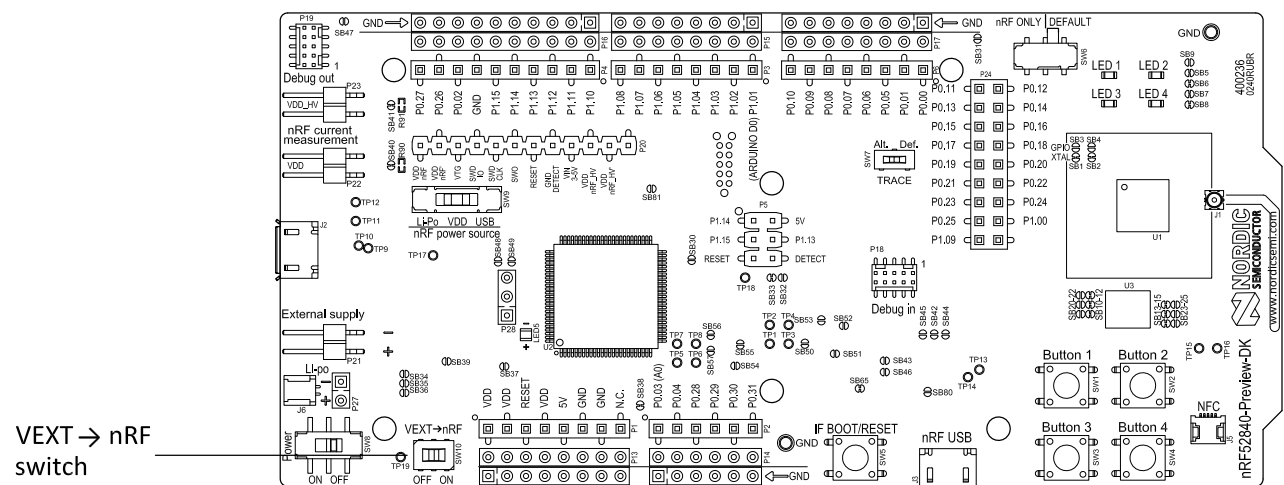
**Figure 13: VDD\_nRF voltage follower and switch**

To make sure that the nRF52840 is not powered when the nRF power switch (**SW8**) is OFF, two load switches are used, one for the high voltage regulator and one for the USB supply. These switches are controlled by VDD.

### 6.3.5 nRF52840 direct supply

It is possible to source the SoC directly from a source without powering the rest of the board from the same source.

The external source can be connected to the external supply connector (**P21**) and the VEXT->nRF switch (**SW10**) put in the ON position. The nRF power source switch (**SW9**) must be in the VDD position, and the allowed voltage range is 1.7–3.6 V.



**Figure 14: VEXT->nRF switch (SW10)**

Since it is only the nRF52840 SoC that is supplied from this source, it is recommended to supply the VDD domain from a different source (coin cell battery or USB) to prevent the pins of the SoC to be connected to unpowered devices. To prevent leakage due to voltage differences, the board should be set in the nRF only mode, see section [Operating modes](#).

**Important:** To reduce trace length and parasitic components, the external memory is connected to the SoC directly instead of using analog switches. It is recommended to cut solder bridges to avoid leakage, see section [External memory](#).

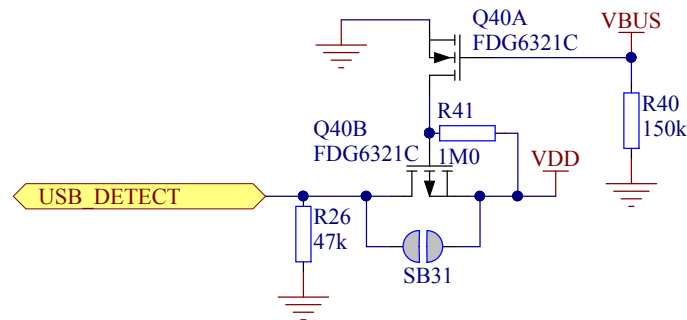
## 6.4 Operating modes

The nRF52840 Preview DK board has various modes of operation.

### 6.4.1 USB detect

To detect when USB for the interface MCU is connected, there is a circuit sensing the VBUS of USB connector J2.

When the USB cable is connected, the VDD is propagated to the USB\_DETECT signal.

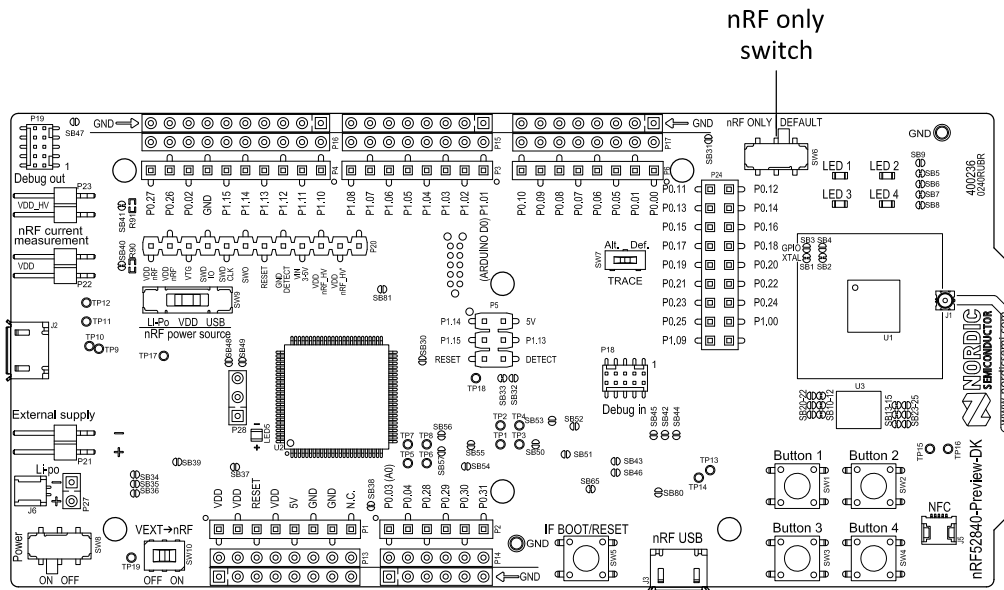


**Figure 15: USB detect**

### 6.4.2 nRF only mode

The nRF only mode disconnects the interface MCU and LEDs from the SoC using analog switches.

