EN: This Datasheet/Document is presented by the manufacturer.

Please visit our website for pricing and availability at www.hestore.hu.

nRF52840 DK Hardware

v1.0.0

User Guide

4440_050 / 2020-12-03

NORDIC® SEMICONDUCTOR

Contents

Revision history . . . iv

Environmental and safety notices . . . v

1 Introduction . . . 6

2 Kit content . . . 7

3 Interface MCU . . . 8

3.1 IF Boot/Reset button . . . 8
3.2 Virtual serial port . . . 9
3.2.1 Dynamic hardware flow control . . . 9
3.3 Mass Storage Device . . . 10

4 Hardware description . . . 11

4.1 Hardware drawings . . . 11
4.2 Block diagram . . . 11
4.3 Power supply . . . 12
4.3.1 5 V power sources . . . 13
4.3.2 VDD power sources . . . 14
4.3.3 Interface MCU power . . . 16
4.3.4 nRF52840 power source . . . 17
4.3.5 nRF52840 SoC direct supply . . . 17
4.4 Modes of operation . . . 18
4.4.1 USB detect . . . 18
4.4.2 nRF only mode . . . 19
4.4.3 Signal switches . . . 19
4.5 External memory . . . 21
4.6 Connector interface . . . 22
4.6.1 Analog pin map . . . 24
4.7 Buttons and LEDs . . . 24
4.8 32.768 kHz crystal . . . 26
4.9 Debug input and trace . . . 26
4.10 Debug output . . . 27
4.10.1 Connectors for programming external boards . . . 28
4.11 NFC antenna interface . . . 29
4.12 Extra op-amp . . . 30
4.13 Solder bridge configuration . . . 31

5 Current measurement . . . 34

5.1 Set up the DK . . . 34
5.2 Measure current profile with an oscilloscope . . . 35
5.3 Measure average current with an ampere meter . . . 36

6 RF measurements . . . 37

Glossary . . . 38

Recommended reading . . . 41

Legal notices 42

Revision history

Date	Description
2020-12-03	Editorial changes
April 2020	Editorial changes
December 2019	Fixed broken links
February 2019	Updated Introduction on page 6: nRF52811 support
September 2018	Updated: • Nordic tools and downloads • Start developing • Set up the DK on page 34
June 2018	Updated Getting started
April 2018	First release

Previous versions

PDF files for relevant previous versions are available here:

- nRF52840 PDK User Guide v1.2
- nRF52840 PDK User Guide v1.0

Environmental and safety notices

Environmental and safety notices for the DK and power supply requirements.

Note: The nRF52840 DK must be powered by a PS1 class (IEC 62368-1) power supply with maximum power less than 15 W.

Skilled persons

The nRF52840 DK is intended for use only by skilled persons.

A skilled person is someone with relevant education or experience that enables them to identify potential hazards and takes appropriate action to reduce the risk of injury to themselves and others.



Electrostatic discharge

The nRF52840 DK is susceptible to *Electrostatic Discharge (ESD)*.

To avoid damage to your device, it should be used in an electrostatic free environment, such as a laboratory.





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.

1 Introduction

The nRF52840 DK is a hardware development platform for designing and developing application firmware on the nRF52840 *System on Chip (SoC)*.

Key features

- nRF52840 flash-based *Bluetooth*® Low Energy and ANT™/ANT+™ SoC solution
- Support for nRF52840 and nRF52811 SoCs development
- 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 a virtual serial port
- Flash memory
- Drag-and-drop *Mass Storage Device (MSD)* programming
- Support for *NFC-A Listen Mode*
- Pins for measuring power consumption

This *Development Kit (DK)* can be used to develop and test applications for the nRF52840 or nRF52811 SoCs.

For access to firmware source code, hardware schematics, and layout files, see www.nordicsemi.com.

2 Kit content

The nRF52840 DK includes hardware, preprogrammed firmware, documentation, hardware schematics, and layout files.

The nRF52840 DK (PCA10056) comes with an *Near Field Communication (NFC)* antenna.

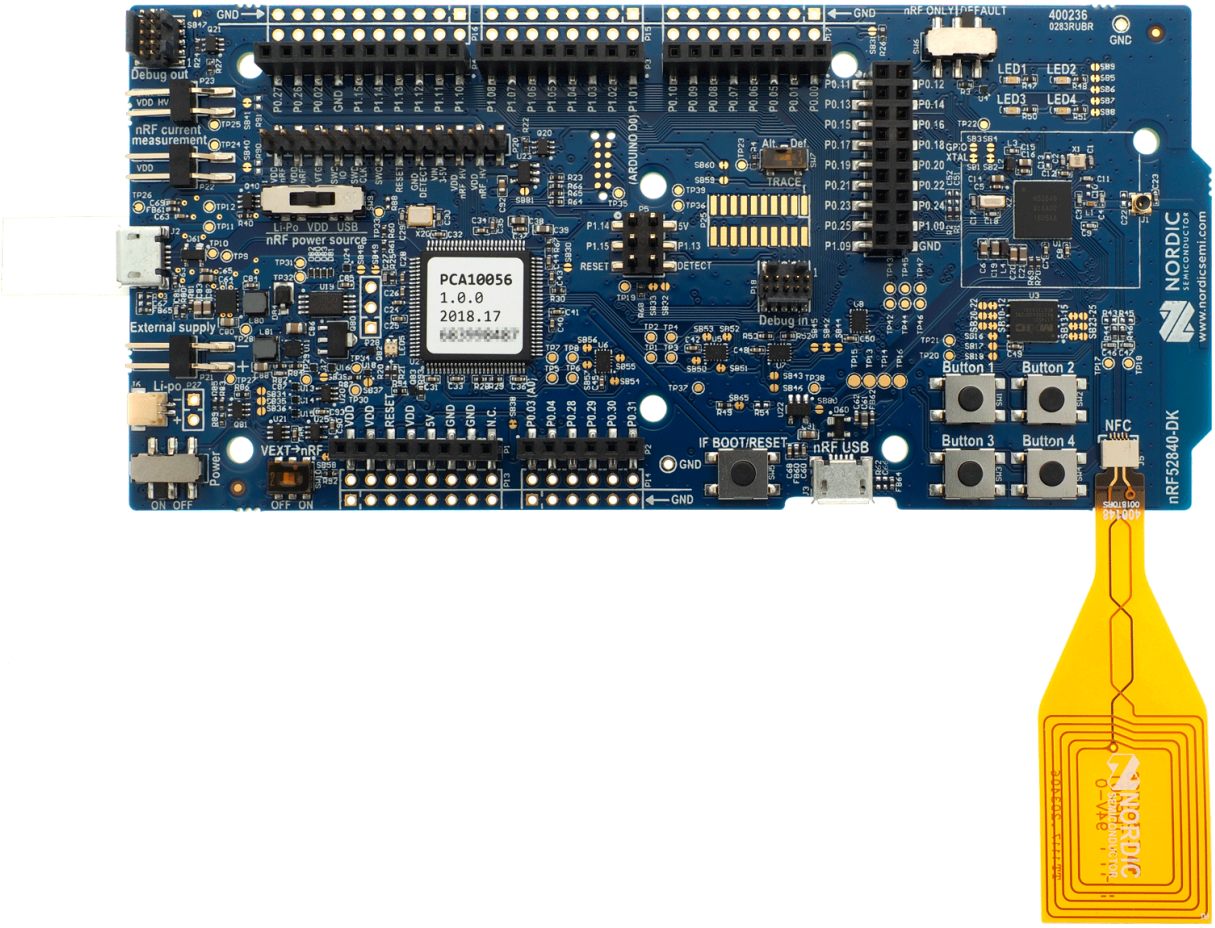


Figure 1: nRF52840 DK (PCA10056) and NFC antenna

Hardware files

The hardware design files for the nRF52840 DK are available on the nRF52840 product page. They include.

- Schematics
- *Printed Circuit Board (PCB)* layout files
- Bill of materials
- Gerber files

3 Interface MCU

The interface MCU on the nRF52840 DK runs SEGGER J-Link *Onboard (OB)* interface firmware. It is used to program and debug the application firmware of the nRF52840 *SoC*.

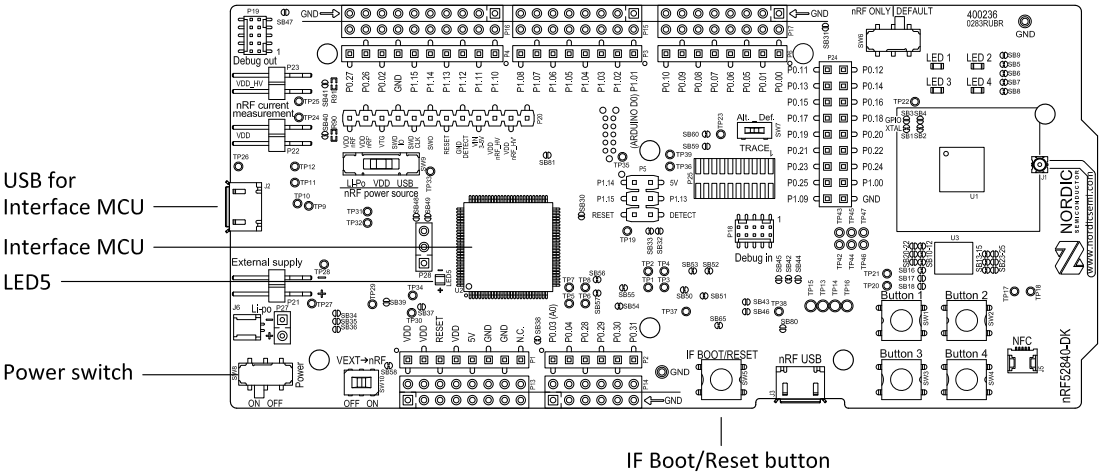


Figure 2: Interface MCU

3.1 IF Boot/Reset button

IF Boot/Reset button (**SW5**) is connected to the interface MCU on the *DK* and resets the nRF52840 *SoC* or any device connected to the external programming connectors.

The button is has two functions:

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

During normal operation the button functions as a reset button for the nRF52840 *SoC*. For this to work, pin reset on **`P0.18`** must be enabled in embedded firmware.

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 DK until **LED5** starts to blink. You can power up the DK by disconnecting and reconnecting the USB cable or by toggling the power switch (**SW8**).

Note: Pin reset can be enabled by adding the CONFIG_GPIO_AS_PINRESET variable to the compiler preprocessor macros. This depends on the *Integrated Development Environment (IDE)*/toolchain in use:

- When using SEGGER Embedded Studio, go to **Project** > **Edit Options** > **Code** > **Preprocessor** > **Preprocessor Definitions** and add the CONFIG_GPIO_AS_PINRESET variable.
- When using Keil, go to **Project** > **Options for Target** > **C/C++** > **Preprocessor Symbols** > **Define** and add the CONFIG_GPIO_AS_PINRESET variable.

If your program does not enable pin reset, this functionality can also be enabled on a previously programmed device by calling `nrfjprog.exe` with argument `--pinresetenable`. To disable pin reset again, reprogram with `--chiperase`.

