nRF9160 DK Hardware

v1.0.0

User Guide

4418_1216 / 2021-10-26

NORDIC®
SEMICONDUCTOR

Contents

Revision history

Date	Description
2021-10-26	• Updated nRF9160 DK board control on page 20 • Editorial changes

November 2020	Updated to match nRF9160 DK v0.15.0 and nRF9160 DK v1.0.0
	• Updated:
	  • Kit content on page 7
	  • Figure 7: nRF9160 DK block diagram on page 13
	  • Firmware development mode on page 8
	  • Performance measurement mode on page 10
	  • Hardware description on page 12
	  • GPS on page 17
	  • nRF9160 DK board control on page 20
	• New:
	  • I/O expander on page 25
	  • External memory on page 27
	  • Switches for I/O expander on page 35
	  • Switches for external memory on page 34

July 2020	Updated document title
April 2020	Editorial changes
March 2020	• Updated Recommended reading on page 48 • Updated Debug output on page 28 • Added Connectors for programming external boards on page 29

October 2019	Updated to match nRF9160 DK v0.9.0
	• Updated:
	  • Kit content on page 7
	  • Recommended reading on page 48
	  • Firmware development mode on page 8
	  • Performance measurement mode on page 10
	  • Block diagram on page 12
	  • Hardware figures on page 12
	  • Power supply on page 13
	  • Antenna interfaces on page 17
	  • GPS on page 17
	  • nRF9160 DK board control on page 20
	  • SIM and eSIM on page 35
	  • Additional interfaces on page 36
	  • Solder bridge configuration on page 37
	  • Preparing the development kit for current measurements on page 39
	• New:

Date	Description
	• Interface MCU disconnect switches on page 31 • Switches for UART interface on page 31 • Switches for nRF52840 interface on page 33 • Removed: • Getting started
July 2019	Updated: • nRF9160 supply on page 16
March 2019	Updated to match nRF9160 DK v0.8.5: • Kit content on page 7 • nRF9160 supply on page 16 • Antenna interfaces on page 17 • GPS on page 17 • nRF9160 DK board control on page 20 • Solder bridge configuration on page 37 • Preparing the development kit for current measurements on page 39 • RF measurements on page 42 • Radiated performance of nRF9160 DK on page 43
December 2018	Updated to match nRF9160 DK v0.8.2
November 2018	Preview DK changed into DK
October 2018	First release

Previous versions

PDF files for the previous versions are available here:

- nRF9160 DK Hardware User Guide v0.9.3
- nRF9160 DK User Guide v0.9
- nRF9160 DK User Guide v0.7.2
- nRF9160 DK User Guide v0.7.1
- nRF9160 DK User Guide v0.7

1 Introduction

The nRF9160 DK is a hardware development platform used to design and develop application firmware on the nRF9160 LTE *Cat-M1* and *Cat-NB1* capable *System in Package (SiP)*.

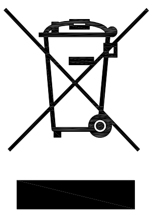
The *Development Kit (DK)* includes all necessary external circuitry like a SIM card holder and an antenna and it provides developers access to all I/O pins and relevant module interfaces.

> **Note:** The *DK* can be connected to a base station by the onboard LTE antenna or to an LTE emulator by an RF cable. See our nRF9160 certifications webpage to find the bands supported by the LTE modem. Without a UICC/SIM connected, the *DK* cannot initiate communication with the LTE network.

The key features of the *DK* are:

- nRF9160 *SiP*
- LTE antenna that supports all bands supported by the *SiP*
- *Global Positioning System (GPS)* antenna
- nRF52840 as a board controller and network processor for *Bluetooth*® and IEEE 802.15.4 protocols
- Buttons, switches, 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 connection for debug/programming and power
- SIM card socket for nano-SIM (4FF SIM)
- Interfaces for nRF9160 current consumption measurements

> **Note:** nRF9160 DK is compliant with the PS1 classification according to the IEC 62368-1 standard.



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.

Skilled person: Person with relevant education or experience to enable him or her to identify hazards and to take appropriate actions to reduce the risks of injury to themselves and others.

2 Kit content

The nRF9160 DK includes hardware, preprogrammed firmware, documentation, hardware schematics, and layout files.

The nRF9160 DK (PCA10090) comes with a *GPS* antenna and a *Subscriber Identity Module (SIM)* card.

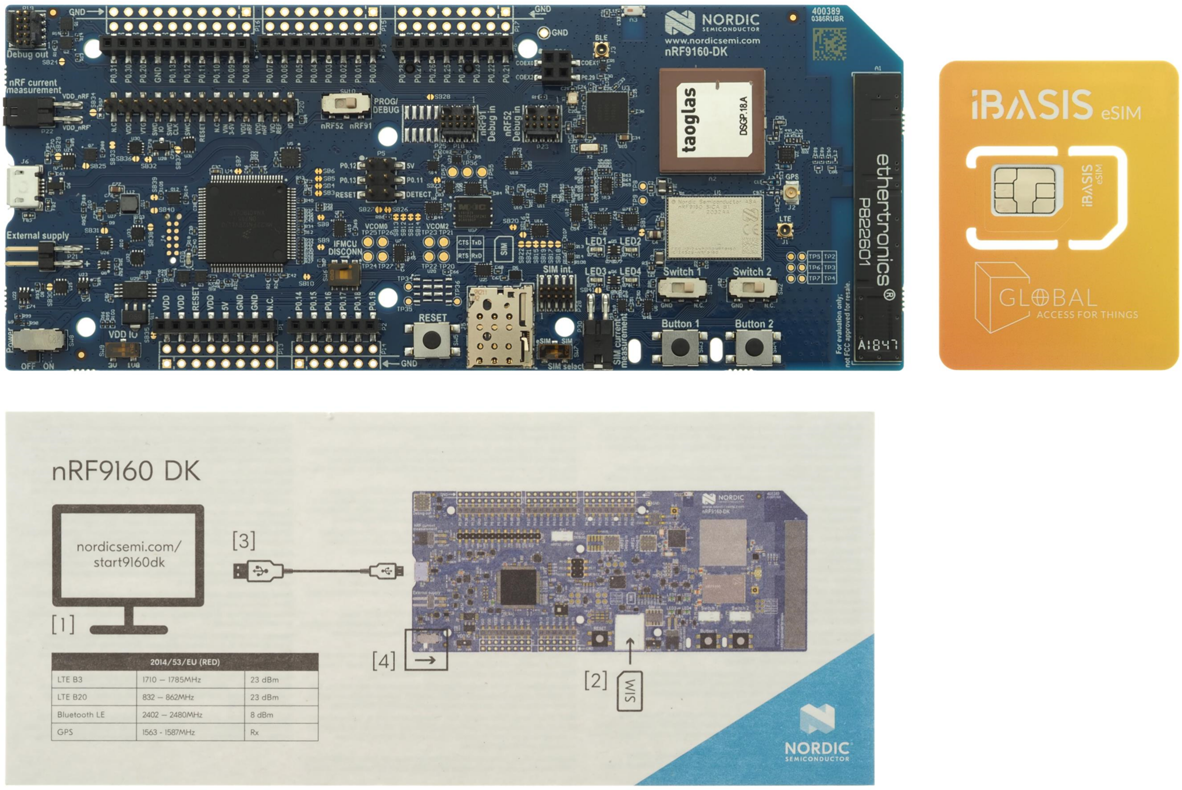


Figure 1: nRF9160 DK kit content

Hardware files

The hardware design files including schematics, *Printed Circuit Board (PCB)* layout files, bill of materials, and Gerber files for the nRF9160 DK are available on the product page, nRF91 SiP Series.

3 Operating modes

The nRF9160 DK has various modes of operation.

3.1 Firmware development mode

The firmware development mode is the default with the IFMCU DISCONN switch (**SW6**) in the left position.

The primary interface for programming and debugging the nRF9160 DK is the USB port (**J6**). The USB port is connected to an interface MCU which embeds a SEGGER J-Link-OB (On Board) debug probe.