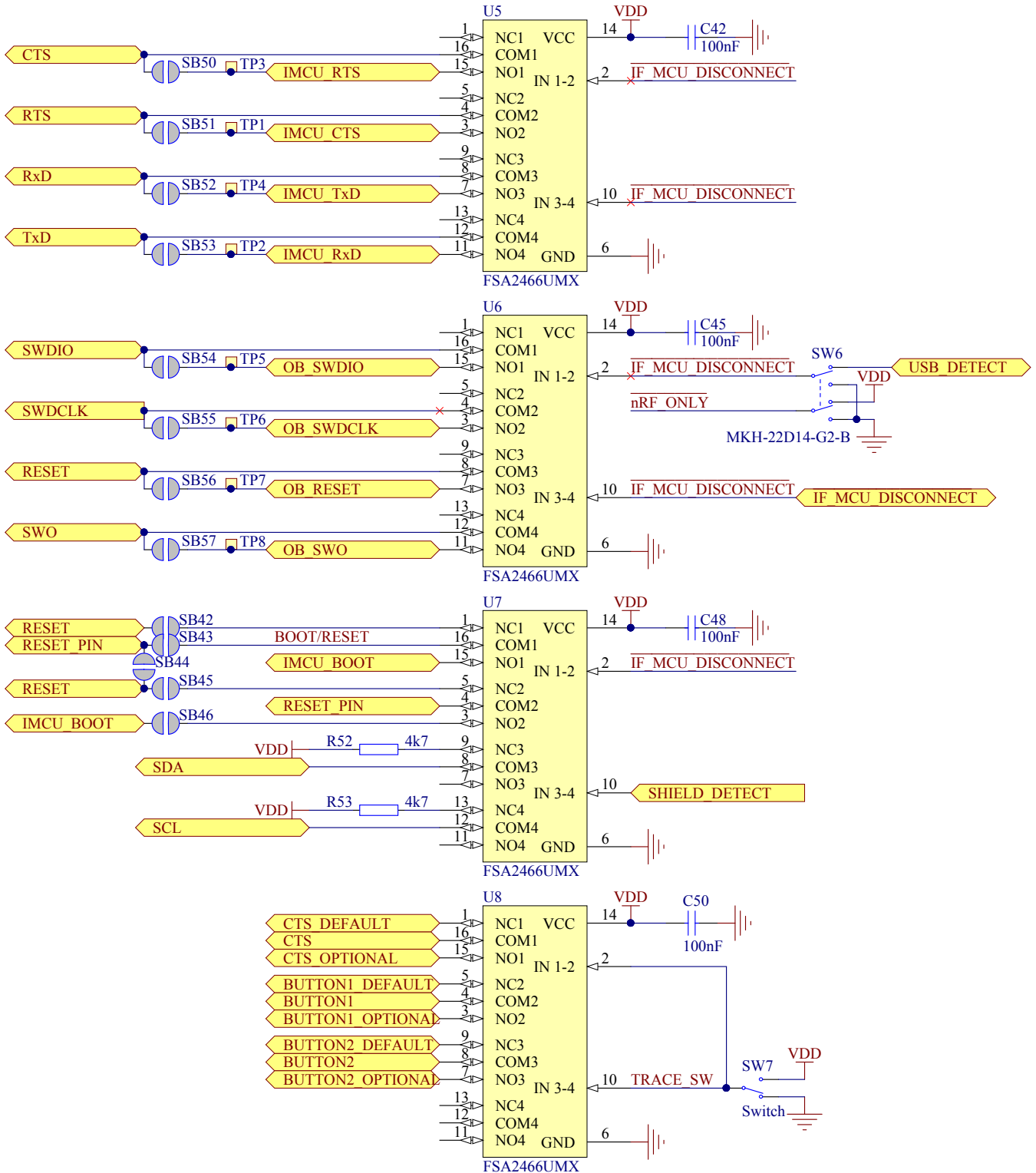
This is done to isolate the chip on the board as much as possible, and can be of use when measuring currents on low-power applications.



**Figure 16: nRF only switch (SW6)**

### 6.4.3 Signal switches

On the nRF52840 Preview DK board, there are multiple analog switches that are used to connect and disconnect signals based on different scenarios.



**Figure 17: Signal switches**

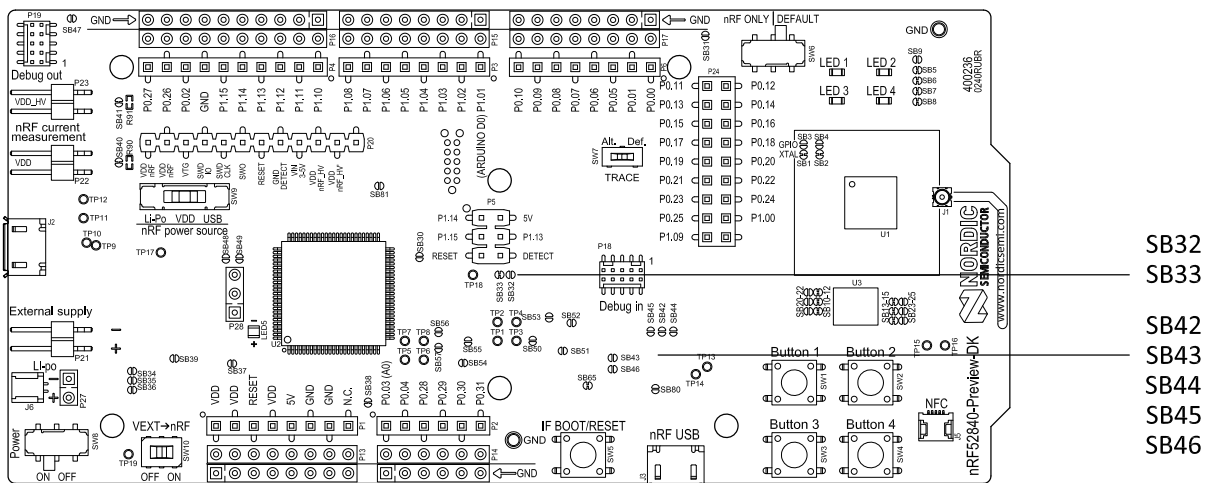
The USB and **SW6** control the signal switches by using USB\_DETECT as an input to **SW6**. Therefore, the interface MCU can be disconnected either by unplugging the USB cable in **J2**, or toggling **SW6**.

The signal controls a set of switches (**U5, U6, U7**) that break the connection between the nRF52840 and the interface MCU, and control the power for the interface MCU. For more information, see section [Interface MCU power](#).

Switches **U5** and **U6** break the connection of the UART lines and SWD/RESET lines. In addition, the signal controls the routing of the RESET signal depending on user preference when the interface MCU is connected/disconnected.

- When the interface MCU is disconnected, shorting **SB42** will connect the IF Boot/Reset button to the reset pin (**P0.18**) of nRF52840.
- When the interface MCU is disconnected, shorting **SB45** will connect the RESET pin in the Arduino interface to the reset pin (**P0.18**) of nRF52840.
- When the interface MCU is connected, shorting **SB46** will connect the RESET pin in the Arduino interface to the BOOT input of the interface MCU.
- Shorting **SB43** will connect the RESET pin in the Arduino interface to the IF Boot/Reset button.
- Shorting **SB44** will connect the RESET pin in the Arduino interface to the reset pin (**P0.18**) of nRF52840.

When a shield is connected, there are two analog switches connecting the pull-up resistors to the I2C bus lines (SDA and SCL). This function is using one ground pin on the Arduino shield to control the switch. This feature can be disabled by cutting **SB33**. To permanently enable pull-up resistors, short **SB32**.



**Figure 18: Solder bridges: Shield detect and reset behavior**

The last switch (**U8**) controls which GPIOs certain signals are routed to. This is due to some features using the same GPIOs as the Trace output by default. These analog switches are controlled by **SW7**. See chapter [Debug input and Trace](#) on page 33 for more information.

## 6.5 External memory

The nRF52840 Preview DK board has a 64-Mb external flash memory. The memory is a multi-I/O memory supporting both regular SPI and Quad SPI.