3.2 Virtual serial port

The interface MCU features a *Universal Asynchronous Receiver/Transmitter (UART)* interface through a virtual serial port. One port is connected to the physical port on the *DK*, and the other is not usable.

The serial port has the following features:

- Flexible baud rate setting up to 1 Mbps (baud rate 921 600 bps is not supported)
- Dynamic *Hardware Flow Control (HWFC)*
- Tri-stated UART lines when no terminal is connected

The following table lists the nRF52840 *SoC* UART GPIO pins and their signals and the interface MCU.

GPIO nRF52840	nRF52840 UART
P0.05	RTS
P0.06	CXD
P0.07	RTS
P0.08	TXD

Table 1: Relationship of UART connections on nRF52840 and interface MCU

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 serial port on the computer. The terminal software must send a *Data Terminal Ready (DTR)* signal to configure the UART interface MCU pins.

P0.05 (*Request to Send (RTS)*) and **P0.07** (*Clear to Send (CTS)*) can be used for other purposes when HWFC is disabled on the SoC.

3.2.1 Dynamic hardware flow control

When the interface MCU receives a *DTR* signal from a terminal, it performs automatic *HWFC* detection.

HWFC detection

Automatic HWFC detection is done by driving *CTS* from the interface MCU and evaluating the state of *RTS* when the first data is sent or received. If the state of RTS is high, it is assumed HWFC is not in use. If HWFC is not detected, pins **P0.10/P0.21** **P0.07** (CTS) and **P0.05** (RTS) are free for the nRF application to use.

After a power-on reset of the interface MCU, all UART lines are tri-stated when no terminal is connected to the virtual serial port. If HWFC has been used and detected, **P0.07** (CTS) is 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.

Note:

Using HWFC pins for other tasks

P0.05 (RTS) and **P0.07** (CTS) can be disconnected from the interface MCU by switching **SW7** to OFF. This might be necessary if *UART* without HWFC is needed while **P0.05** (RTS) and **P0.07** (CTS) are used for other purposes. To ensure that the **P0.06** (TXD) and **P0.08** (RXD) lines are not affected by the interface MCU, the solder bridges for these signals can be cut and later resoldered if needed.

3.3 Mass Storage Device

The interface MCU features an *MSD*. This makes the *DK* appear as an external drive on your computer.

This drive can be used for drag-and-drop programming. However, files cannot be stored on the drive. When a HEX file is copied to the drive, the interface MCU programs the file to the DK.

The following issues might occur during MSD operation:

- If Windows tries to defragment the MSD, the interface MCU disconnects and becomes unresponsive. To return to normal operation, power cycle the DK.
- Your antivirus software might try to scan the MSD. Some antivirus programs trigger a false positive alert in one of the files and quarantine the unit. If this happens, the interface MCU becomes unresponsive.
- If the computer is set up to boot from USB, it can try to boot from the DK if it is connected. This can be avoided by unplugging the DK before a computer restart or changing the boot sequence of the computer.

You can disable the MSD of the DK 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 DK and stay this way until changed again.

4 Hardware description

The nRF52840 DK (PCA10056) features an onboard programming and debugging solution.

In addition to radio communication, the *SoC* can communicate with a computer through *Universal Serial Bus (USB)* and a virtual serial port provided by the interface MCU.

4.1 Hardware drawings

nRF52840 DK hardware drawings show both sides of the PCA10056.

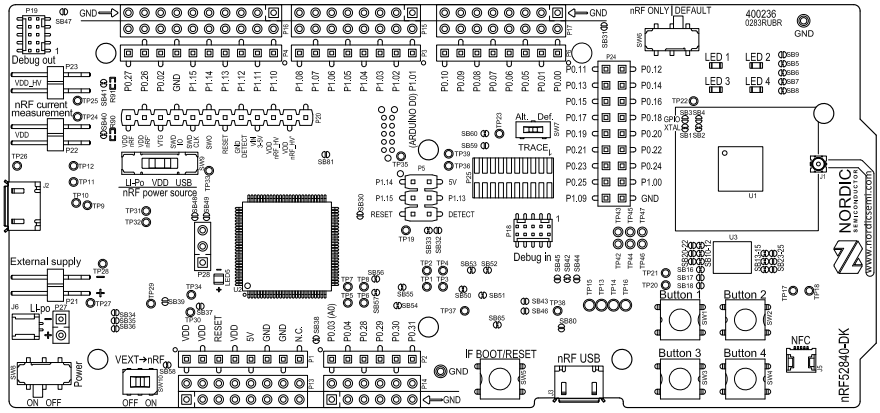


Figure 3: nRF52840 DK front view

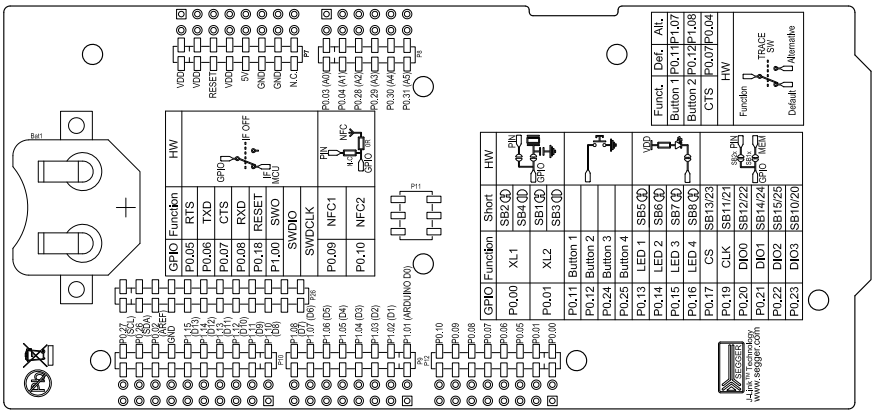


Figure 4: nRF52840 DK back view

4.2 Block diagram

The block diagram illustrates the nRF52840 DK functional architecture.

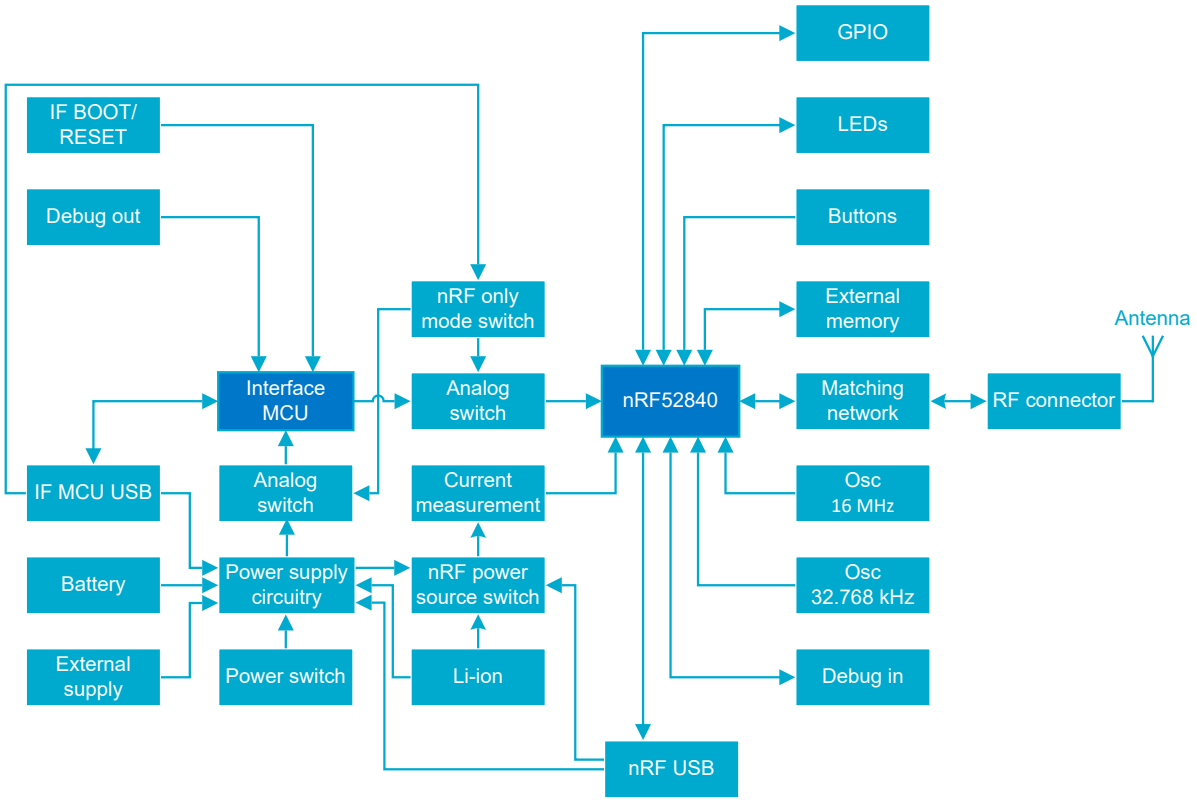


Figure 5: Block diagram

4.3 Power supply

The nRF52840 DK has multiple power options.

The power options are the following:

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

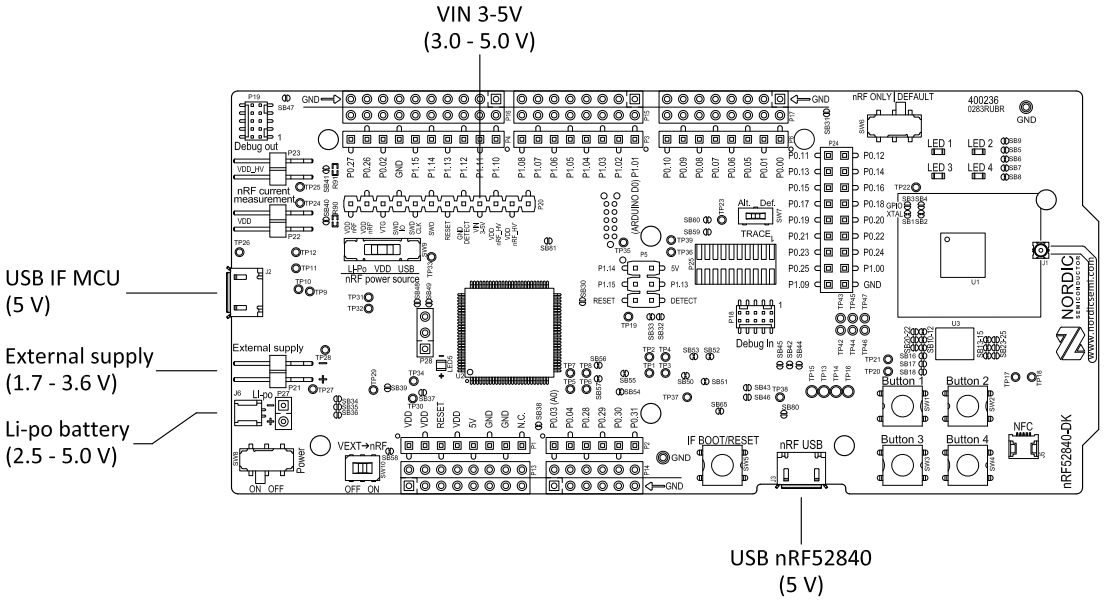


Figure 6: Power supply options (front)

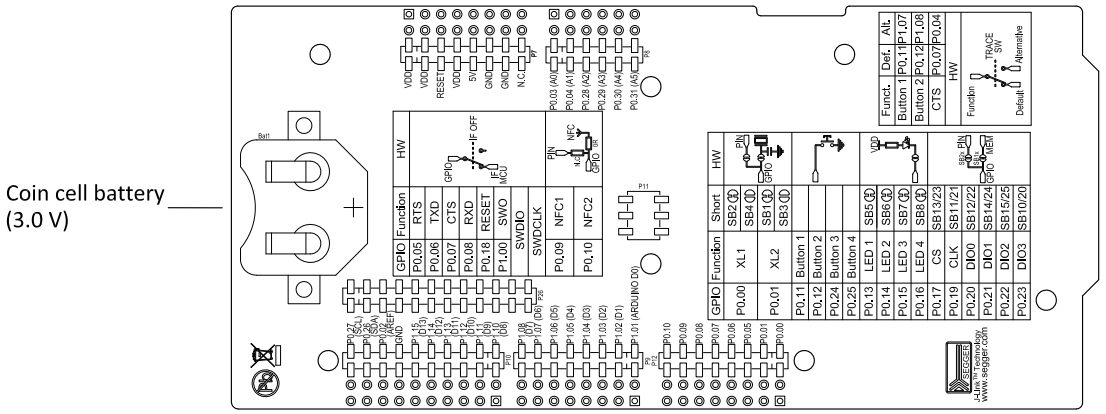


Figure 7: Power supply options (back)

4.3.1 5 V power sources

The nRF52840 DK has a 5 V boost regulator.