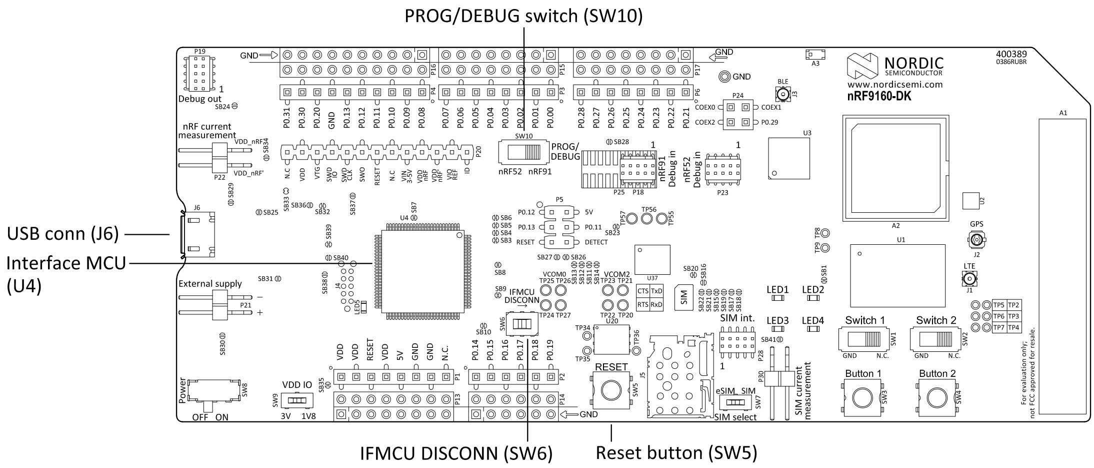


Figure 2: nRF9160 DK firmware development mode

3.1.1 Device programming

The nRF9160 DK supports SWD programming interfaces for both onboard and off-board nRF targets.

The primary target for programming and debugging in the *DK* is the nRF9160. The interface MCU also supports programming the onboard nRF52840 as well as external nRF devices fitted on a shield or through a connector to external boards such as the user's own prototypes.

The interface MCU will automatically detect if external targets are plugged in. The PROG/DEBUG switch (**SW10**) is used to select nRF9160 or nRF52840 for programming or debugging on board. Note that there are significant limitations on using the nRF52840 on the nRF9160 DK. For more information, see nRF9160 DK board control on page 20.

3.1.2 Virtual COM port

The interface MCU also features three UART interfaces through three virtual COM ports.

The virtual COM ports are the following:

- VCOM0 – Connected to nRF9160 (default)
- VCOM1 – Connected to nRF52840 (nonconfigurable)
- VCOM2 – Connected to nRF9160 (default)

For details on routing VCOM0 and VCOM2, see nRF9160 DK board control on page 20.

The virtual COM ports have the following features:

- Flexible baud rate settings up to 1 Mbps
- RTS/CTS-style *Hardware Flow Control (HWFC)* handling

The following table shows an overview of the GPIOs used for the UART connections on the nRF9160 and nRF52840:

	nRF91 APP1	nRF91 APP2	nRF52840
TXD	P0.29	P0.01	P0.03
RXD	P0.28	P0.00	P0.05
CTS	P0.26	P0.15	P0.07
RTS	P0.27	P0.14	P1.08

Table 1: GPIOs used for virtual COM ports on nRF9160 and nRF52840

Note:

- Baud rate 921 600 is not supported through the virtual COM port.

3.1.3 MSD

The interface MCU features a mass storage device (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 this drive. By copying a HEX file to the drive, the interface MCU will program the file to the device. The J-Link OB will program the target that is selected with the nRF52/nRF91 switch (**SW10**). For limitations on nRF52840 use, see nRF9160 DK board control on page 20.

Note:

- Windows might try to defragment the MSD part of the interface MCU. If this happens, the interface MCU will disconnect and become unresponsive. To return to normal operation, the *DK* must be power cycled.
- Your antivirus software might try to scan the MSD part of the interface MCU. Some antivirus programs trigger a false positive alert in one of the files and quarantine the unit. If this happens, the interface MCU will become unresponsive.
- If your computer is set up to boot from USB, it can try to boot from the *DK* if the *DK* is connected during boot. This can be avoided by unplugging the *DK* 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 *DK* and stay this way until changed again.

3.1.4 Reset

The nRF9160 DK is equipped with a RESET button (**SW5**).

By default, the RESET button is connected to the interface MCU that will forward the reset signal to the nRF9160 or nRF52840, depending on the state of the nRF52/nRF91 switch. If IF MCU DISCONNECT is activated, the RESET button will be connected to the nRF9160 directly.

3.2 Performance measurement mode

The performance measurement mode can be selected by moving the IFMCU DISCONN switch (**SW6**) to the right position, which disconnects the interface MCU from the nRF9160 *SiP* using analog switches.

This is done to isolate the nRF9160 *SiP* as much as possible and can be of use when measuring currents on low-power applications.

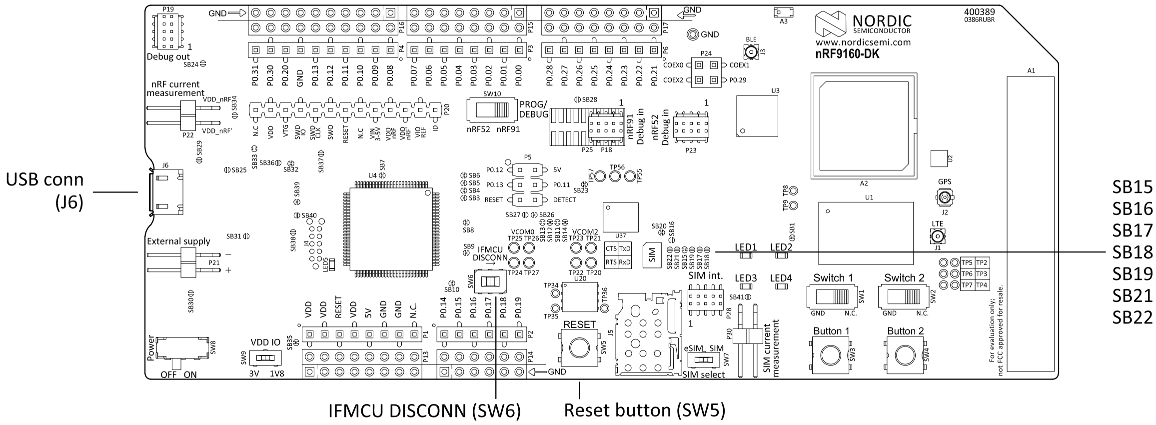


Figure 3: nRF9160 DK performance measurement mode

The *DK* detects if there is a USB cable plugged in (see USB detect on page 10) and will disconnect and power down the interface MCU when another supply than the USB is used. If the USB connector is used for power supply only (USB battery back) you can also disable the interface MCU using **SW6**.

USB detect and **SW6** also control the routing of the RESET signal from the RESET button (**SW5**). Normally, it is routed through the interface MCU, but if that is disabled, the reset button will be routed directly to the nRF9160 SiP.

There are also a number of other reset routing options available through the use of solder bridges:

- When the interface MCU is disconnected, the RESET button is connected to pin 32 (nRESET) of the nRF9160 SiP. The RESET button can be disconnected from the nRF9160 SiP by cutting **SB15**.
- When the interface MCU is disconnected, shorting **SB18** will connect the RESET pin of the Arduino interface to the reset pin of the nRF9160 SiP.
- When the interface MCU is connected, shorting **SB19** will connect the RESET pin of the Arduino interface to the BOOT input of the interface MCU.
- Shorting **SB16** will connect the RESET pin of the Arduino interface to the RESET button.
- Shorting **SB17** will connect the RESET pin of the Arduino interface to the nRESET pin of the nRF9160 SiP.
- When the interface MCU is disconnected, shorting **SB21** will connect the RESET pin of the nRF52840 to the RESET button.
- Shorting **SB22** will connect the RESET pin of the nRF52840 to the RESET button.

3.2.1 USB detect

To detect when the USB for the interface MCU is connected, there is a circuit sensing the VBUS of USB connector **J6**.

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

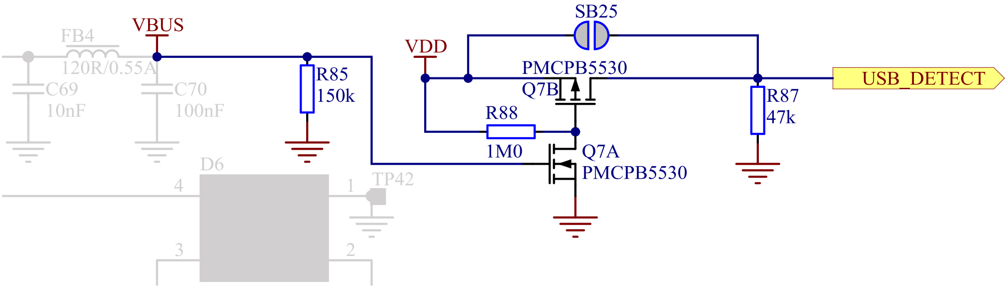


Figure 4: USB detect switch

4 Hardware description

The nRF9160 DK (PCA10090) features an onboard programming and debugging solution.

4.1 Hardware figures

The nRF9160 DK hardware drawings show both sides of the PCA10090.