The memory is connected to the chip using the following GPIOs:

**Table 4: Flash memory GPIO usage and connecting solder bridges**

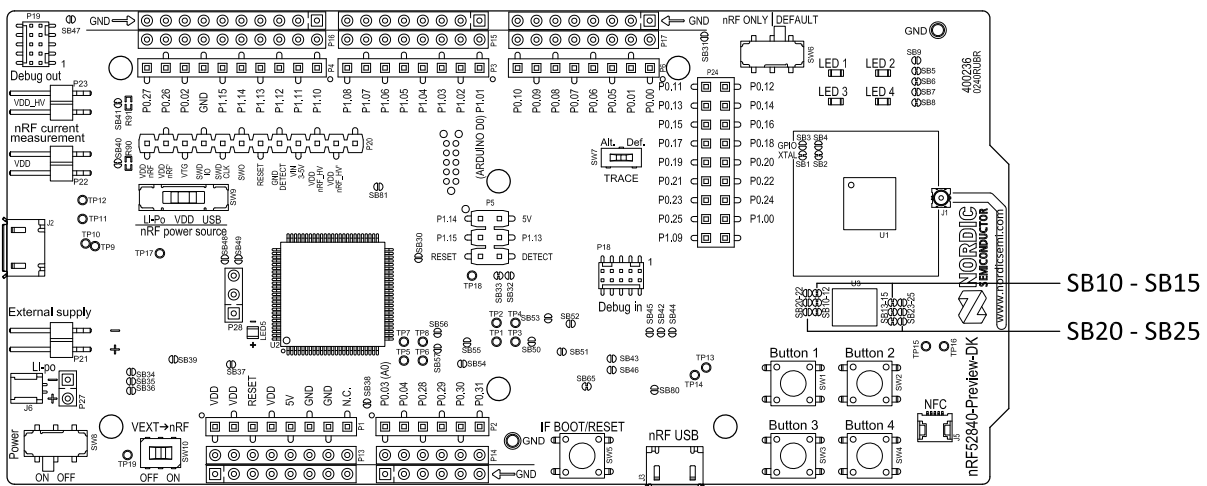
GPIO	Flash memory pin	Solder bridge for memory use (default: shorted)	Solder bridge for GPIO use (default: open)
P0.17	CS	SB15	SB25
P0.19	SCLK	SB11	SB21
P0.20	SIO_0/SI	SB12	SB22



GPIO	Flash memory pin	Solder bridge for memory use (default: shorted)	Solder bridge for GPIO use (default: open)
P0.21	SIO_1/SO	SB13	SB23
P0.22	SIO_2/WP	SB14	SB24
P0.23	SIO_3/HOLD	SB10	SB20

To use the GPIOs for a purpose other than the onboard external memory and have them available on the **P24** connector, six solder bridges (**SB10–SB15**) must be cut and six solder bridges (**SB20–SB25**) must be shorted. See [Figure 19: Configuring GPIOs for external memory](#) on page 25 for details.

**Important:** If debugging the QSPI communication is needed, the **SB20–SB25** can be shorted without cutting **SB10–SB15**, but the pins should not be driven externally.

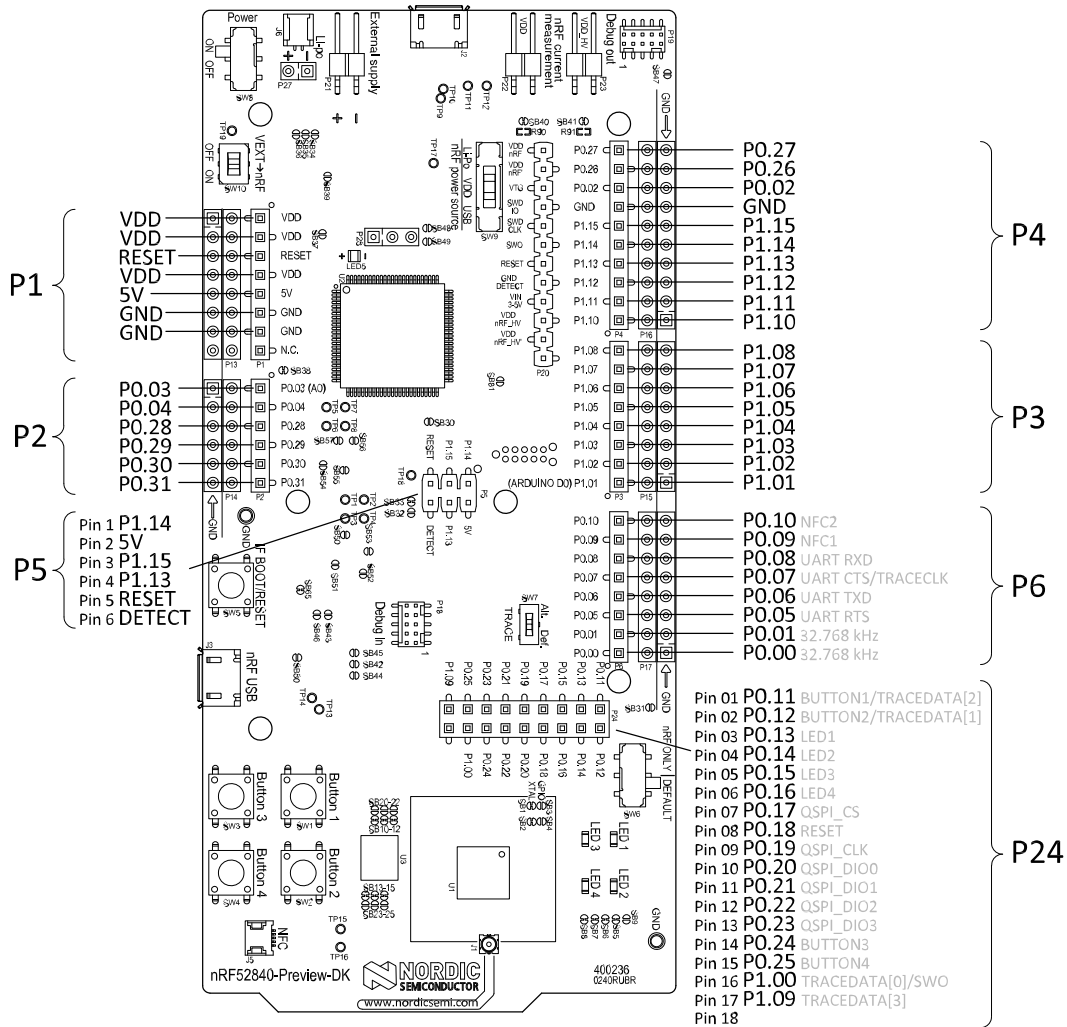


**Figure 19: Configuring GPIOs for external memory**

## 6.6 Connector interface

Access to the nRF52840 GPIOs is available from connectors **P2, P3, P4, P5, P6** and **P24**.

The **P1** connector provides access to ground and power on the nRF52840 Preview DK board.



**Figure 20: nRF52840 Preview DK board connectors**

Some of the signals are also available on connectors **P7**, **P8**, **P9**, **P10**, **P11**, and **P12**, which are on the bottom side of the board. By mounting pin lists on the connector footprints, the nRF52840 Preview DK board can be used as a shield for Arduino motherboards<sup>2</sup> or other boards that follow the Arduino standard.

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

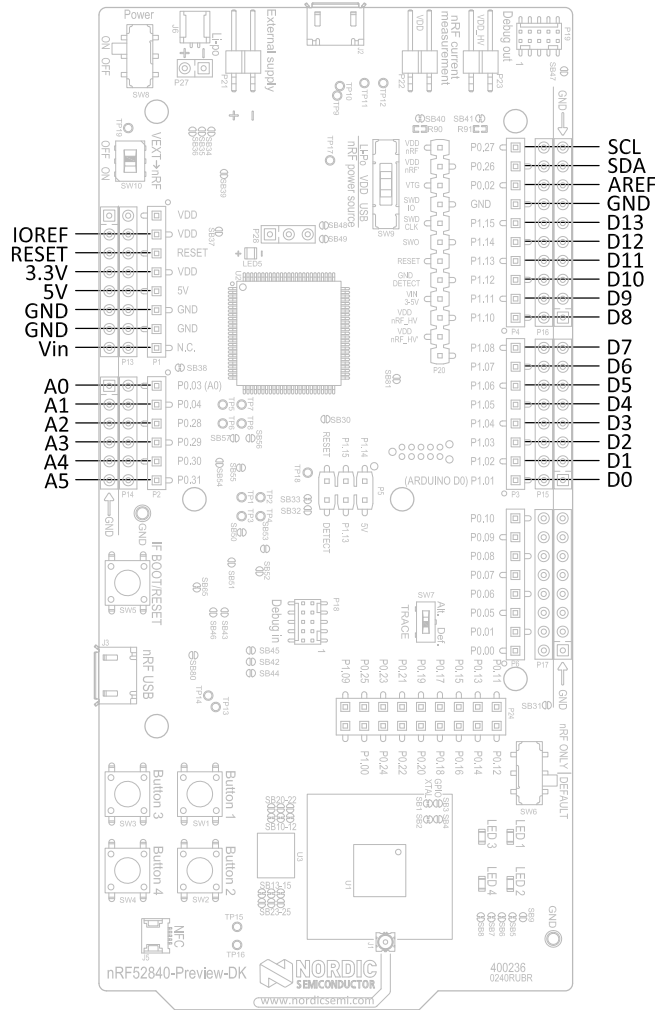
**Important:**

Some pins have default settings:

- **P0.00** and **P0.01** are used for the 32.768 kHz crystal and are not available on the connectors. For more information, see section [32.768 kHz crystal](#).
- **P0.05**, **P0.06**, **P0.07**, and **P0.08** are used by the UART connected to the interface MCU. For more information, see section [Virtual COM port](#).
- **P0.09** and **P0.10** are by default used by NFC1 and NFC2. For more information, see section [NFC antenna interface](#).
- **P0.11–P0.16** and **P0.24–P0.25** are by default connected to the buttons and LEDs. For more information, see section [Buttons and LEDs](#).
- **P0.17** and **P0.19–P0.23** are by default connected to the external memory. For more information, see section [External memory](#).

<sup>2</sup> Only 3.3 V Arduino boards.

When the nRF52840 Preview DK board is used as a shield together with an Arduino standard motherboard, the Arduino signals are routed as shown in the figure below.



**Figure 21: Arduino signals routing on the nRF52840 Preview DK board**

## 6.7 Buttons and LEDs

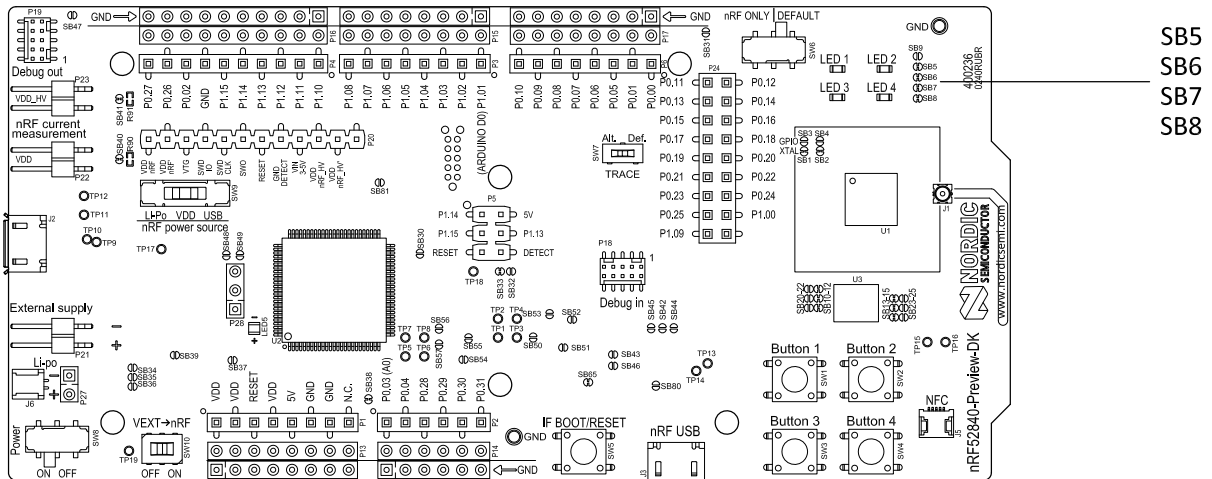
The four buttons and four LEDs on the nRF52840 Preview DK board are connected to dedicated GPIOs on the nRF52840 chip.

**Table 5: Button and LED connection**

Part	GPIO	GPIO alternative	Solder bridge
Button 1	P0.11	P1.07	-
Button 2	P0.12	P1.08	-
Button 3	P0.24		-
Button 4	P0.25		-
LED 1	P0.13		SB5
LED 2	P0.14		SB6
LED 3	P0.15		SB7

Part	GPIO	GPIO alternative	Solder bridge
LED 4	P0.16		SB8

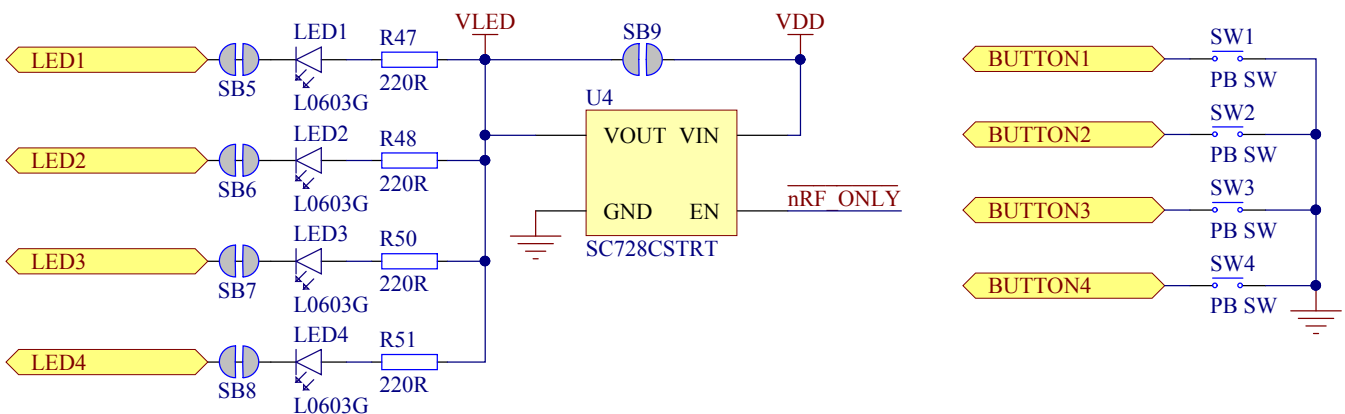
If GPIO **P0.13–P0.16** are needed elsewhere, the LEDs can be disconnected by cutting the short on **SB5–SB8**, see figure [Figure 22: Disconnecting the LEDs](#) on page 28. Since **P0.11** and **P0.12** are used as a part of the Trace functionality, Button 1 and Button 2 can be moved to alternative GPIOs, see chapter [Debug input and Trace](#) on page 33 for more information.