It gives a stable 5 V output from the following sources:

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

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

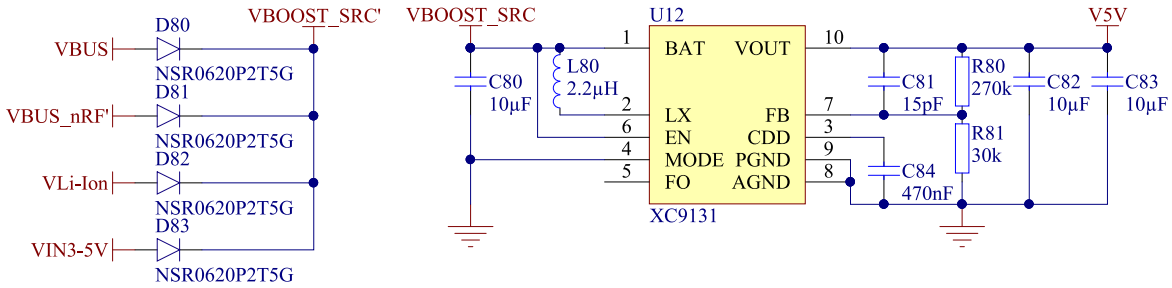


Figure 8: 5 V regulator and protecting diodes

4.3.2 VDD power sources

The main supply voltage (VDD) is 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 nRF52840 power source on page 17.

The power sources are routed through a set of load switches, which are controlled by logic to prioritize the power sources in the correct manner.

If the high voltage regulator of the nRF52840 is used, the *DK* is supplied from the VDD_nRF voltage follower regardless of the state of the other power sources.

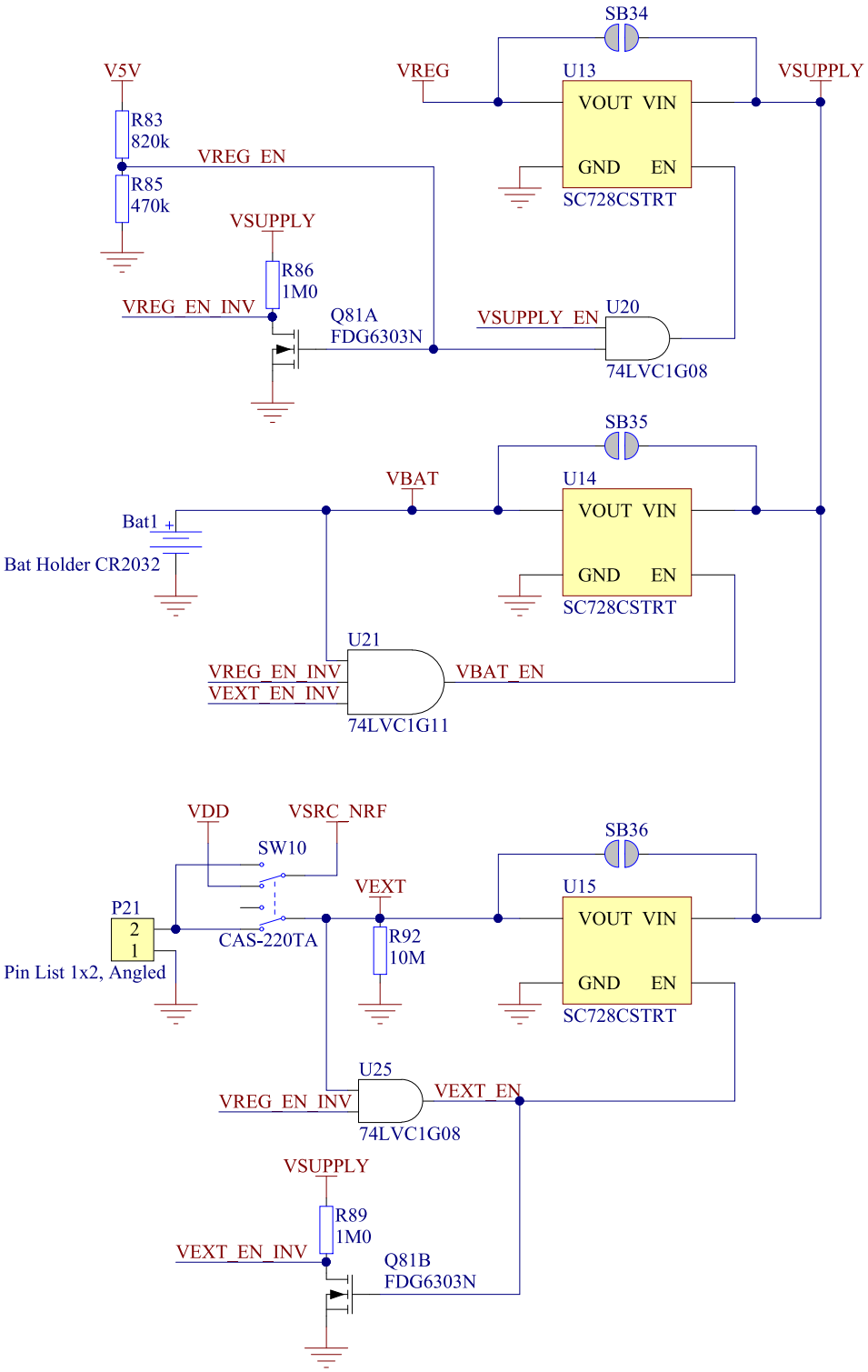


Figure 9: Power supply circuitry

The internal transistor body diode powers the VSUPPLY net, which is used to supply the gates that control the enable signal for the power 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.

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

Table 2: Power switch bypass solder bridges

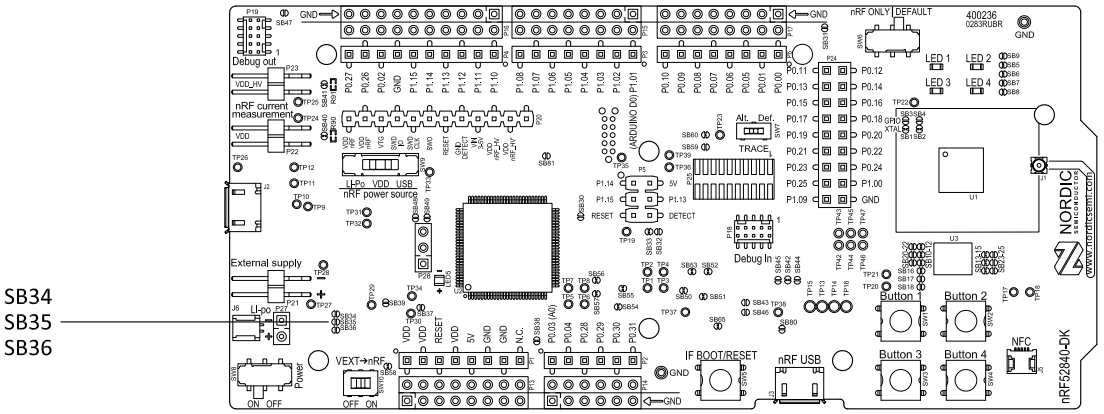


Figure 10: Power switch bypass solder bridges

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

4.3.3 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. This makes it possible to disconnect the interface MCU from the power domain when not in use.

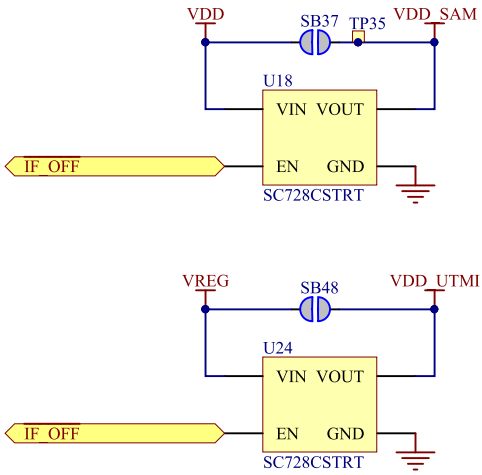


Figure 11: Interface MCU power switch

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 Modes of operation on page 18 for more information.

4.3.4 nRF52840 power source

The nRF52840 DK has a power source switch (**SW9**) for selecting between VDD (default), Li-Po, and USB power supplies for the nRF52840 SoC.

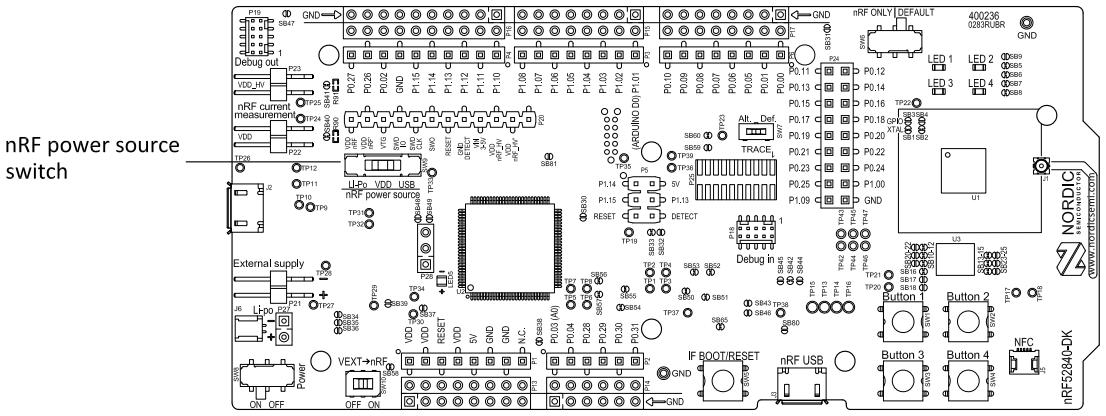


Figure 12: nRF52840 DK power source switch

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 from the onboard 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-Poly* battery connectors (**J6** or **P27**). In the `USB` position, the *USB* high voltage regulator is powered by thenRF52840 USB connector (**J3**).