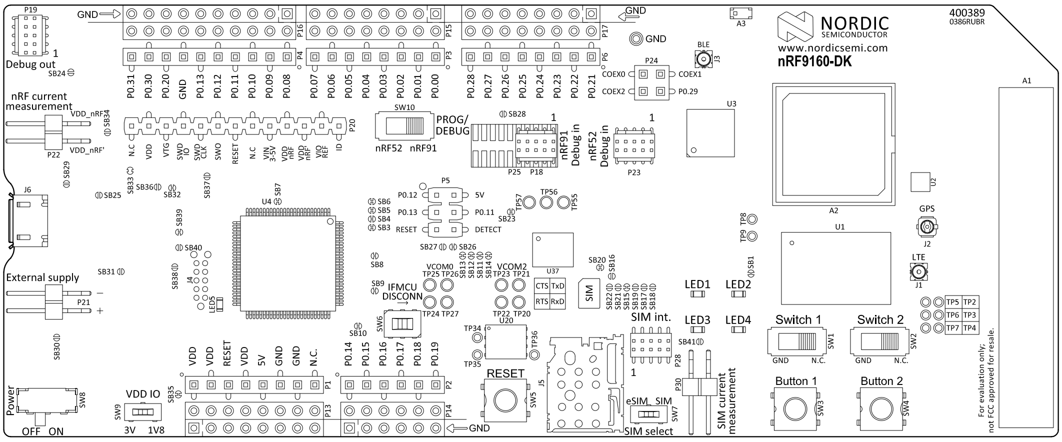


Figure 5: nRF9160 DK (PCA10090), front view

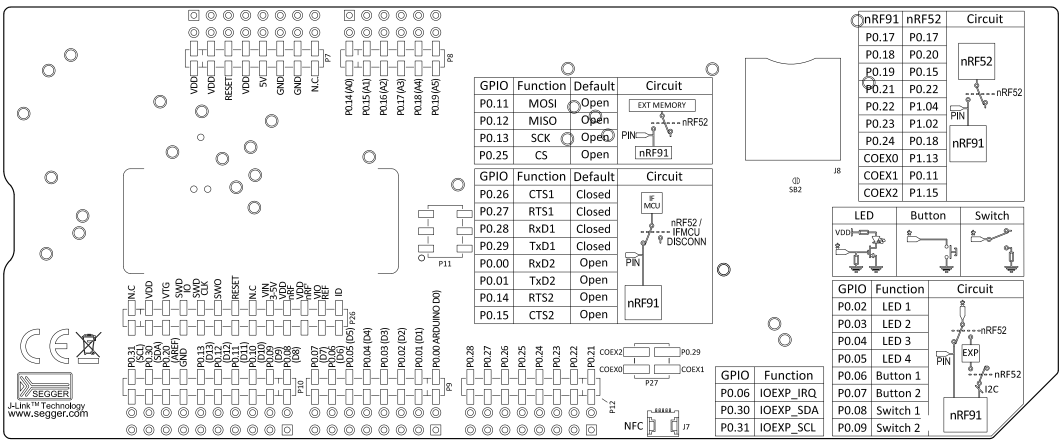


Figure 6: nRF9160 DK (PCA10090), back view

4.2 Block diagram

The block diagram shows the main functionality of the nRF9160 DK.

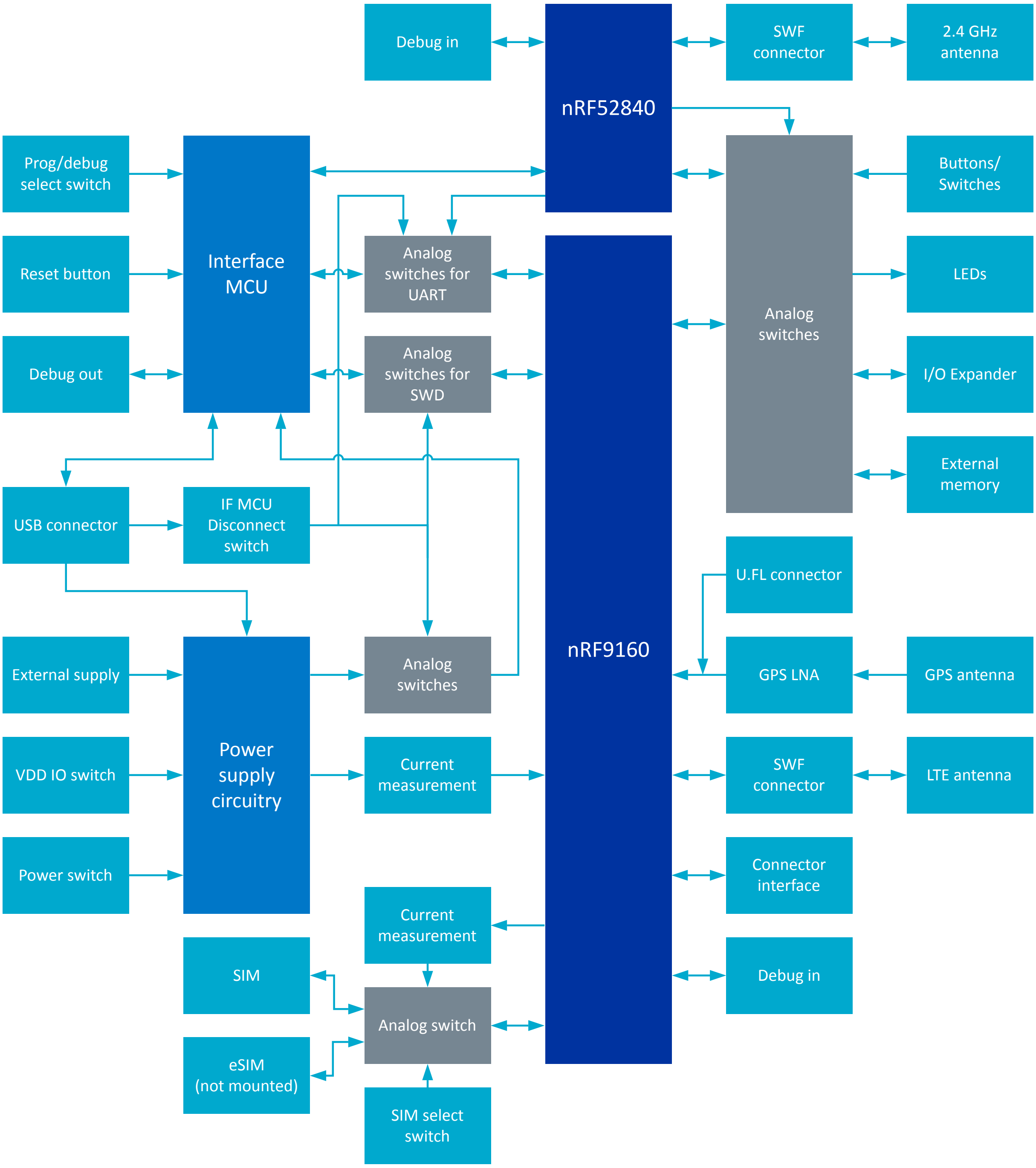


Figure 7: nRF9160 DK block diagram

4.3 Power supply

nRF9160 DK has a flexible and configurable power supply system to allow software development and testing using different power sources and to facilitate accurate power measurements.

The power source options are:

- USB connector **J6** (5 V)
- External supply on **P21** (3.0 V–5.5 V)
- VIN 3–5 V on **P20** (3.0 V–5.5 V)