**Figure 22: Disconnecting the LEDs**

The buttons are active low, meaning that the input will be connected to ground when the button is activated. The buttons have no external pull-up resistor, and therefore, to use the buttons, the **P0.11, P0.12, P0.24, P0.25** pins must be configured as an input with an 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 23: Button and LED configuration**

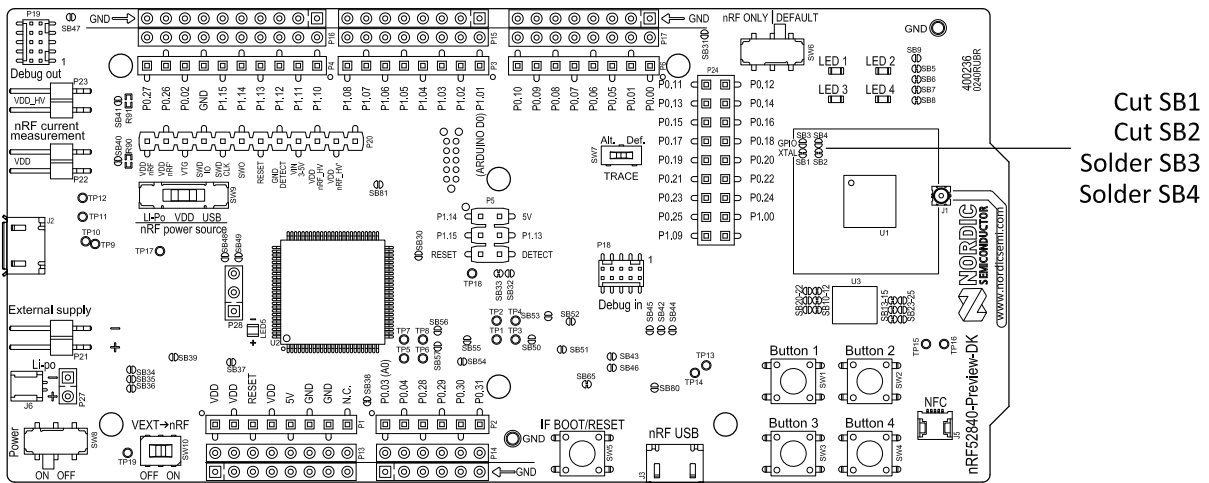
## 6.8 32.768 kHz crystal

The nRF52840 SoC can use an optional 32.768 kHz crystal (**X2**) for higher accuracy and lower average power consumption.

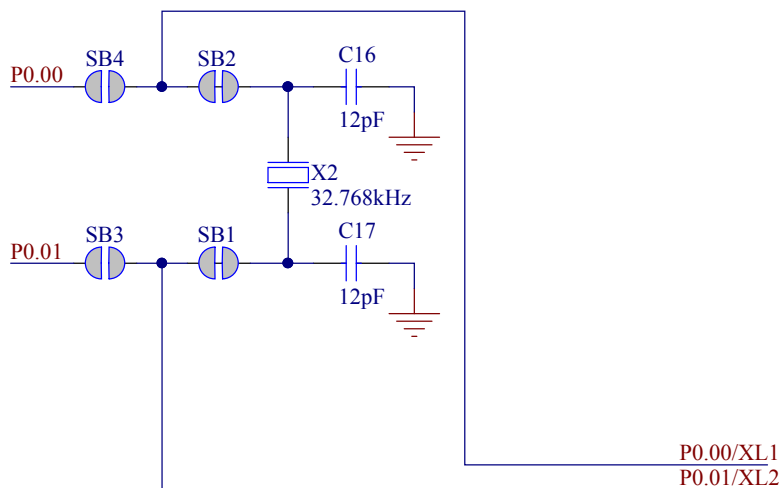
On the nRF52840 Preview DK board, **P0.00** and **P0.01** are by default used for the 32.768 kHz crystal and are not available as GPIO on the connectors.

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

If **P0.00** and **P0.01** 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 the figure below for reference.



**Figure 24: Configuring P0.00 and P0.01**



**Figure 25: 32.768 kHz crystal and SB1–SB4 schematic**

## 6.9 Measuring current

The current drawn by the nRF52840 SoC can be monitored on the nRF52840 Preview DK board.

There are several types of test equipment that can be used to measure current, and each type has some advantages and some disadvantages. The test equipment types are:

- Power analyzer
- Oscilloscope
- Ampere-meter
- Power Profiler Kit

Power analyzer and Power Profiler Kit measurements will not be described in the present document. For more information on Power Profiler Kit, see [Power Profiler Kit User Guide](#) documentation.

For instructions for measuring, see sections [Using an oscilloscope for current profile measurement](#) on page 31 and [Using an ampere-meter for current measurement](#) on page 31.

The nRF52840 SoC has two possible power supplies, VDD (1.7–3.6 V) and VDDH (2.5–5.5 V). The nRF52840 Preview DK is prepared for measuring current on both domains. Only the VDD domain current measurement is described here, but the approach is the same with the VDDH supply, see the table below for the corresponding components.

**Table 6: Components for current measurement on VDD and VDDH**

Component	VDD	VDDH
Measurement connector	P22	P23
Solder bridge	SB40	SB41
Series resistor	R90	R91

**Important:** When measuring the current consumption:

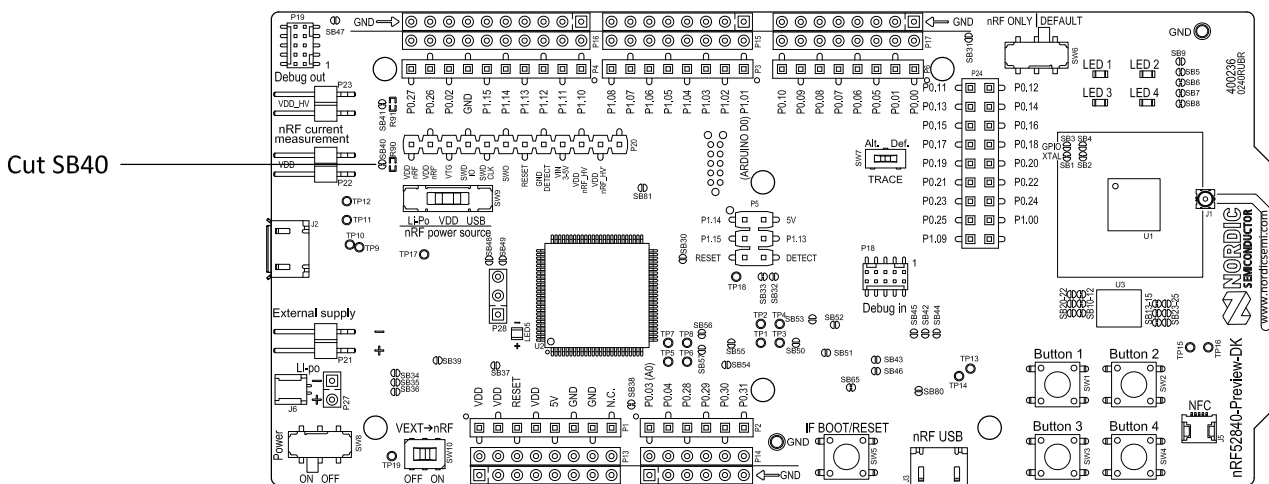
- It is not recommended to use a USB connector to power the board during current measurements. However, when measuring current on an application using the USB interface of the nRF52840 SOC, the USB must be connected. It is recommended to power the board from a coin cell battery, external power supply on connector **P21** (1.7– 3.6 V), or through the Li-Po connector **J6** or **P27** (2.5– 5.0 V).
- The current measurements will become unreliable when a serial terminal is connected to the virtual COM port.
- After programming the nRF52840 SoC, the USB for the interface MCU must be disconnected.

For more information on current measurement, see the tutorial [Current measurement guide: Introduction](#).

### 6.9.1 Preparing the development kit board

To measure current, you must first prepare the board.

The suggested configurations actually split the power domains for the nRF52840 SoC and the rest of the board.



**Figure 26: Prepare the development kit board for current measurements**

1. Cut the PCB track shorting solder bridge **SB40** to put **P22** in series with the load.
2. To restore normal kit function after measurement, solder **SB9** or apply a jumper on **P22**
3. To reprogram the nRF52840 SoC while the board is prepared for current measurements, remove measurement devices from **P22**, and then connect the USB cable.