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

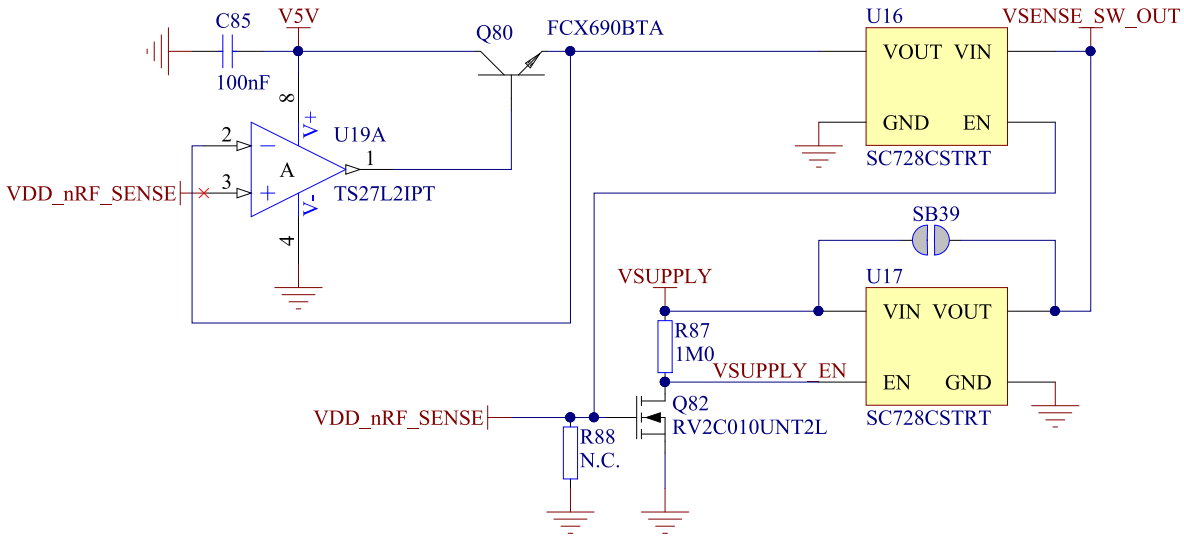


Figure 13: VDD_nRF voltage follower and switch

To make sure that the nRF52840 SoC is not powered when the nRF power switch (**SW8**) is OFF, two load switches are used, one for the high voltage regulator (**U23**) and one for the USB supply (**U22**). These switches are controlled by VDD.

4.3.5 nRF52840 SoC direct supply

It is possible to power the application *SoC* directly from a source without powering the rest of the *DK* from that source.

This is done by connecting the external source to the external supply connector (**P21**) and sliding the **VEXT->nRF** switch (**SW10**) to the ON position. The nRF power source switch (**SW9**) must be in the VDD position. The allowed voltage range is from 1.7 V to 3.6 V.

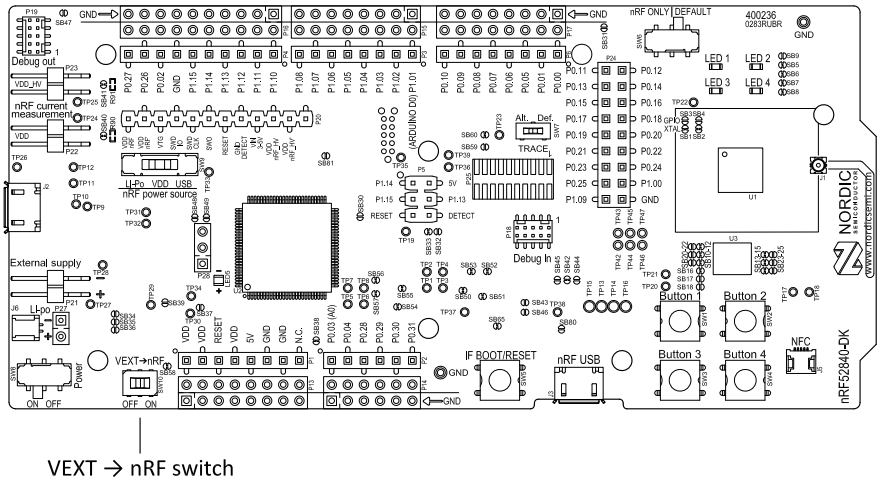


Figure 14: VEXT->nRF switch (SW10)

Since only the nRF52840 SoC is supplied from this source, it is recommended that the VDD domain is supplied from a different source to ensure the pins of the SoC are not connected to unpowered devices.

To avoid voltage differences on the DK, the external supply is also connected to the input of the voltage follower when the **VEXT->nRF** switch (**SW10**) is in the ON position. The voltage follower circuit requires 5 V supplied to the DK, see 5 V power sources on page 13.

The voltage follower can be disconnected from the External supply by cutting **SB58**. To prevent leakage due to voltage differences, the DK should be set in the nRF only mode, see nRF only mode on page 19.

Note: 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 External memory on page 21.

4.4 Modes of operation

The following sections describe signal switches and the various modes of operation of the nRF52840 DK.

4.4.1 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.

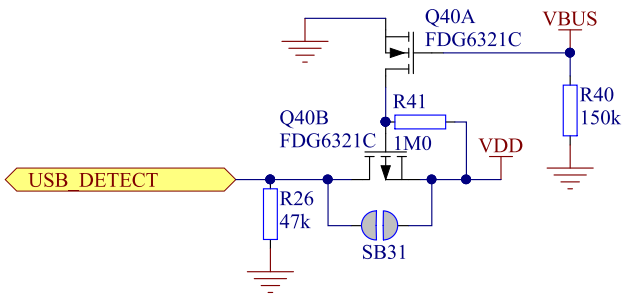


Figure 15: USB detect

4.4.2 nRF only mode

The nRF only mode disconnects the power supply, external memory, and LEDs of the interface MCU. It also disconnects the signal lines between the nRF52840 *SoC* and the interface MCU using analog switches.

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

The power supply of the external memory can be changed to maintain operation in the nRF only mode. See External memory on page 21.

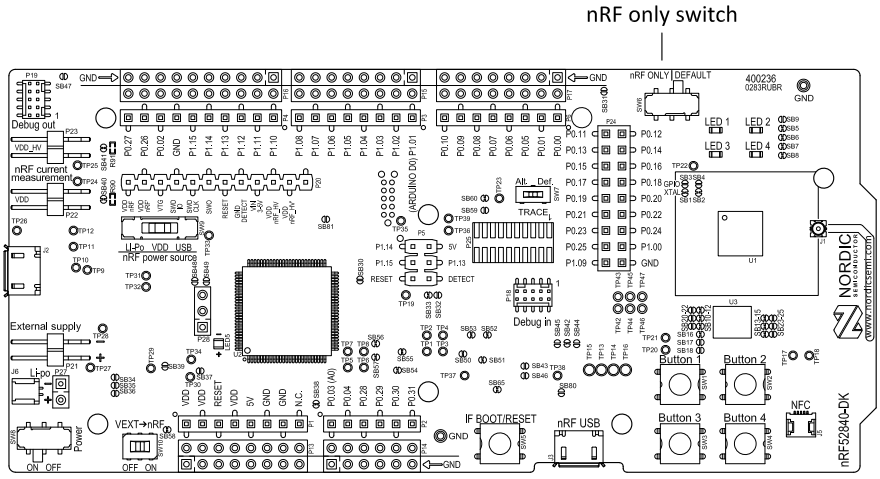


Figure 16: nRF ONLY switch (SW6)

4.4.3 Signal switches

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

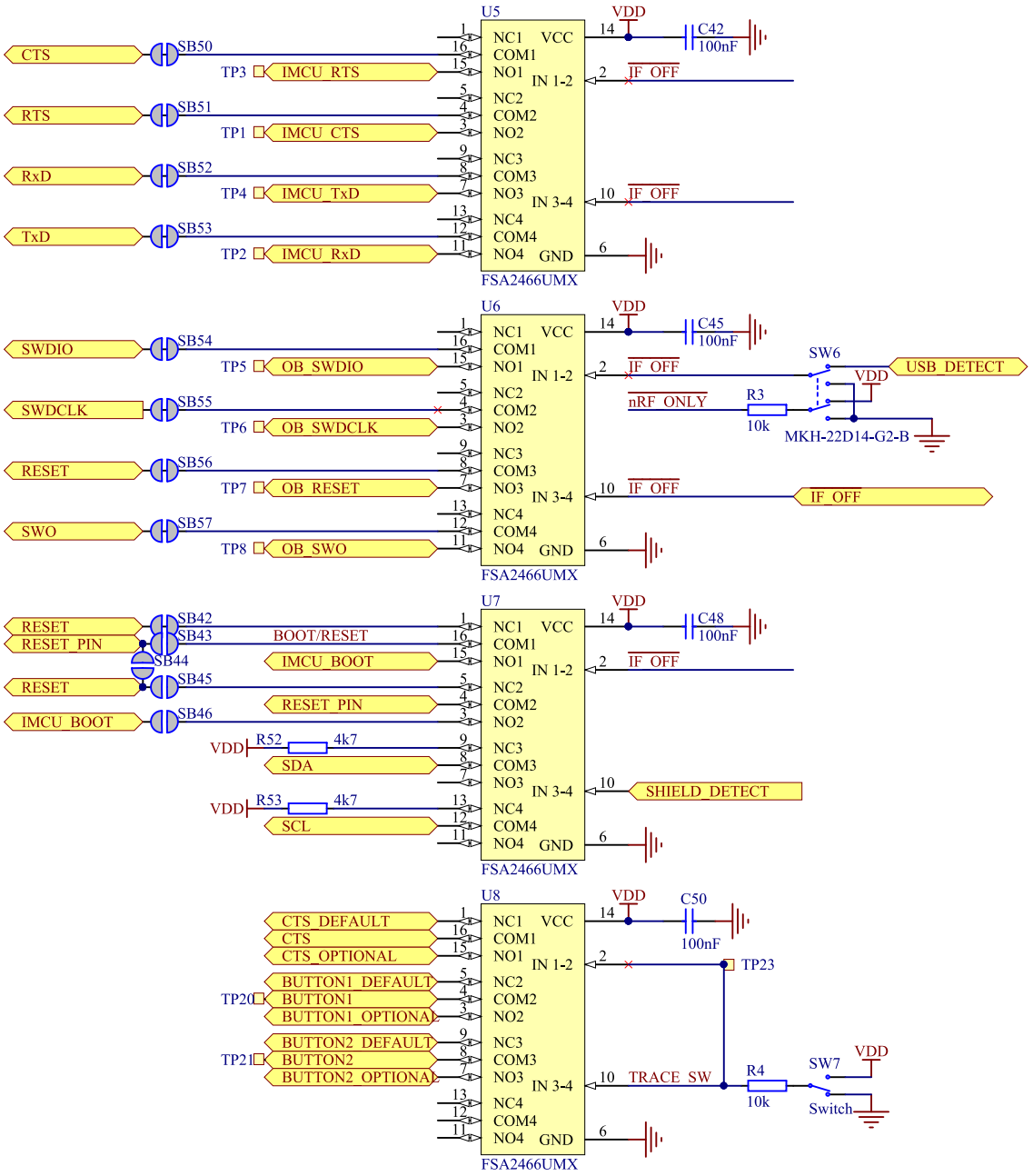


Figure 17: Signal switches

The USB and **SW6** control the signal switches by using USB_DETECT as an input to **SW6**. The interface MCU can be disconnected either by unplugging the USB cable from **J2** or by toggling the **nRF ONLY** switch **SW6**.