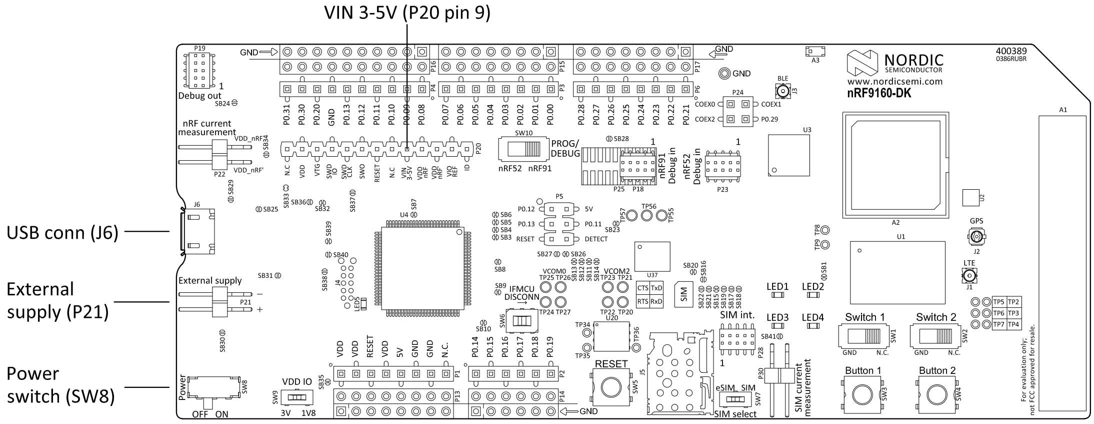


Figure 8: nRF9160 DK power supply options

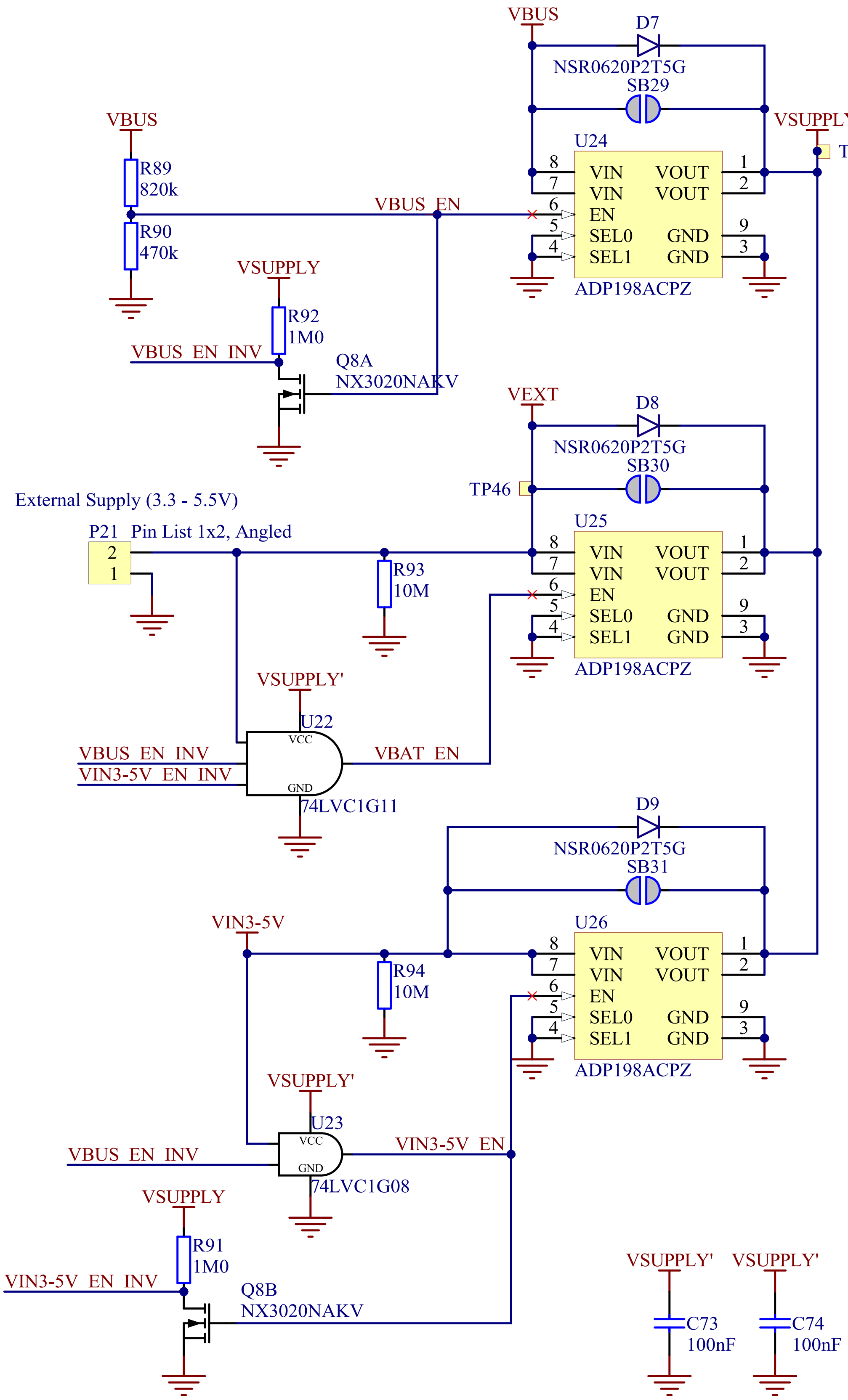


Figure 9: Power sources and switches

To ensure that only one of the power sources are used on the *DK* at a time, power switches are implemented on each of them as shown in Figure 9: Power sources and switches on page 15. These switches prioritize the supply sources in the following manner:

1. USB
2. P20
3. P21

This means that if power is connected to more than one the interfaces, the higher priority interface will be chosen to supply the *DK*.

The supply voltage is then routed through the ON/OFF switch (**SW8**) to the common rail VSUPPLY, which acts as the source for the supply voltage regulators for the circuitry on the *DK*.

The supply flows from VOUT to VIN, which is correct. The body diode of the internal transistor powers the VSUPPLY net, which supplies the gates controlling the enable signal of the switches.

The power switches will introduce a small voltage drop between the power source connected to the *DK* and the VSUPPLY. To avoid this, the power switches can be bypassed by shorting one of the solder bridges as shown in the following table:

Power source	Power switch bypass	Voltage level
USB connector (J6)	SB29	5 V
External supply (P21)	SB30	3.0 V–5.5 V
VIN 3–5 V (P20)	SB31	3.0 V–5.5 V

Table 2: Bypassing power switch

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

4.3.1 nRF9160 supply

The nRF9160 has a supply range of 3.0–5.5 V and is therefore powered by the VSUPPLY rail directly.

4.3.2 VDD supply rail

VDD is the main supply for the rest of the circuitry on the *DK*. It is regulated down from VSUPPLY by a buck regulator (**U27**).

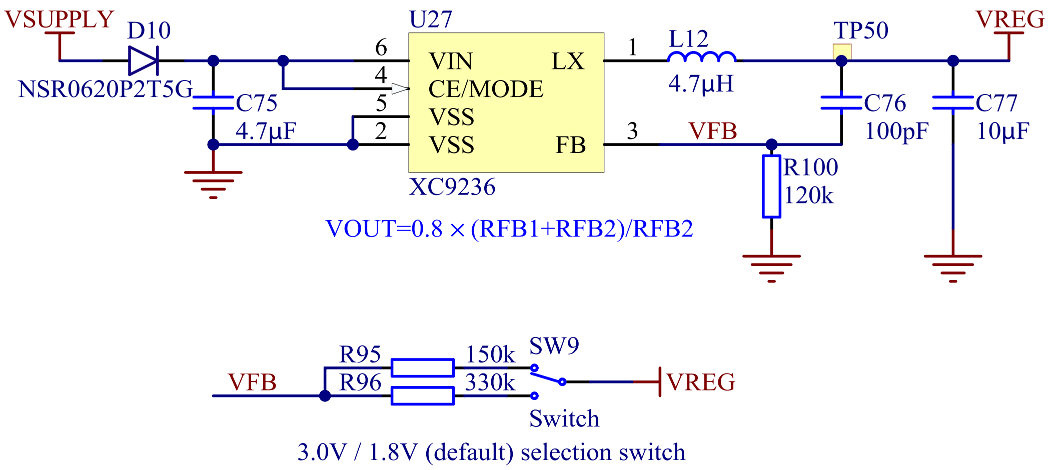


Figure 10: VDD buck regulator and selection switch

You can set the VDD voltage to 1.8 V (default) or 3 V with **SW9**. Running 3 V GPIO with heavy load may degrade the LTE RF performance. A third option for customizing VDD voltage level is applying a custom voltage level to VIO_REF (**P20**). For more information, see GPIO - General purpose input/output in the nRF9160 Objective Product Specification.

VDD powers most of the other circuits and will set the GPIO signal amplitude between nRF9160 and other circuits on the *DK* including connectors and PIN headers.

4.3.3 Other power domains

The interface MCU needs a 3.0 V for its USB interface supply, a low-dropout voltage regulator (**U28**) is used for this. This regulator also supplies the LEDs on the *DK*, giving these a fixed supply.

4.4 Antenna interfaces

nRF9160 DK has three antenna interfaces mounted representing LTE, GPS, and the 2.4 GHz radio.

The LTE and 2.4 GHz RF signals are propagated through two coaxial connectors with switches that will disconnect the corresponding antenna from the radio if adapter cables are connected. This makes it possible to perform conducted measurements or attach external antennas to the radio.

The *GPS* signal is RX only. There is a *Low-Noise Amplifier (LNA)* with integrated filters that amplify and filter the signal before it is fed to the *GPS* RF port on the nRF9160. An external active *GPS* antenna can be connected to **J2**. When using an external antenna, the *LNA* should be disabled.