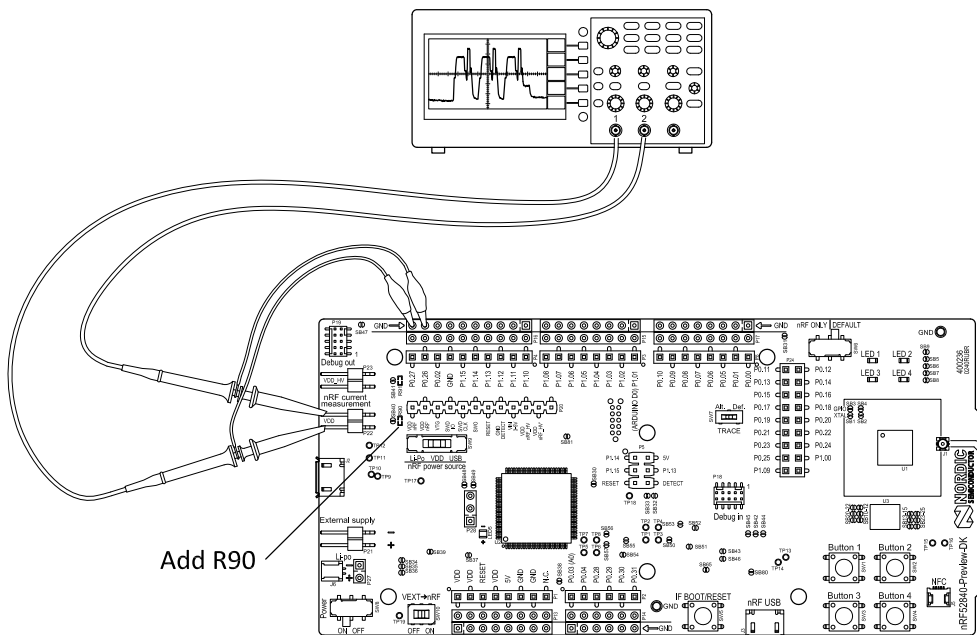
### 6.9.2 Using an oscilloscope for current profile measurement

An oscilloscope can be used to measure both the average current over a given time interval and capture the current profile.

Make sure you have prepared the development kit board as described in section [Preparing the development kit board](#) on page 30

1. Mount a 10 Ω resistor on the footprint for **R90**.
2. Connect an oscilloscope in differential mode or similar with two probes on the pins of the **P22** connector as shown in [Figure 27: Current measurement with oscilloscope](#) on page 31.
3. Calculate or plot the instantaneous current from the voltage drop across the 10 Ω resistor by taking the difference of the voltages measured on the two probes. The voltage drop will be proportional to the current. The 10 Ω resistor will cause a 10 mV drop for each 1 mA drawn by the circuit being measured.

The plotted voltage drop can be used to calculate the current at a given point in time, calculate average current over a period, or integrated to calculate the energy used over a period.



**Figure 27: Current measurement with oscilloscope**

Some tips to reduce noise:

- Use probes with 1x attenuation
- Enable averaging mode to reduce random noise
- Enable high resolution function if available

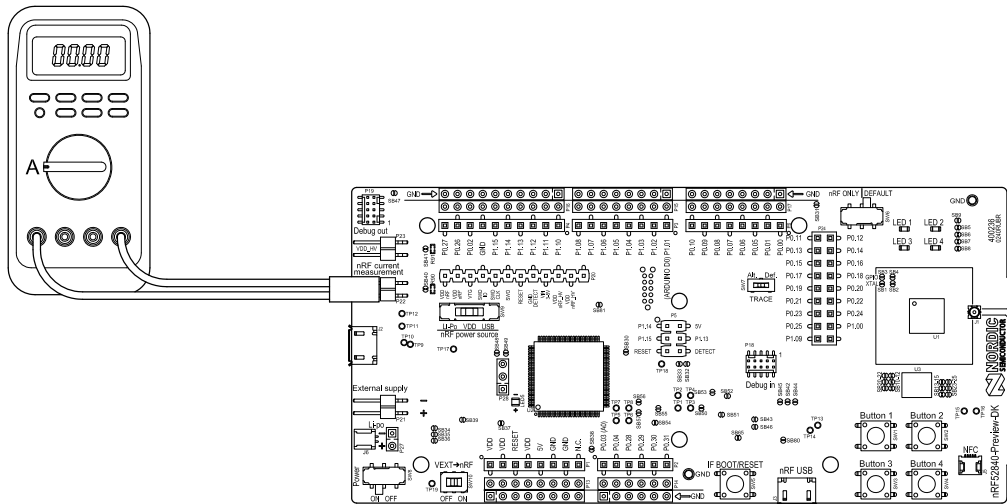
Use a minimum of 200 kSa/s (one sample every 5 μs) to get the correct average current measurement.

### 6.9.3 Using an ampere-meter for current measurement

The average current drawn by nRF52840 SoC can be measured using an ampere-meter. This method will monitor the current in series with the nRF device.

Make sure you have prepared the development kit board as described in section [Preparing the development kit board](#).

Connect an ampere-meter between the pins of connector **P22** as shown in the figure below.



**Figure 28: Current measurement with an amper-meter**

**Important:** An ampere-meter will measure the average current drawn by the nRF52840 SoC if:

- The SoC is in a state where it draws a constant current, or, the activity on the device changing load current, like BLE connection events, is repeated continuously and has a short cycle time (less than 100 ms) so that the ampere-meter will average whole load cycles and not parts of the cycle.
- The dynamic range of the ampere-meter is wide enough to give accurate measurements from 1  $\mu$ A to 15 mA.
- Recommendation: Use a true RMS ampere-meter.

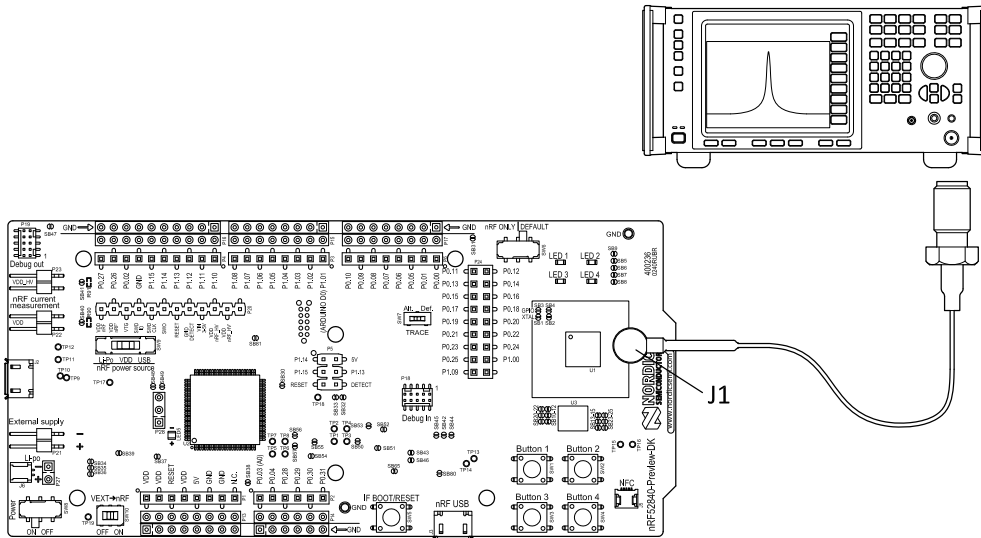
## 6.10 RF measurements

The nRF52840 Preview DK board is equipped with a small size coaxial connector (**J1**) for conducted measurements of the RF signal.

The connector is of SWF type (Murata part no. MM8130-2600) with an internal switch. By default, when no cable is attached, the RF signal is routed to the on-board PCB trace antenna.

A test probe is available (Murata part no. MXHS83QE3000) with a standard SMA connection on the other end for connecting instruments (the test probe is not included with the kit). When connecting the test probe, the internal switch in the SWF connector will disconnect the PCB antenna and connect the RF signal from the nRF52840 SoC to the test probe.