The signal controls a set of switches (**U5**, **U6**, **U7**) that break the connection between the nRF52840 SoC and the interface MCU, and control the power for the interface MCU. For more information, see Interface MCU power on page 16.

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

- When the interface MCU is disconnected, cutting **SB42** disconnects the IF BOOT/RESET button (**SW5**) from the reset pin (**P0.18**) of nRF52840.
- When the interface MCU is disconnected, shorting **SB45** connects the RESET pin in the Arduino interface to the reset pin (**P0.18**) of nRF52840.

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

When a shield is connected, two analog switches connect the pull-up resistors to the I2C bus lines (SDA and SCL). This function uses 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**.

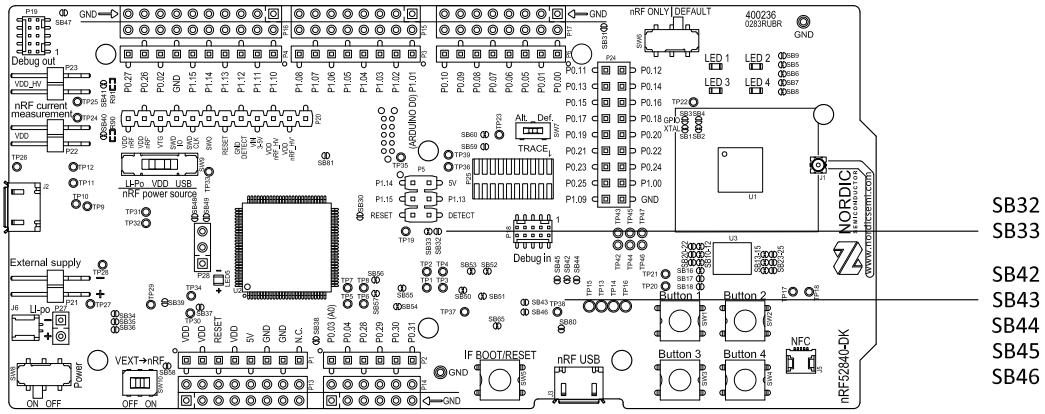


Figure 18: Solder bridges: Shield detect and reset behavior

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 Debug input and trace on page 26 for more information.

4.5 External memory

The nRF52840 DK has a 64 megabit external flash memory. The memory is a multi-I/O memory supporting both *Serial Peripheral Interface (SPI)* and *Quad Serial Peripheral Interface (QSPI)*.

The memory is connected to the chip using the following *General-Purpose Input/Output (GPIO)*s:

GPIO	Flash memory pin	Solder bridge for memory use (default: shorted)	Solder bridge for GPIO use (default: open)
0.17	CS	SB13	SB23
0.19	SCLK	SB11	SB21
0.20	SIO_0/SI	SB12	SB22
0.21	SIO_1/SO	SB14	SB24
0.22	SIO_2/WP	SB15	SB25
0.23	SIO_3/HOLD	SB10	SB20

Table 3: Flash memory GPIO usage and connecting solder bridges

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 the following figure for details.

Note: 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.

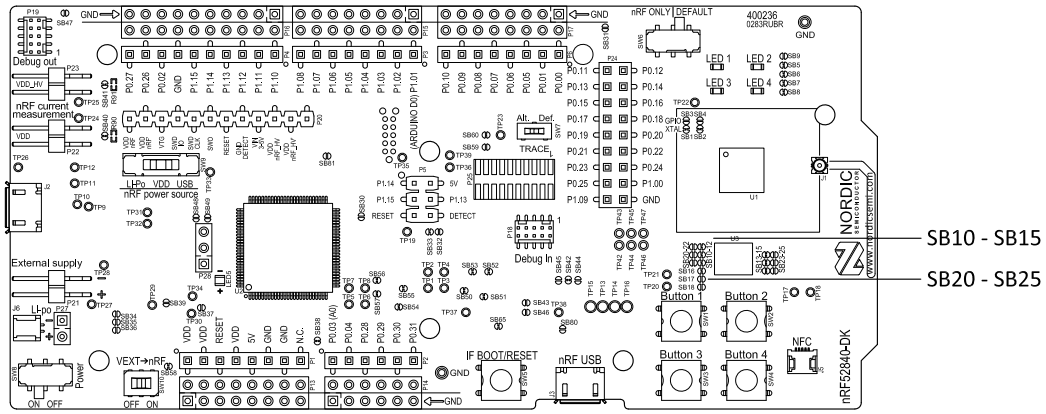


Figure 19: Configuring GPIOs for external memory

By default, the power supply of the external memory is from the VDD domain and it is controlled by the nRF only switch (**SW6**). In the nRF only mode, there are two optional power sources for keeping the external memory powered, VDD and VDD_nRF. If VDD_nRF is selected, the power consumption of the external memory is added to the nRF52840 current measured on **P22** or **P23**. See the following table for configuration.

Power source	Solder bridge	Default state
VDD_PER	SB16	Shorted
VDD	SB17	Open
VDD_nRF	SB18	Open

Table 4: Flash memory power source configuration

4.6 Connector interface

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

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

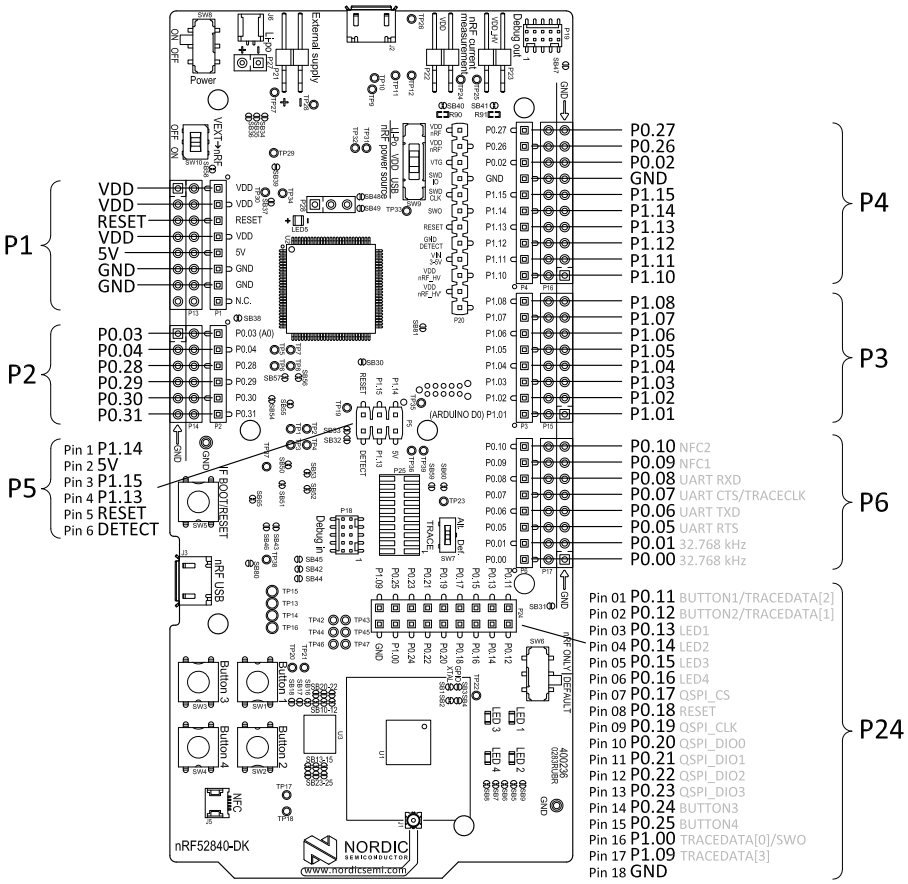


Figure 20: nRF52840 DK connectors

Some of the signals are also available on connectors **P7**, **P8**, **P9**, **P10**, **P11**, and **P12**, located on the back side of the *DK*. By mounting pin lists on the connector footprints, the nRF52840 DK can be used as a shield for 3.3 V Arduino motherboards 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**.

The following are default pin settings:

- **P0.00** and **P0.01** are used for the 32.768 kHz crystal and are not available on the connectors. See 32.768 kHz crystal on page 26 for more information.
- **P0.05**, **P0.06**, **P0.07**, and **P0.08** are used by the *UART* connected to the interface MCU. See Virtual serial port on page 9 for more information.
- **P0.09** and **P0.10** are by default used by signals NFC1 and NFC2. See NFC antenna interface on page 29 for more information.
- **P0.11**–**P0.16** and **P0.24**–**P0.25** are by default connected to the buttons and LEDs. See Buttons and LEDs on page 24 for more information.
- **0.17** and **0.19**–**0.23** are by default connected to the external memory. For more information, see External memory on page 21.