The relation between the connectors, radios, and antennas are the following:

- **J1** – Connector with a switch for the LTE antenna (**A1**)
- **J2** – Connector for an external GPS antenna
- **J3** – Connector with a switch for the 2.4 GHz antenna (**A3**)

For more details of the *GPS* antenna interface, see GPS on page 17.

4.5 GPS

The nRF9160 has a dedicated *GPS* port to support global navigation.

The *GPS* signal is received either from the onboard or an external active *GPS* antenna. The onboard antenna (**A2**) is connected to the *LNA* (**U2**).

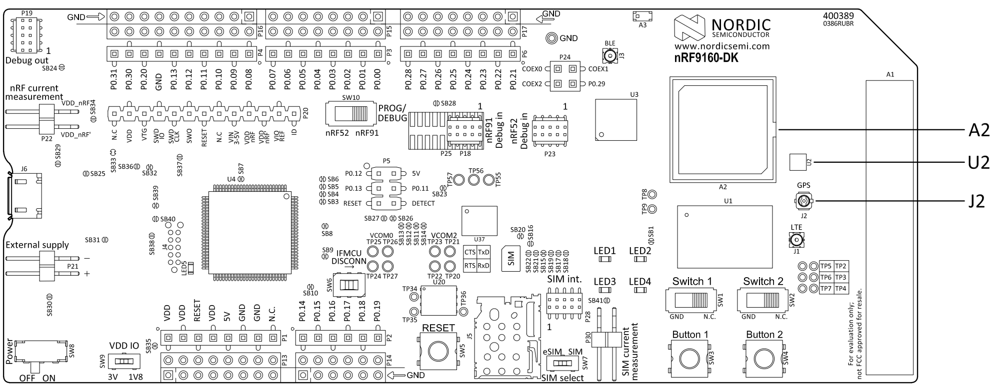


Figure 11: Onboard GPS antenna (A2), LNA (U2) and connector J2

The *GPS* from the onboard antenna is amplified and filtered in the *LNA* (**U2**) that has integrated pre-filter and post-filter before it is fed to the nRF9160. This makes the *GPS* receiver more sensitive to *GPS* signals and less sensitive to interference from other sources on the *DK* or nearby. The *LNA* is enabled by a GPS-enable signal from the nRF9160, using the COEX0 pin.

An external active *GPS* antenna can be connected to connector **J2** (U.FL compatible), and 3.3 V DC is fed through **J2** to supply the *LNA* of the external antenna. When using an external *GPS* antenna, the onboard *GPS LNA* (**U2**) must be disabled to avoid interference. This is done by ensuring that the GPS_EN signal is low and it is set by the `AT%XCOEX0` AT command.

Note:

- *GPS* signals do not usually penetrate ceilings or other structures that well. Therefore, for best *GPS* performance, the *DK* should be placed on a flat surface in an open space outside, far from sources of interference and other structures that may block the signals from space.
- This functionality is only available if the modem firmware used in the nRF9160 supports *GPS*.
- The Molex patch antenna achieves the highest gain when placed horizontally on a surface (x-y) facing the z-axis since it can receive all propagated *GPS* signals. A lower gain will be experienced if the patch antenna is mounted on a surface is at an angle with the horizontal surface.

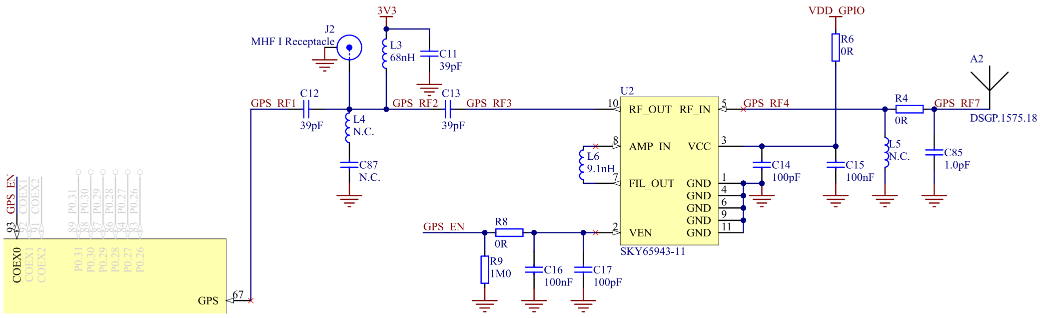


Figure 12: GPS connected to the nRF9160

4.6 GPIO interfaces

Access to the nRF9160 GPIOs is available from connectors **P2**, **P3**, **P4**, **P6**, and **P24**. The nRF9160 DK supports the Arduino UNO interface.

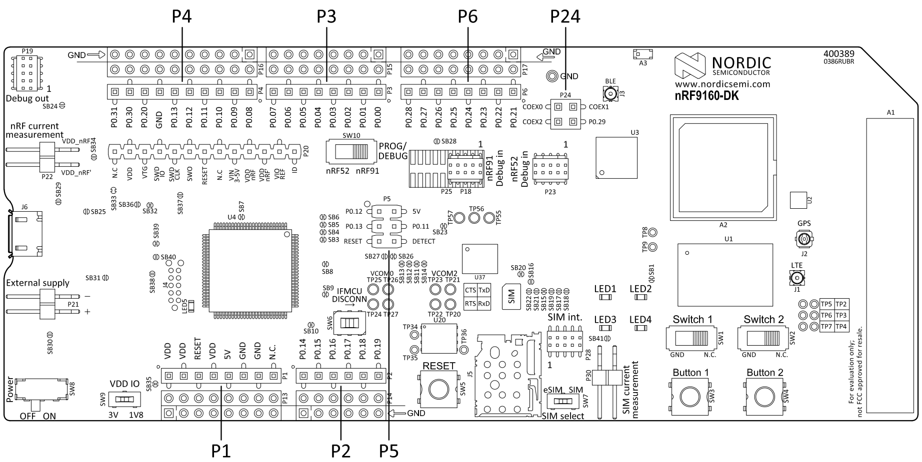


Figure 13: Access to nRF9160 GPIOs

GPIO signals are also available on connectors **P8**, **P9**, **P10**, **P12**, and **P27**, which are on the bottom side of the *DK*. By mounting pin lists on the connector footprints, the nRF9160 DK can be used as a shield for Arduino motherboards.

For easy access to GPIO, power, and ground, the signals can also be found on the through-hole connectors **P14**, **P15**, **P16**, and **P17**.

Note: GPIO **P0.29** is not available on any through-hole connector.

GPIO nRF9160 DK	Function
P0.00, P0.01, P0.14, and P0.15	Used as a second UART connection to the interface MCU. For more information, see Virtual COM port on page 8.
P0.02, P0.03, P0.04, P0.05, P0.06, P0.07, P0.08, and P0.09	Connected by default to buttons, slide switches, and LEDs. For more information, see Buttons, slide switches, and LEDs on page 24.
P0.17, P0.18, P0.19, P0.21, P0.22, P0.23, P0.24, COEX0, COEX1, and COEX2	Used to connect the nRF9160 to the nRF9160 DK board control on page 20.
P0.26, P0.27, P0.28, and P0.29	Used as the primary UART connection to the interface MCU. For more information, see Virtual COM port on page 8.
P0.11, P0.12, P0.23, and P0.25	Used for external memory interface. For more information, see External memory on page 27.

Table 3: Default pin settings

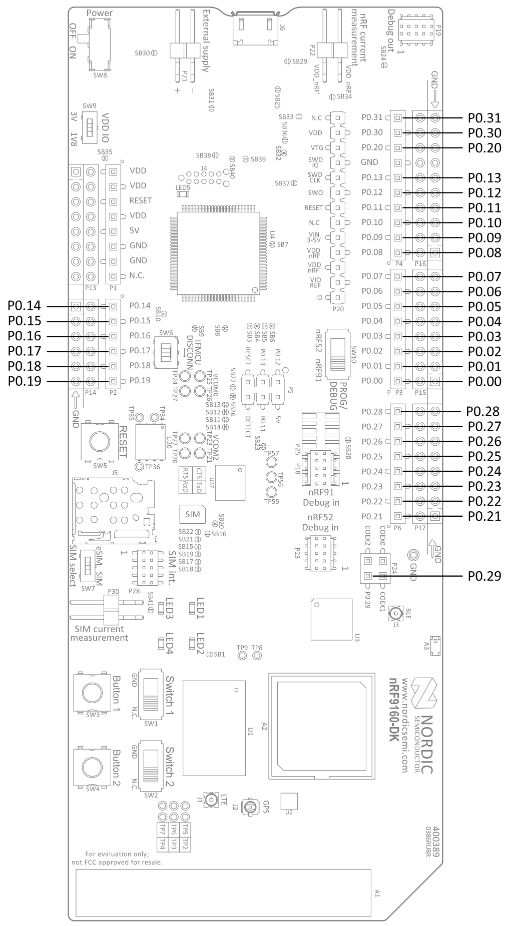


Figure 14: nRF9160 DK pins

4.7 nRF52840

An nRF52840 Bluetooth/IEEE 802.15.4 *System on Chip (SoC)* is included on the nRF9160 DK.