**Figure 29: Connecting a spectrum analyzer**

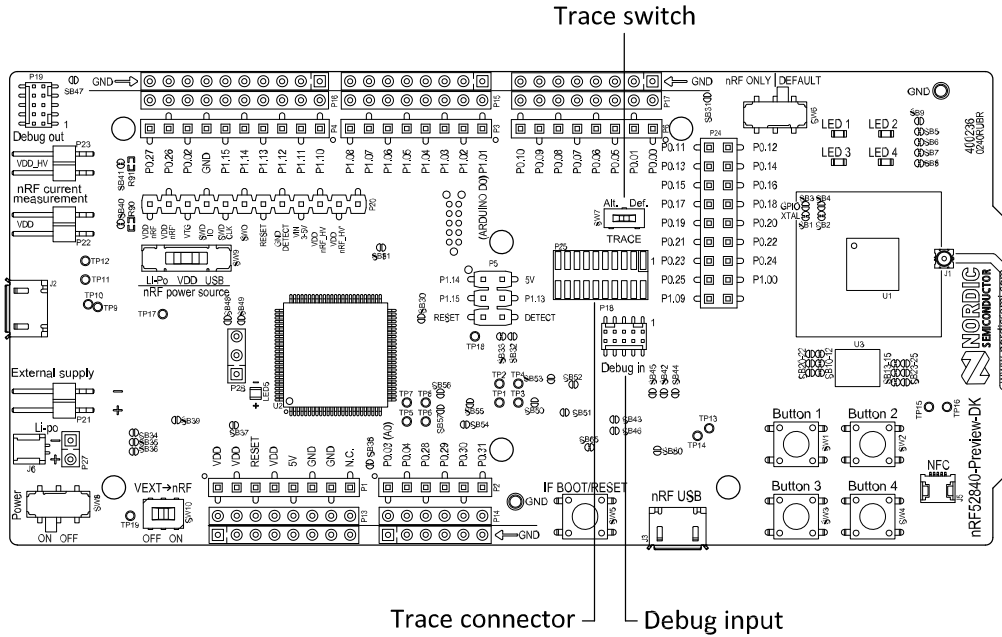
The connector and test probe will add loss to the RF signal, which should be taken into account when measuring, see the table below.

**Table 7: Typical loss in connector and test probe**

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

### 6.11 Debug input and Trace

The Debug in connector (**P18**) makes it possible to connect external debuggers for debugging when the interface MCU USB cable is not connected or the board is in nRF only mode.



**Figure 30: Debug input and Trace connectors**

For trace, a footprint for a 20-pin connector is available (**P25**). If trace functionality is required, a 2×10 pin 1.27 mm pitch surface mount pin header can be mounted.

Some of the trace pins are by default used for other functionality on the board. By sliding the Trace switch from Def. to Alt., the functionality is moved to other GPIOs. See the table below.

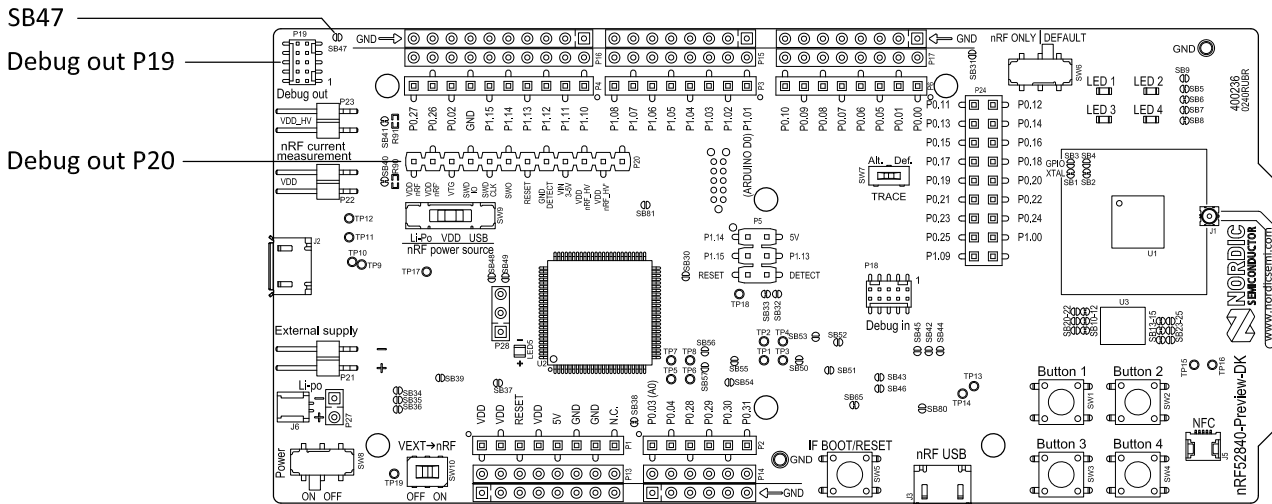
**Table 8: Default and Trace GPIOs**

GPIO	Trace	Default use	Optional GPIO
P0.07	TRACECLK	UART CTS	P0.04
P1.00	TRACEDATA[0]		
P0.11	TRACEDATA[1]	Button 1	P1.07
P0.12	TRACEDATA[2]	Button 2	P1.08
P1.09	TRACEDATA[3]		

The switching between default and optional GPIO for the signals is controlled by the Trace switch (**SW7**).

## 6.12 Debug output

The nRF52840 Preview DK board supports programming and debugging external boards with nRF51 or nRF52 SoCs. To debug an external board with SEGGER J-Link OB IF, connect to the Debug out connector (**P19**) with a 10-pin cable.



**Figure 31: 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 the external board instead of the on-board nRF52840 SoC.

**Important:** The voltage supported by external debugging/programming is the VDD voltage. Normally, this is 3 V when running from USB, but if the on-board nRF52840 is supplied from either USB or Li-ion, VDD can be set by the nRF52840 firmware. Make sure the voltage level of the external board matches the VDD of the nRF52840 Preview DK.

You can also use **P20** as a debug out connection to program shield-mounted targets. As for the Debug out header (**P19**), the interface MCU will detect the supply voltage on the mounted shield and program/debug the shield target.

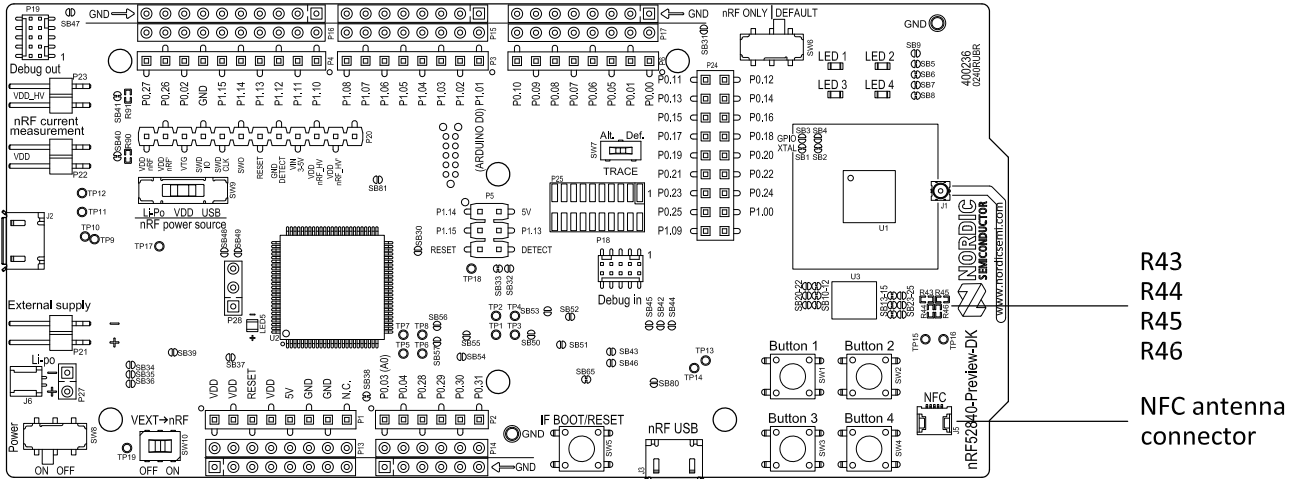
If the interface MCU detects target power on both **P19** and **P20**, it will by default program/debug the target connected to **P19**.