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

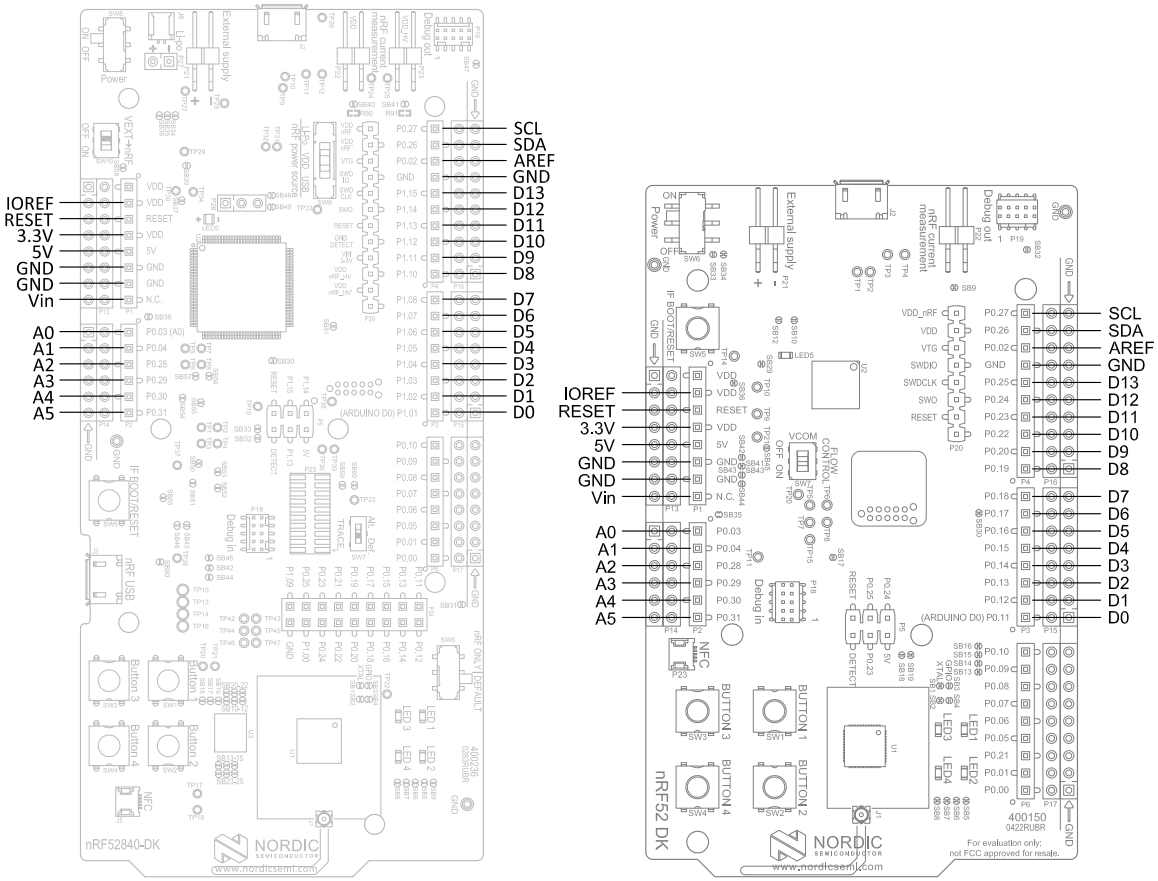


Figure 21: Arduino signals routing on the nRF52840 DK

4.6.1 Analog pin map

The following table shows the mapping between GPIO pins, analog inputs, and the corresponding Arduino analog input naming.

GPIO	Analog input	Arduino naming
P0.03	AIN1	A0
P0.04	AIN2	A1
P0.28	AIN4	A2
P0.29	AIN5	A3
P0.30	AIN6	A4
P0.31	AIN7	A5

Table 5: Mapping of analog pins

4.7 Buttons and LEDs

The four buttons and four LEDs on the nRF52840 DK are connected to dedicated GPIOs on the nRF52840 *SoC*.

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
LED 4	P0.16		SB8

Table 6: Button and LED connections

If **P0.13–P0.16** are needed elsewhere, the LEDs can be disconnected by cutting the short on **SB5–SB8**. See the following figure for more information.

Since **P0.11** and **P0.12** are used as part of the Trace functionality, **BUTTON 1** and **BUTTON 2** can be moved to alternative GPIOs. See Debug input and trace on page 26 for more information.

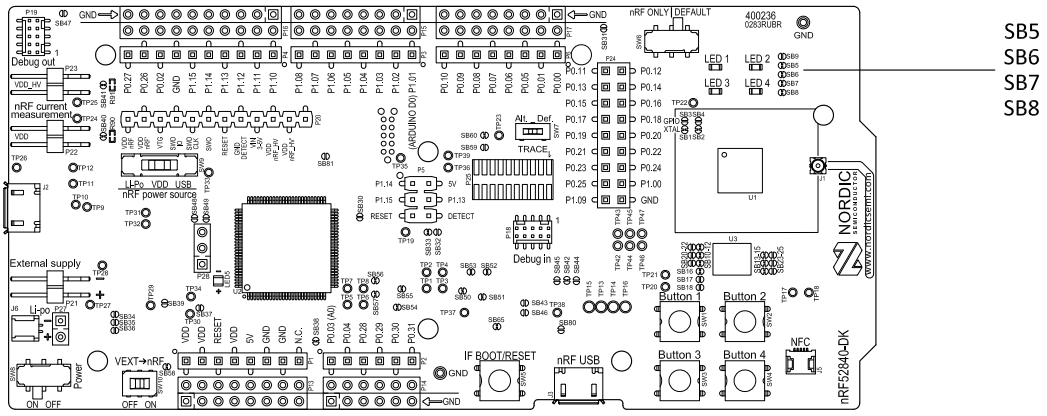


Figure 22: Disconnecting the LEDs

The buttons are active low, which means that input is connected to ground when the button is activated. The buttons do not have an external pull-up resistor, so the **P0.11**, **P0.12**, **P0.24**, and **P0.25** pins must be configured as input with an internal pull-up resistor to use the buttons.

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

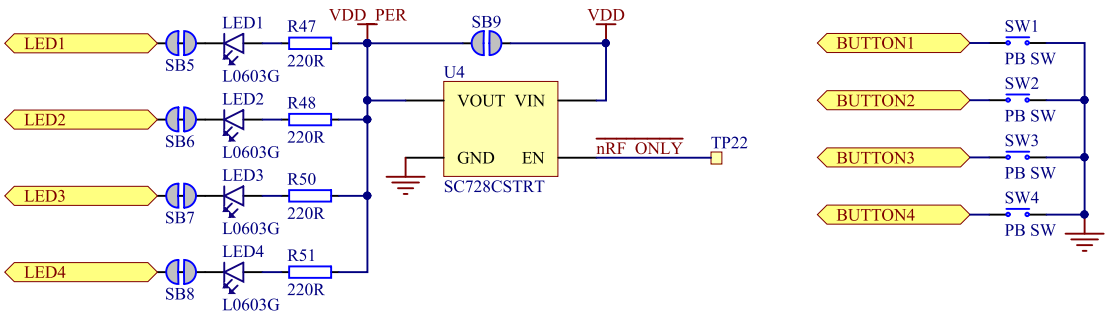


Figure 23: Button and LED configuration

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

Note: 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, then the 32.768 kHz crystal can be disconnected and the GPIO routed to the connectors. Cut the solder bridges on **SB1** and **SB2**, and solder **SB3** and **SB4**. See the following figure for reference.

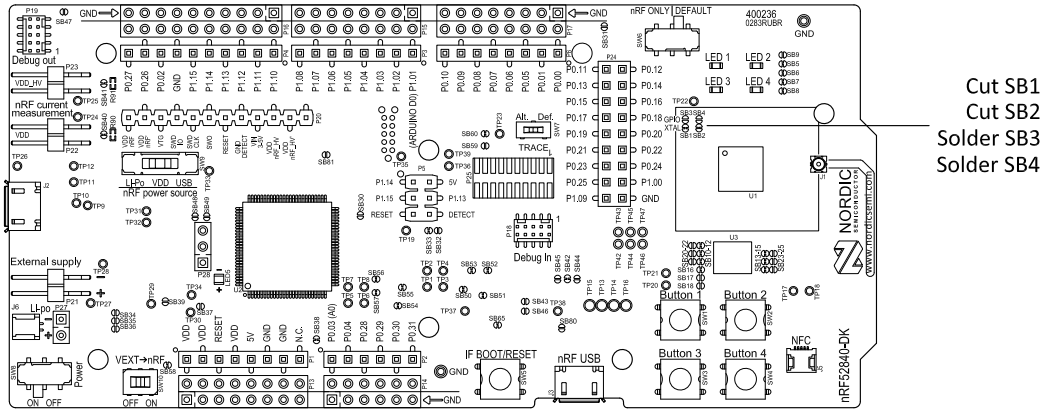


Figure 24: Configuring P0.00 and P0.01

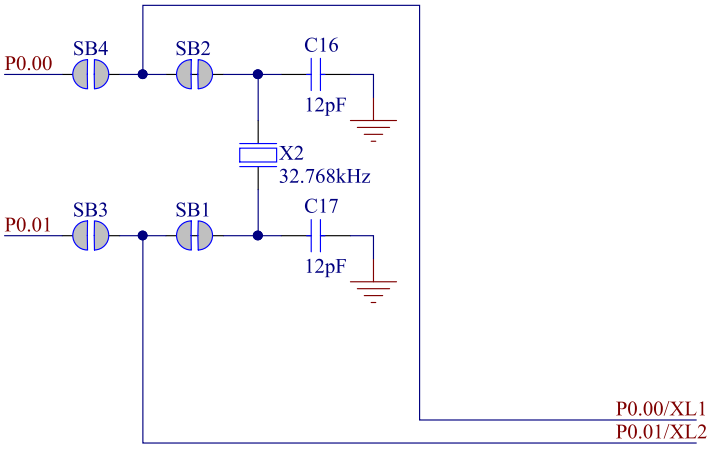


*Figure 25: 32.768 kHz crystal and **SB1–SB4***

4.9 Debug input and trace

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

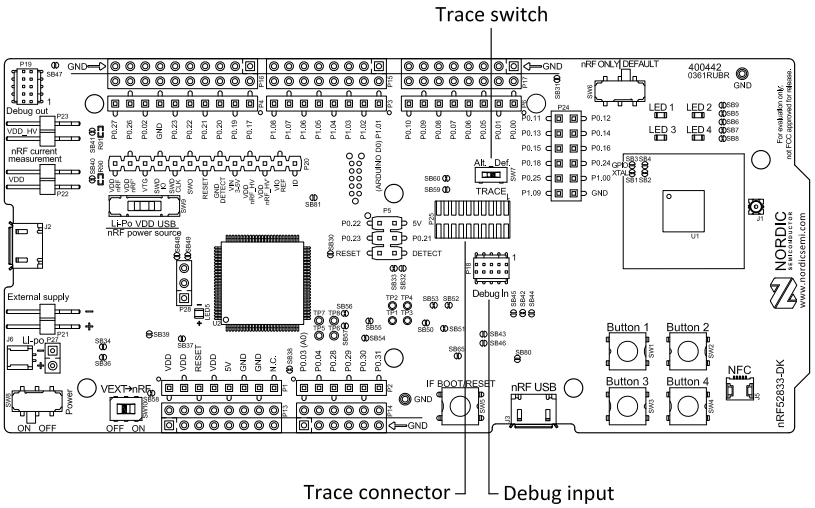


Figure 26: Debug input and trace connectors

For trace, a footprint for a 20-pin connector is available (**P25**). If trace functionality is required, it is possible to mount a 2x10 pin 1.27 mm pitch surface-mount pin header. To use trace, you must disable flow control on **UART_0**, by switching Flow Control (**SW7**) for **serial port 0** to `off`.

Some of the trace pins are by default used for other functionality on the *DK*. By sliding the TRACE switch (**SW7**) from Def. to Alt., the functionality is moved to other GPIOs. See the following table for more information.

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]		

Table 7: Default and Trace GPIOs

By default, the reference voltage for the debug input and trace is connected to VDD_nRF'. This can be connected to the **VDD** by cutting **SB60** and soldering **SB59**.