This device has two functions:

- nRF9160 DK control
- Bluetooth/IEEE 802.15.4 network processor

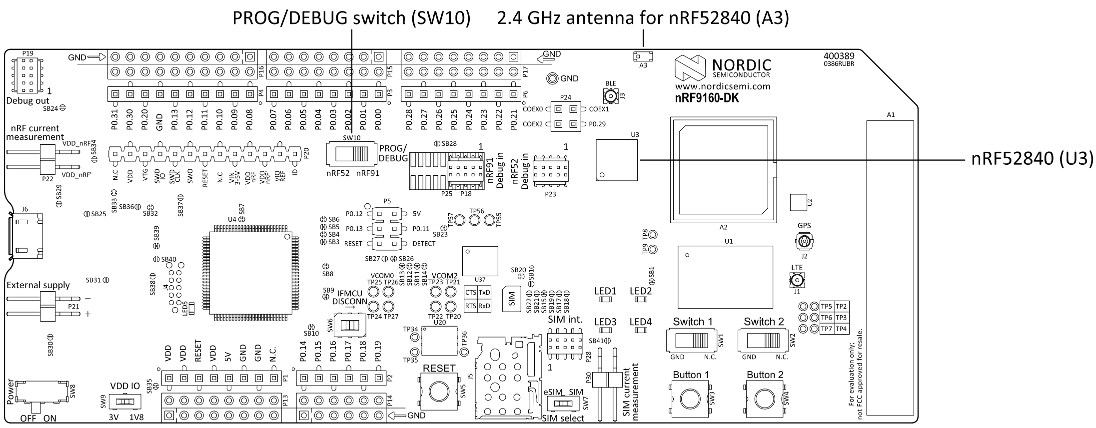


Figure 15: nRF52840 SoC on the nRF9160 DK

4.7.1 nRF9160 DK board control

The nRF52840 controls analog switches on the nRF9160 DK, enabling routing of some of the nRF9160 GPIO pins to onboard functionality, for example LEDs, or the regular GPIO interfaces.

For details on which GPIOs on the nRF9160 can be routed by these analog switches, see the following table.

Name	nRF52 control GPIO	Hardware default	nRF9160 GPIO	External[1]	Default	Optional
nRF91_UART1 (nRF91_APP1)	P1.14	0	P0.26	P6_pin6 (D19)	VCOM0_RTS	–
			P0.27	P6_pin7 (D20)	VCOM0_CTS	–
			P0.28	P6_pin8 (D21)	VCOM0_TXD	–
			P0.29	P24_pin4 (D22)	VCOM0_RXD	–
nRF91_UART2 (nRF91_APP2)	P1.12	1	P0.01	P3_pin2 (D1)	VCOM2_RXD	–
			P0.00	P3_pin1 (D0)	VCOM2_TXD	–
			P0.15/ AIN2	P2_pin2 (A1)	VCOM2_RTS	–
			P0.14/ AIN1	P2_pin1 (A0)	VCOM2_CTS	–
nRF91_LED1	P1.05	0	P0.02	P3_pin3 (D2)	LED 1	–
nRF91_LED2	P1.07	0	P0.03	P3_pin4 (D3)	LED 2	–
nRF91_LED3	P1.01	0	P0.04	P3_pin5 (D4)	LED 3	–
nRF91_LED4	P1.03	0	P0.05	P3_pin6 (D5)	LED 4	–

Name	nRF52 control GPIO	Hardware default	nRF9160 GPIO	External[1]	Default	Optional
nRF91_SWITCH1	P1.09	0	P0.08	P3_pin7 (D6)	Switch 1	–
nRF91_SWITCH2	P0.08	0	P0.09	P3_pin8 (D7)	Switch 2	–
nRF91_BUTTON1	P0.06	0	P0.06[2]	P4_pin1 (D8)	Button 1	–
nRF91_BUTTON2	P0.26	0	P0.07	P4_pin2 (D9)	Button 2	–
nRF_IF0-2_CTRL (nRF91_GPIO)	P0.13	0	P0.17/ AIN4	P2_pin4 (A3)	–	nRF52_P0.17
			P0.18/ AIN5	P2_pin5 (A4)	–	nRF52_P0.20
			P0.19/ AIN6	P2_pin6 (A5)	–	nRF52_P0.15
nRF_IF3-5_CTRL (nRF91_TRACE)	P0.24	0	P0.21	P6_pin1 (D14) / P25_pin12 (TRACECLK)	–	nRF52_P0.22
			P0.22	P6_pin2 (D15) / P25_pin14 (TRACEDATA0)[3]	–	nRF52_P1.04
			P0.23	P6_pin3 (D16) / P25_pin16 (TRACEDATA1)[3]	–	nRF52_P1.02
nRF_IF6-8_CTRL (nRF91_COEX)	P1.10	0	GPS_EN	P24_pin1 (COEX0)	–	nRF52_P1.13
			COEX1	P24_pin2 (COEX1)	–	nRF52_P1.11
			COEX2	P24_pin3 (COEX2)	–	nRF52_P1.15
nRF_IF9_CTRL	P0.16	0	P0.24	P6_pin4 (D17) / P25_pin28 (TRACEDATA2)[3]	–	nRF52_P0.18_Reset
IO_EXP_EN	P0.14	0	P0.06[2]	P3_pin7 (D6)	–	IO_EXP nINT
			P0.30	P4_pin9 (SDA)	–	IO_EXP SDA
			P0.31	P4_pin10 (SCL)	–	IO_EXP SCL
EXT_MEM_CTRL	P0.19	0	P0.11 (MOSI)	P4_pin4 (D11)	–	EXT_FLASH SI
			P0.12 (MISO)	P4_pin5 (D12)	–	EXT_FLASH SO
			P0.13/ AIN0 (SCK)	P4_pin6 (D13)	–	EXT_FLASH CSK

Name	nRF52 control GPIO	Hardware default	nRF9160 GPIO	External[1]	Default	Optional
			P0.25 (nCS)	P6_pin5 (D18) / P25_pin20 (TRACEDATA3)[3,4]	–	EXT_FLASH nCS

[1]For the location of connectors, see Figure 5: nRF9160 DK (PCA10090), front view on page 12.

[2]When the I/O expander is enabled, the interrupt line from the I/O expander shares the same GPIO as Button 1.

[3]Pins are shared with TRACE, and when TRACE functionality is to be used, make sure that the nRF52 control GPIO is in **Hardware default** state.

[4]Solder bridge and test point are available and so functionality can be maintained by manually connecting the signal to a GPIO of choice even when TRACE is in use.

Table 4: Board control routing

Name
: The name to be used when referring to this signal path and the configuration needed to control it.

nRF52 control GPIO
: The GPIOs on nRF52840 that are used to control the analog switches.

Hardware default
: The default logic level of the control signal if pin is not driven.

nRF9160 GPIO
: The GPIOs on nRF9160 that can be configured.

External
: The connector and pin with permanent routing of the nRF9160 GPIO, not affected by the board controller.

Default
: The default connection for the nRF9160 GPIO if the nRF52 control GPIO pins are not driven (Hardware default logic level).

Optional
: The optional connection for the nRF9160 GPIO if the nRF52 control GPIO pins are driven to the opposite logic level than the hardware default.

The setup described as Default in the table above is active if all the GPIOs on nRF52840 listed as nRF52 control GPIO are not driven or driven according to the hardware default setting. There are pull resistors on the switch control lines, and therefore, Default can also be received if the control lines from the nRF52840 are not driven. If the optional routing is desired, the nRF52840 control GPIO pins must be set as outputs and driven high.

The nRF52840 is preprogrammed with firmware that provides a control setting. This might not be the same as the hardware default. Configurable code examples for the nRF52840 allowing to change the board routing can be found in the nRF Connect SDK. After the changes are done, the new program must be compiled and programmed to the nRF52840. See the nRF Connect SDK documentation on nRF9160 board controller for information on changing the configuration.

> **Note:** To program and debug the nRF52840, nRF52 needs to be selected on the **PROG/DEBUG SW10** switch (**SW5** for v0.9.0 and earlier).