If it is inconvenient to have a separate power supply on the external board, the nRF52840 Preview DK board can supply power through the Debug out connector (**P19**). To enable this, short solder bridge **SB47**. Note that as long as **SB47** is shorted, it is not possible to program the onboard nRF52840 SoC even if the external board is unplugged.

### 6.13 NFC antenna interface

The nRF52840 Preview DK board supports a near field communication (NFC) tag.

NFC-A listen mode operation is supported on the nRF52840 SoC. The NFC antenna input is available on connector **J5** on the nRF52840 Preview DK board.



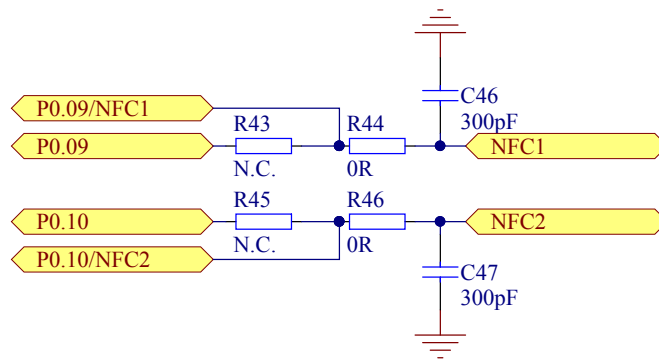
**Figure 32: NFC antenna connector**

NFC uses two pins, **L24** (NFC1) and **J24** (NFC2), to connect the antenna. These pins are shared with GPIOs (**P0.09** and **P0.10**) and the PROTECT field in the NFCPINS register in UICR defines the usage of these pins and their protection level against abnormal voltages. The content of the NFCPINS register is reloaded at every reset.

**Important:** The NFC pins are enabled by default.

NFC can be disabled and GPIOs enabled by defining the CONFIG\_NFCT\_PINS\_AS\_GPIO variable in the project settings. This can be done by defining the preprocessor symbol in Keil®. Go to: **Project > Options for Target > C/C++ > Preprocessor Symbols > Define**, and add the CONFIG\_NFCT\_PINS\_AS\_GPIO variable after NRF52. This functionality can be removed by doing a nRFJprog --recover.

Pins **L24** and **J24** are by default configured to use the NFC antenna, but if they are needed as normal GPIOs, **R44** and **R46** must be NC and **R43** and **R45** must be shorted by **0R**.



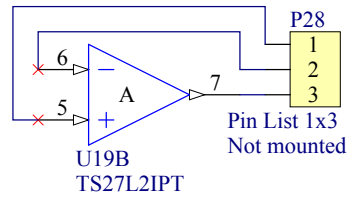
**Figure 33: NFC input schematic**

## 6.14 Extra op-amp

The voltage follower for the power supply uses a dual package op-amp.

The extra op-amp has been routed out to a connector (**P28**, not mounted) so that it is accessible for the user.

For more information on the power supply, see section [nRF52840 Power source](#).



**Figure 34: Extra op-amp**

## 6.15 Solder bridge configuration

This complete overview of solder bridges on the nRF52840 Preview DK helps you decide if you want to change the default state to achieve a different function.

**Table 9: Solder bridge configuration**

Solderbridge	Default	Function
SB1	Closed	Cut to disconnect the 32.768 kHz on P0.01
SB2	Closed	Cut to disconnect the 32.768 kHz on P0.00
SB3	Open	Short to enable P0.01 as normal GPIO
SB4	Open	Short to enable P0.00 as normal GPIO
SB5	Closed	Cut to disconnect LED1
SB6	Closed	Cut to disconnect LED2
SB7	Closed	Cut to disconnect LED3
SB8	Closed	Cut to disconnect LED4
SB9	Open	Short to bypass LED power switch
SB10	Closed	Cut to disconnect the QSPI memory from P0.23
SB11	Closed	Cut to disconnect the QSPI memory from P0.19
SB12	Closed	Cut to disconnect the QSPI memory from P0.20
SB13	Closed	Cut to disconnect the QSPI memory from P0.21
SB14	Closed	Cut to disconnect the QSPI memory from P0.22
SB15	Closed	Cut to disconnect the QSPI memory from P0.17
SB20	Open	Short to enable P0.23 as a normal GPIO
SB21	Open	Short to enable P0.19 as a normal GPIO
SB22	Open	Short to enable P0.20 as a normal GPIO
SB23	Open	Short to enable P0.21 as a normal GPIO
SB24	Open	Short to enable P0.22 as a normal GPIO
SB25	Open	Short to enable P0.17 as a normal GPIO
SB30	Open	Short to reset the interface MCU
SB31	Open	Short to bypass the USB detect switch
SB32	Open	Short to permanently enable the I2C pull-up resistors

Solderbridge	Default	Function
SB33	Closed	Cut to permanently disable the I2C pull-up resistors
SB34	Open	Short to bypass the power switch on the USB power
SB35	Open	Short to bypass the power switch on the coin cell battery power
SB36	Open	Short to bypass the power switch on the external supply power
SB37	Open	Short to bypass the interface MCU power switch
SB38	Closed	Cut to disable VDD power to the Arduino interface
SB39	Open	Short to bypass the power switch for regulator, coin cell, or external supply
SB40	Closed	Cut for current measurements of the VDD_nRF
SB41	Closed	Cut for current measurements of the VDD_nRF_HV
SB42	Open	Short to connect IF Boot/Reset button to nRF52840 reset pin when the interface MCU is disconnected
SB43	Open	Short to connect IF Boot/Reset button to RESET pin on the Arduino interface
SB44	Open	Short to connect the RESET pin on the Arduino interface to the nRF52840 reset pin
SB45	Open	Short to connect the RESET pin on the Arduino interface to the nRF52840 reset pin when the interface MCU is disconnected
SB46	Open	Short to connect the RESET pin on the Arduino interface to the interface MCU Boot when the interface MCU is disconnected
SB47	Open	Short to enable power supply of the external device when using the debug out connector
SB48	Open	Short to bypass the interface MCU USB power switch
SB49	Open	Short to connect VDD_UTMI to VDD_SAM
SB50	Open	Short to bypass the signal switch of the CTS line between nRF52840 and the interface MCU
SB51	Open	Short to bypass the signal switch of the RTS line between nRF52840 and the interface MCU
SB52	Open	Short to bypass the signal switch of the RxD line between nRF52840 and the interface MCU
SB53	Open	Short to bypass the signal switch of the TxD line between nRF52840 and the interface MCU
SB54	Open	Short to bypass the signal switch of the SWDIO line between nRF52840 and the interface MCU
SB55	Open	Short to bypass the signal switch of the SWDCLK line between nRF52840 and the interface MCU
SB56	Open	Short to bypass the signal switch of the RESET line between nRF52840 and the interface MCU
SB57	Open	Short to bypass the signal switch of the SWO line between nRF52840 and the interface MCU

---

Solderbridge	Default	Function
SB65	Closed	Cut to disable the pull-up resistor of the IF Boot/Reset button
SB80	Open	Short to bypass the power switch for the VBUS of nRF52840
SB81	Open	Short to bypass the power switch for VDD_HV of nRF52840

## **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.