4.10 Debug output

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

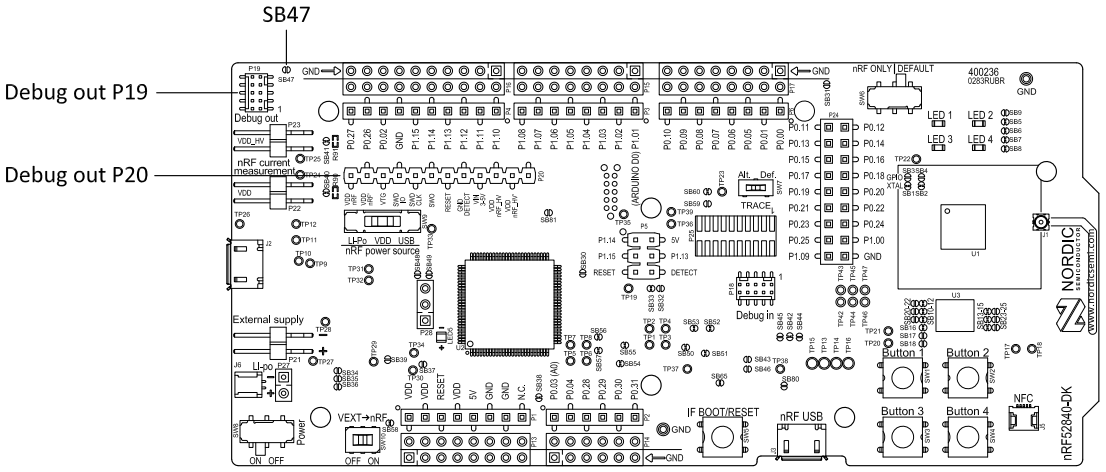


Figure 27: Debug output connector

When the external board is powered, the interface MCU detects the supply voltage of the board and programs/debugs the target chip on the external board instead of the onboard nRF52840 SoC.

> **Note:** The voltage supported by external debugging/programming is VDD voltage. Normally, this is 3 V when running from USB. But if the onboard nRF52840 SoC is supplied from either USB or lithium-ion (Li-ion) and the nRF power source switch (**SW9**) is in either **Li-Po** or **USB** position, then the VDD can be set by the nRF52840 firmware. Make sure the voltage level of the external board matches the VDD of the nRF52840 DK.

You can also use **P20** as a Debug out connection to program shield-mounted targets. For both **P19** and **P20**, the interface MCU detects the supply voltage on the mounted shield and programs or debugs the target.

If the interface MCU detects target power on **P19** and **P20**, it programs or debugs the target connected to **P19** by default.

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

4.10.1 Connectors for programming external boards

The tables describe pinouts of connectors **P19** and **P20**. The voltage on the external board must match that of the *DK*.

Pin number	Signal	Description
1	EXT_VTG	Voltage supply from the external target, used as a select pin for the interface MCU to enable this port
2	EXT_SWDIO	*Serial Wire Debug (SWD)* data line
3	GND	Ground
4	EXT_SWDCLK	SWD clock line
5	GND	Ground
6	EXT_SWO	The *Serial Wire Output (SWO)* line is not needed for programming and debugging over SWD
7	N.C.	Not used
8	N.C.	Not used
9	EXT_GND_DETECT	Ground detect
10	EXT_RESET	Reset line

Table 8: Pinout of connector P19 for programming external targets

Pin number	Signal	Description
1	VDD_nRF	nRF52840 SoC power domain
2	VDD_nRF’	Main nRF52840 DK power domain
3	SH_VTG	Voltage supply from the external target, used as a select pin for the interface MCU to enable this port
4	SH_SWDIO	*SWD* data line
5	SH_SWDCLK	SWD clock line
6	SH_SWO	The *SWO* line is not needed for programming and debugging over SWD
7	SH_RESET	Reset line
8	SH_GND_DETECT	Ground detect

Table 9: Pinout of connector P20 for programming external targets

4.11 NFC antenna interface

The nRF52840 DK supports an *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 DK.

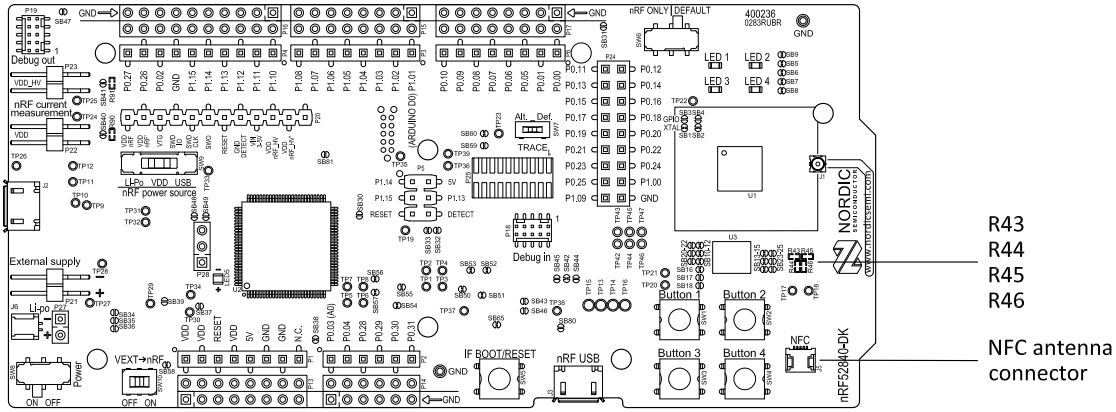


Figure 28: NFC antenna connector

NFC uses two pins, NFC1 and NFC2, to connect the antenna. These pins are shared with *GPIOs* (`P0.09` and `P0.10`). The PROTECT field of the NFCPINS register in *User Information Configuration Registers (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.

Configuring NFC pins as GPIOs

The NFC pins are enabled by default. NFC can be disabled and GPIOs enabled by defining the CONFIG_NFCT_PINS_AS_GPIOS variable in the project settings. This is configured based on the *IDE* or toolchain in use:

- SEGGER Embedded Studio – Select **Project** > **Edit Options** > **Code** > **Preprocessor** > **Preprocessor Definitions** and add the CONFIG_NFCT_PINS_AS_GPIOS variable.
- nRF Connect SDK – Set CONFIG_NFCT_PINS_AS_GPIOS to `y`. See Configuring and building for instructions.

Pins `L24` and `J24` are by default configured to use the NFC antenna. To use these pins as GPIO, **R44** and **R46** must be not connected (NC) and **R43** and **R45** must be shorted with an OR resistor.

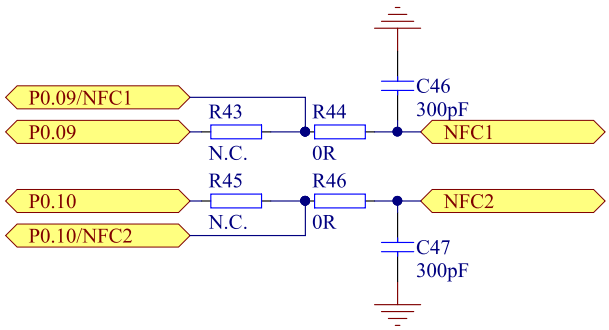


Figure 29: NFC input

4.12 Extra op-amp

The voltage follower for the power supply uses a dual package *Operational Amplifier (op-amp)*.

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

For more information on the power supply, see Power supply on page 12.

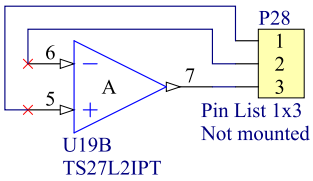


Figure 30: Extra op-amp

4.13 Solder bridge configuration

The nRF52840 DK has a range of solder bridges for enabling or disabling functionality on the *DK*. Changes to these are not needed for normal use of the *DK*.

The following table is an overview of the solder bridges on the nRF52840 DK.

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 peripheral 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 P0.17
SB14	Closed	Cut to disconnect the QSPI memory from P0.21
SB15	Closed	Cut to disconnect the QSPI memory from P0.22
SB16	Closed	Cut to disconnect QSPI memory power supply from VDD_PER
SB17	Open	Short to connect QSPI memory power supply to VDD
SB18	Open	Short to connect QSPI memory power supply to VDD_nRF
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 P0.17 as a normal GPIO
SB24	Open	Short to enable P0.21 as a normal GPIO
SB25	Open	Short to enable P0.22 as a normal GPIO

Solderbridge	Default	Function
SB30	Open	Short to reset the interface MCU
SB31	Open	Short to bypass the USB detect switch
SB32	Open	Short to permanently enable the I^2C pull-up resistors
SB33	Closed	Cut to permanently disable the I^2C 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	Closed	Cut to disconnect IF Boot/Reset button from 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 interface 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	Closed	Cut to disconnect the nRF52840 CTS line from the signal switch and interface MCU
SB51	Closed	Cut to disconnect the nRF52840 RTS line from the signal switch and interface MCU
SB52	Closed	Cut to disconnect the nRF52840 RxD line from the signal switch and the interface MCU
SB53	Closed	Cut to disconnect the nRF52840 TxD line from the signal switch and interface MCU
SB54	Closed	Cut to disconnect the nRF52840 SWDIO line from the signal switch and interface MCU

Solderbridge	Default	Function
SB55	Closed	Cut to disconnect the nRF52840 SWDCLK line from the signal switch and interface MCU
SB56	Closed	Cut to disconnect the nRF52840 RESET line from the signal switch and interface MCU
SB57	Closed	Cut to disconnect the nRF52840 SWO line from the signal switch and the interface MCU
SB58	Closed	Cut to disconnect voltage follower from external supply when SW10 is in ON position
SB59	Open	Solder to connect debug in and trace reference voltage to VDD
SB60	Closed	Cut to disconnect debug in and trace reference voltage from VDD_nRF'
SB65	Closed	Cut to disable the pull-up resistor of the IMCU_BOOT line
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

Table 10: Solder bridge configuration

The following table is a complete overview of the test points on the nRF52840 DK.

5 Current measurement

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

Current can be measured using any of the following test instruments:

- Oscilloscope
- Ampere meter
- Power Profiler Kit II (PPK2)
- Power analyzer

See the following chapter for important information on the DK measurement setup. If the PPK2 will be used for measuring current, see the Power Profiler Kit II for additional instructions. Power analyzer measurements are not described in this document.

The application nRF52840 SoC has two possible power supplies: VDD (1.7 V to 3.6 V) and VDDH (2.5 V to 5.5 V). The nRF52840 DK can measure 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 following table for the corresponding components.

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

Table 11: Components for current measurement on VDD and VDDH

It is not recommended to use a USB connector to power the *DK* during current measurements due to potential noise from the USB power supply. However, when measuring current on an application using the USB interface of the nRF52840 SoC, USB must be connected. It is recommended to power the DK from a coin cell battery, external power supply on connector **P21** (1.7 V to 3.6 V) or through the *Li-Poly* connector **J6** or **P27** (2.5 V to 5.0 V).

> **Note:** The current measurements are unreliable if a serial terminal is connected to the virtual serial port.

After programming the nRF52840 SoC, disconnect the USB for the interface MCU.

For more information on current measurement, see the tutorial Current measurement guide: Introduction.

5.1 Set up the DK

To measure current, you must first prepare the *DK*.

The suggested configurations split the power domains for the application nRF52840 *SoC* and the rest of the DK.

- To put **P22** in series with the load, cut the PCB track shorting solder bridge **SB40**.
- To restore normal kit function after measurement, solder **SB40** or apply a jumper on **P22**.

- To reprogram the nRF52840 SoC while the DK is prepared for current measurements, remove measurement devices from **P22**, and then connect the USB cable.