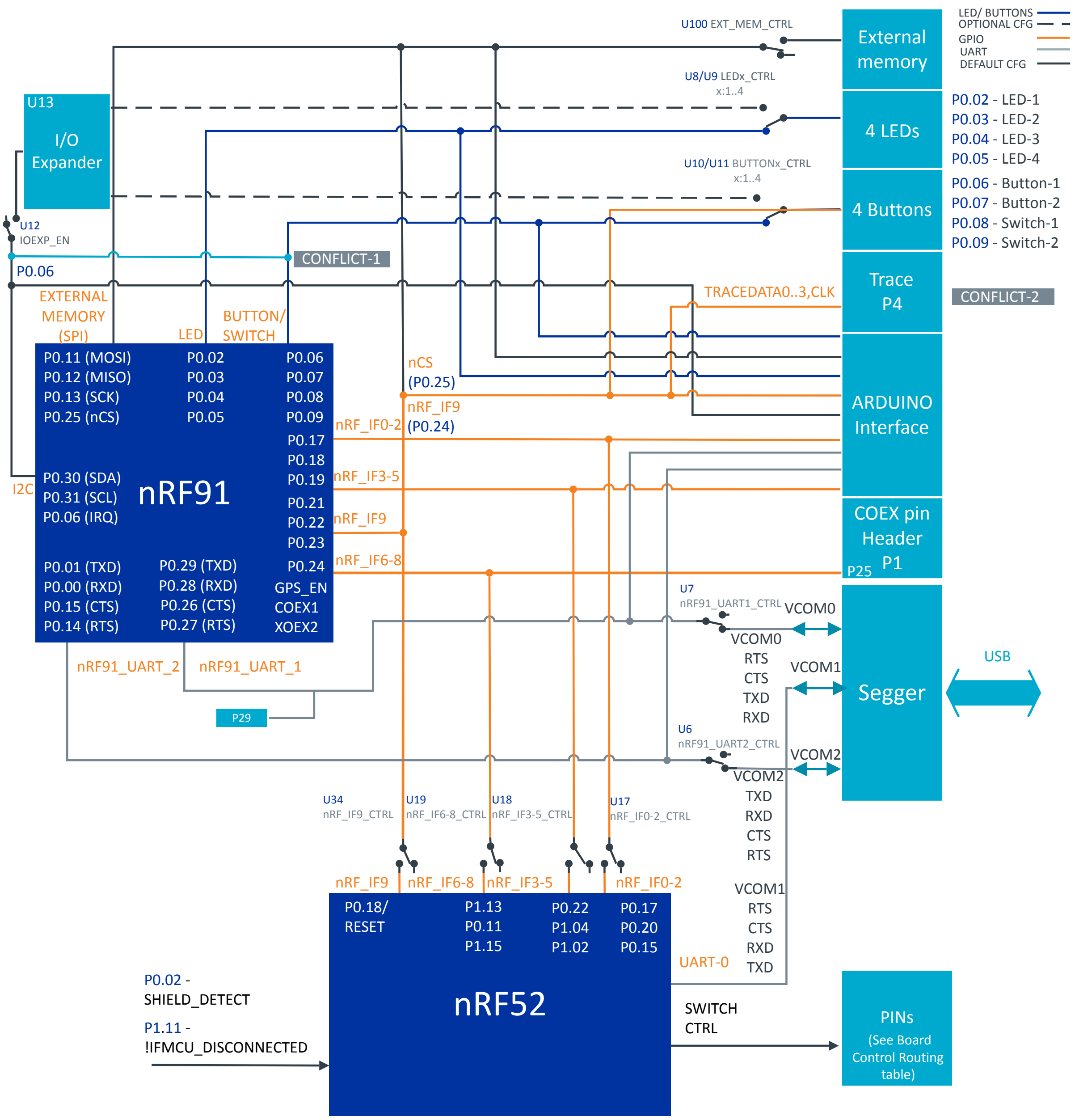


Figure 16: nRF9160 DK board control

Since this firmware in the nRF52840 decides the nRF9160 DK behavior, it is vital that it is always present in the nRF52840. If it is accidentally erased or firmware affecting the use of the key nRF52840 GPIOs is programmed in, nRF9160 DK functionality is not guaranteed. In such cases, nRF52840 needs to be preprogrammed with the default board control firmware from the nRF Connect SDK.

4.7.2 Bluetooth/IEEE 802.15.4 network processor

A second function of the onboard nRF52840 is to act as a Bluetooth or IEEE 802.15.4 network processor. This makes it possible to develop products that utilize both LTE and other RF protocols supported by Nordic on the nRF9160 DK.

Since most of the GPIOs of the nRF52840 are used for board control (see the schematic below), it is not intended to be a full programming and development target on the nRF9160 DK.

However, if combined with board control firmware, it can run code that requires limited access to GPIO, such as wireless controllers or network processor implementations of Bluetooth and IEEE 802.15.4 protocols from Nordic Semiconductor.

This enables the nRF9160 DK to implement a cellular to Bluetooth (and/or IEEE 802.15.4) bridge or gateway. To enable this, nRF52840 has a 2.4 GHz antenna fitted along with nine GPIOs that can be connected to nRF9160 through the board control routing. For details, see Table 4: Board control routing on page 20.

Firmware examples for such bridge or gateway implementations can be found in nRF9160: LTE Sensor Gateway.

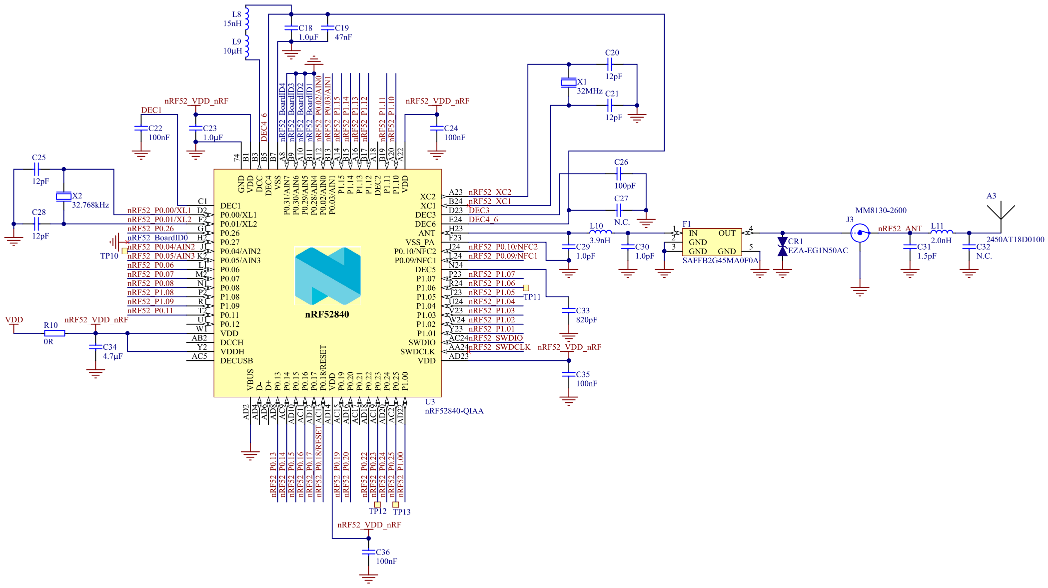


Figure 17: nRF52840 board controller

4.8 Buttons, slide switches, and LEDs

The nRF9160 DK has four LEDs, two buttons, and two switches for simple user interaction. By default, they are connected to nRF9160 GPIOs as shown in the following table.

To change default nRF9160 GPIO connections, see nRF9160 DK board control on page 20

Part	GPIO
LED 1	P0.02
LED 2	P0.03
LED 3	P0.04
LED 4	P0.05
Button 1	P0.06
Button 2	P0.07
Switch 1	P0.08
Switch 2	P0.09

Table 5: Button and LED connection

The nRF52840 can be used to control the connection of any of the LEDs, slide switches, or buttons. Optionally, any LED, slide switch, or button can be routed to an I/O expander. See I/O expander on page 25 for more information.

The buttons and switches are active low, meaning that the input will be connected to ground when the buttons are pushed or switches slid to the GND position. The buttons and switches have no external pull-up resistor, and therefore the `P0.06`, `P0.07`, `P0.08`, and `P0.09` pins must be configured as an input with an internal pull-up resistor.

The LEDs are active high, meaning that writing a logical one ('1') to the output pin will illuminate the LED. The nRF9160 GPIOs control power transistors and LEDs are fed from a separate 3.0 V domain. Therefore, LED current will not be drawn from nRF9160 GPIOs or the nRF9160 supply.