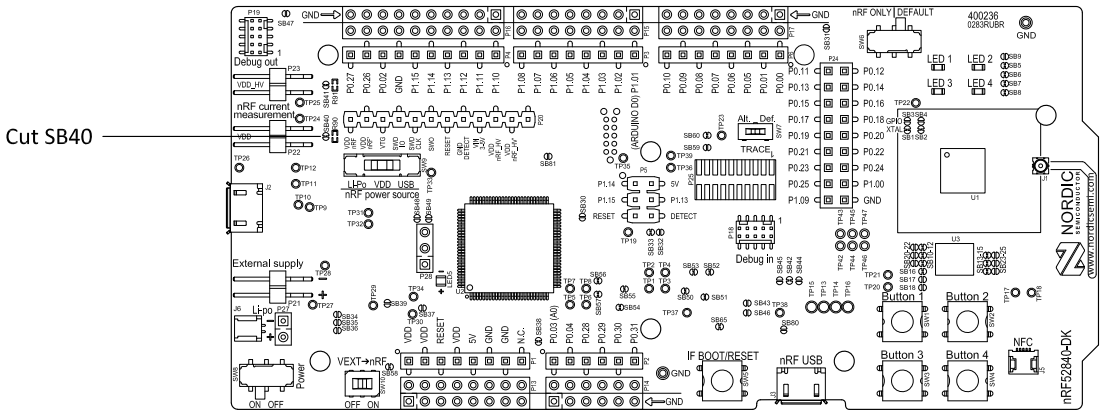


Figure 31: Preparing the DK for current measurements

5.2 Measure current profile with an oscilloscope

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

Make sure you have prepared the *DK* as described in Set up the DK on page 34.

1. Mount a 10 Ω resistor on the footprint for **R90**.
2. Set the oscilloscope to differential mode or a mode that is similar.
3. Connect the oscilloscope using two probes on the pins of the **P22** connector, as shown in the following figure.
4. 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 is proportional to the current. The 10 Ω resistor causes 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. The current can then be averaged or integrated to analyze current and energy consumption over a period.

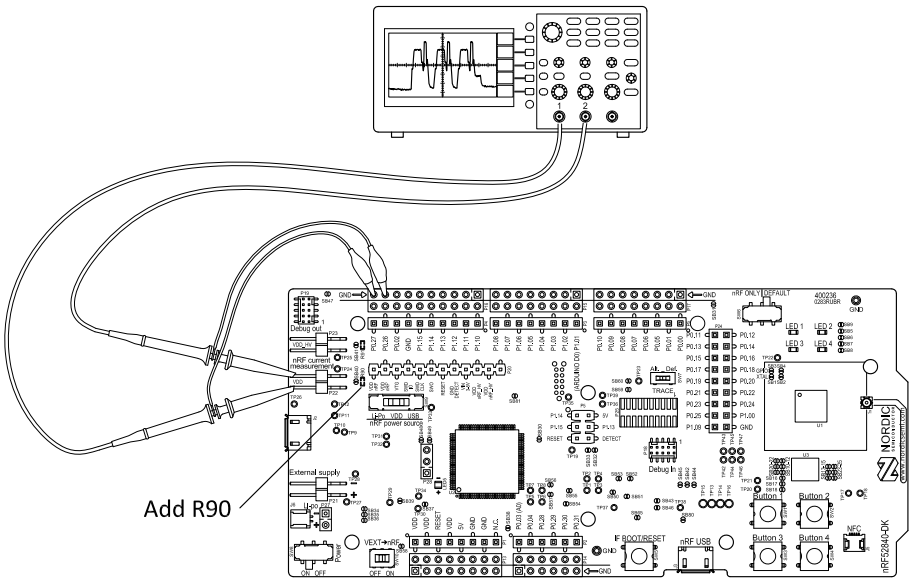


Figure 32: Current measurement with an oscilloscope

To reduce noise, do the following:

- 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.

5.3 Measure average current with an ampere meter

The average current drawn by the nRF52840 *SoC* can be measured using an ampere meter. This method monitors the current in series with the nRF device. A true ampere meter is recommended.

Make sure you have prepared the *DK* as described in Set up the DK on page 34.

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

To measure the average current drawn by the nRF52840 SoC, perform the following steps.:

- Set the average timing of the ampere meter to a long interval, such as 1 s or longer.
- Set the dynamic range of the ampere meter between 1 uA and 15 mA, so that it is wide enough to provide accurate measurements.

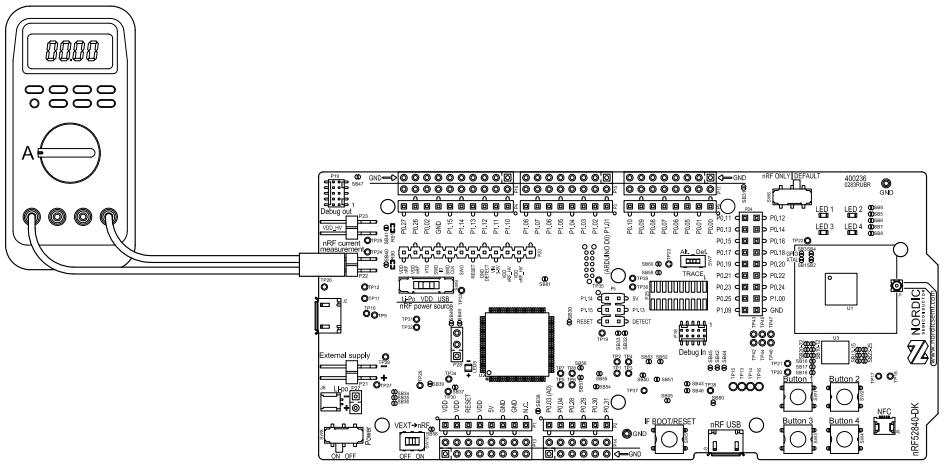


Figure 33: Current measurement with an ampere meter

6 RF measurements

The nRF52840 DK is equipped with a small coaxial connector (**J1**) for conducting measurements of the RF signal with a spectrum analyzer.

The connector is of *Microwave coaxial connector with switch (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 onboard trace antenna.

In this example, a test probe (Murata part no. MXHS83QE3000) is used with a standard *SubMiniature Version A (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 disconnects the onboard antenna and connects the RF signal from the nRF52840 *SoC* to the test probe.

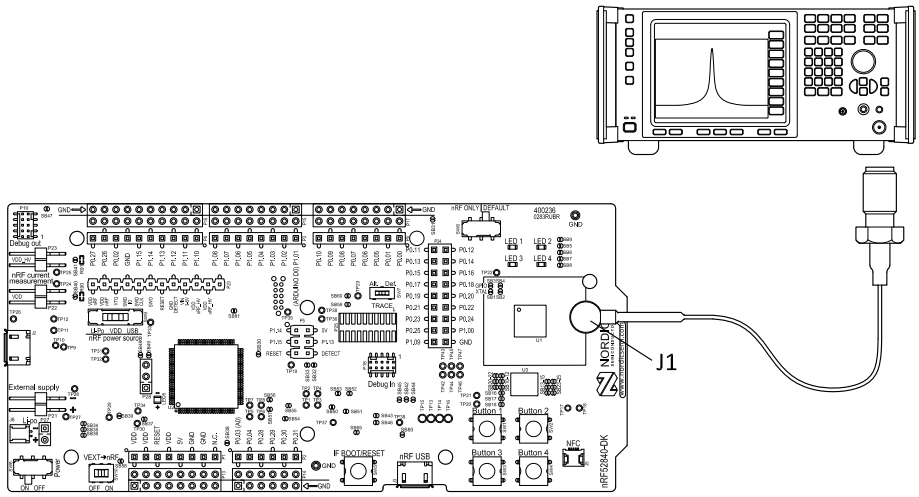


Figure 34: Connecting a spectrum analyzer

The connector and test probe add loss to the RF signal. See the following table for more information or consult the test probe user guide if you are using another model.

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

Table 12: Typical loss in connector and test probe, using Murata part no. MXHS83QE3000

Glossary

Clear to Send (CTS)

In flow control, the receiving end is ready and telling the far end to start sending.

Data Terminal Ready (DTR)

A control signal in RS-232 serial communications transmitted from data terminal equipment, such as a computer, to data communications equipment.

Development Kit (DK)

A hardware development platform used for application development.

Electrostatic Discharge (ESD)

A sudden discharge of electric current between two electrically charged objects.

General-Purpose Input/Output (GPIO)

A digital signal pin that can be used as input, output, or both. It is uncommitted and can be controlled by the user at runtime.

Hardware Flow Control (HWFC)

A handshaking mechanism used to prevent an overflow of bytes in modems. It uses two dedicated pins on the RS-232 connector, Request to Send and Clear to Send.

Integrated Circuit (IC)

A semiconductor chip consisting of fabricated transistors, resistors, and capacitors.

Integrated Development Environment (IDE)

A software application that provides facilities for software development.

Lithium-polymer (Li-Poly)

A rechargeable battery of lithium-ion technology using a polymer electrolyte instead of a liquid electrolyte.

Mass Storage Device (MSD)

Any storage device that makes it possible to store and port large amounts of data in a permanent and machine-readable fashion.

Near Field Communication (NFC)

A standards-based short-range wireless connectivity technology that enables two electronic devices to establish communication by bringing them close to each other.

NFC-A Listen Mode

Initial mode of an NFC Forum Device when it does not generate a carrier. The device listens for the remote field of another device. See Near Field Communication (NFC) on page 38.

Onboard (OB)

A function that is delivered on the chip microcontroller.

Operational Amplifier (op-amp)

A high-gain voltage amplifier that has a differential input and, usually, a single output.

Printed Circuit Board (PCB)

A board that connects electronic components.

Quad Serial Peripheral Interface (QSPI)

A Serial Peripheral Interface (SPI) controller that allows the use of multiple data lines.

Receive Data (RXD)

A signal line in a serial interface that receives data from another device.

Request to Send (RTS)

In flow control, the transmitting end is ready and requesting the far end for a permission to transfer data.

Root Mean Square (RMS)

An RMS meter calculates the equivalent Direct Current (DC) value of an Alternating Current (AC) waveform. A true RMS meter can accurately measure both pure waves and the more complex nonsinusoidal waves.

System in Package (SiP)

Several integrated circuits, often from different technologies, enclosed in a single module that performs as a system or subsystem.

Serial Peripheral Interface (SPI)

Synchronous serial communication interface specification used for short-distance communication.

SubMiniature Version A (SMA)

A semi-precision coaxial RF connector for coaxial cables with a screw-type coupling mechanism.

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.

Serial Wire Debug (SWD)

A standard two-wire interface for programming and debugging Arm® CPUs.

Microwave coaxial connector with switch (SWF)

A small, RF surface-mount switch connector series for wireless applications.

Serial Wire Output (SWO)

A data line for tracing and logging.

Transmit Data (TXD)

A signal line in a serial interface that transmits data to another device.

Universal Asynchronous Receiver/Transmitter (UART)

A hardware device for asynchronous serial communication between devices.

User Information Configuration Registers (UICR)

Non-volatile memory registers used to configure user-specific settings.

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 consult other documents.

Nordic documentation

- nRF52840 Product Specification
- nRF52840 Compatibility Matrix
- nRF5 SDK v17.1.0
- nRF52840 Errata

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

© 2025 Nordic Semiconductor ASA. All rights are reserved. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