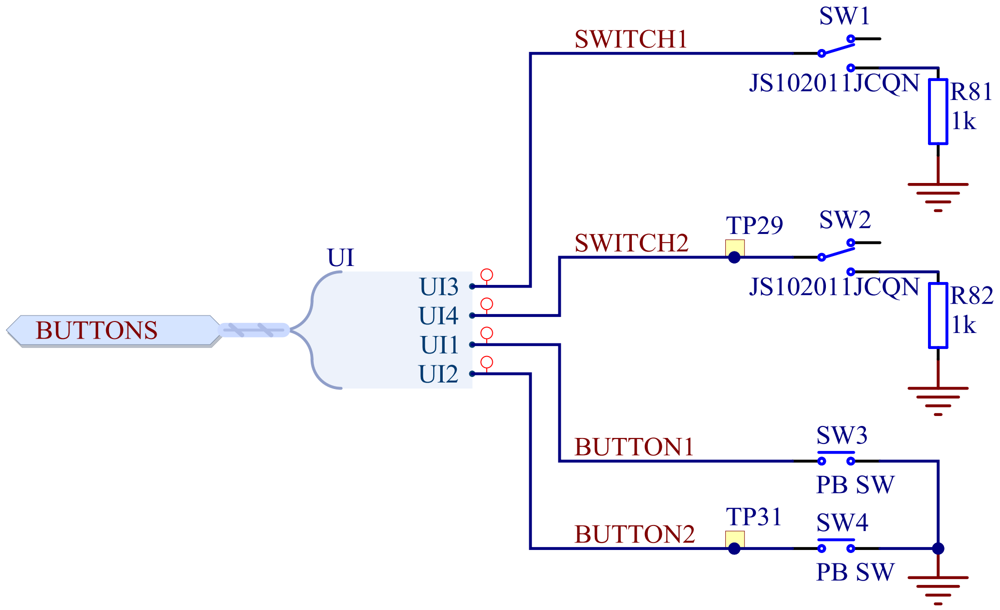


Figure 18: Buttons and switches

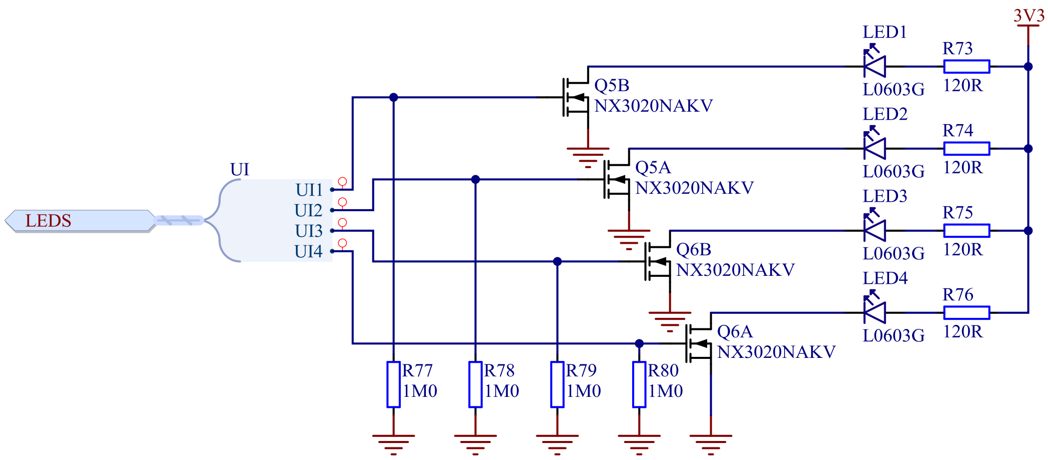


Figure 19: LEDs

4.9 I/O expander

The nRF9160 DK has an I/O expander that can optionally be used to interface the LEDs, slide switches, and buttons.

The connection of each component is controlled by the nRF52840 using analog switches (see Switches for buttons and LEDs on page 32). The I/O expander can be enabled by the nRF52840. See Switches for I/O expander on page 35. When the I/O expander is enabled, the I/O expander is interfaced by the nRF9160 GPIOs shown in the following table.

Signal	GPIO	Description
SDA	P0.30	I2C data line
SCL	P0.31	I2C clock line
IOEXP_IRQ	P0.06	Interrupt line from the I/O expander

Table 6: nRF9160 I/O expander interface

The following table shows the I/O expander connections:

Component	I/O expander pin
LED 1	I/O_EXP_IO4
LED 2	I/O_EXP_IO5
LED 3	I/O_EXP_IO6
LED 4	I/O_EXP_IO7
Switch 1	I/O_EXP_IO2
Switch 2	I/O_EXP_IO3
Button 1	I/O_EXP_IO0
Button 2	I/O_EXP_IO1

Table 7: nRF9160 I/O expander connections

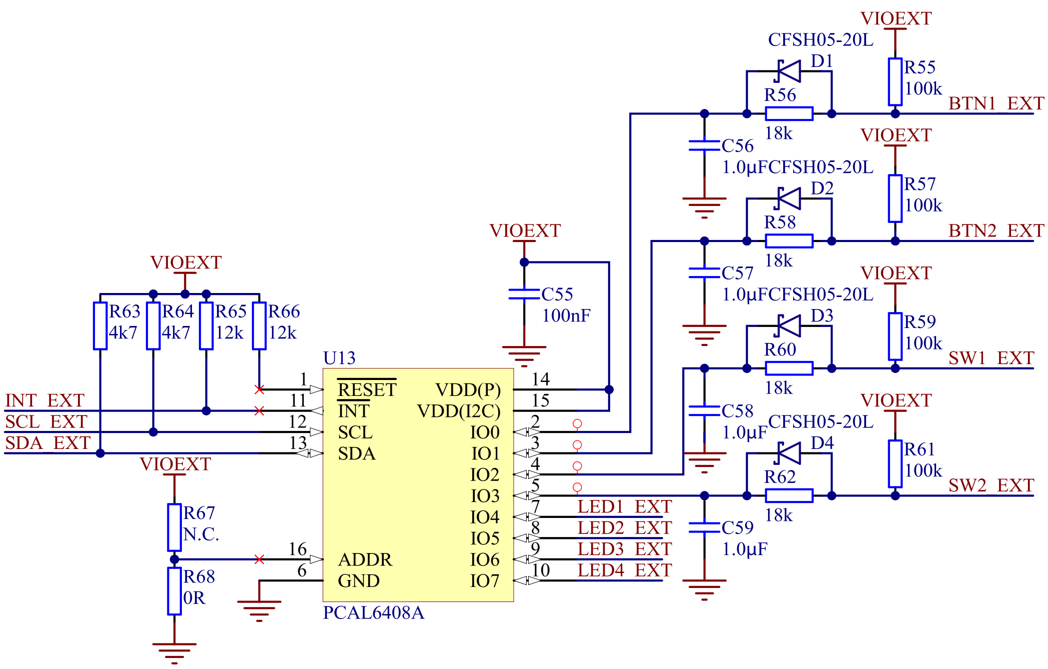


Figure 20: I/O expander

4.10 External memory

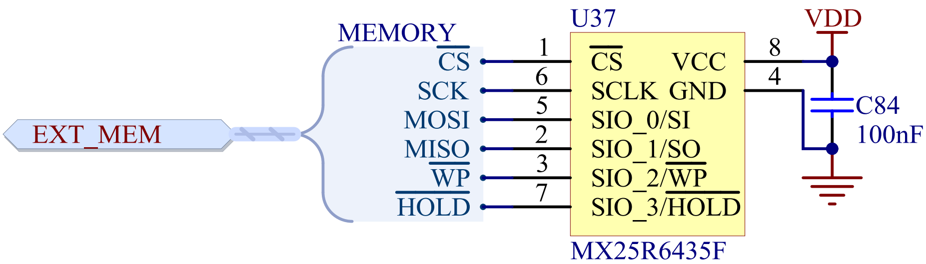
The nRF9160 DK has a 64 Mb external flash memory. The memory is a multi-I/O memory supporting both SPIs.

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

GPIO	Flash memory pin	Solder bridge for default use	Testpoint
P0.25	CS	SB23	TP57
P0.13	SCLK	N/A	N/A
P0.11	SIO_0/SI	N/A	N/A
P0.12	SIO_1/SO	N/A	N/A
N/A	SIO_2/WP	N/A	TP55
N/A	SIO_3/HOLD	N/A	TP56

Table 8: Flash memory GPIO usage, solder bridges and testpoints

The flash memory CS pin is by default connected to `P0.25`, this pin is also used for trace. To be able to use the external memory together with trace, Solder bridge **SB23** can be cut and a wire can be strapped from test point **TP57** to a GPIO of choice. The WP and HOLD pins of the external memory are available only on test points **TP55** and **TP56**, and to use these, wires can be strapped form the test points to the GPIOs of choice.



4.11 Debug input and trace options

The primary debug interface on the nRF9160 DK is the Segger OB debugger available through the USB port. However, if other than USB supply is used on the *DK*, this functionality will be disabled.

The Debug in connectors **P18** (nRF9160) and **P23** (nRF52840) make it possible to connect external debuggers for debugging when the interface USB cable is not connected or if the *DK* is in IF MCU DISCONNECT mode.

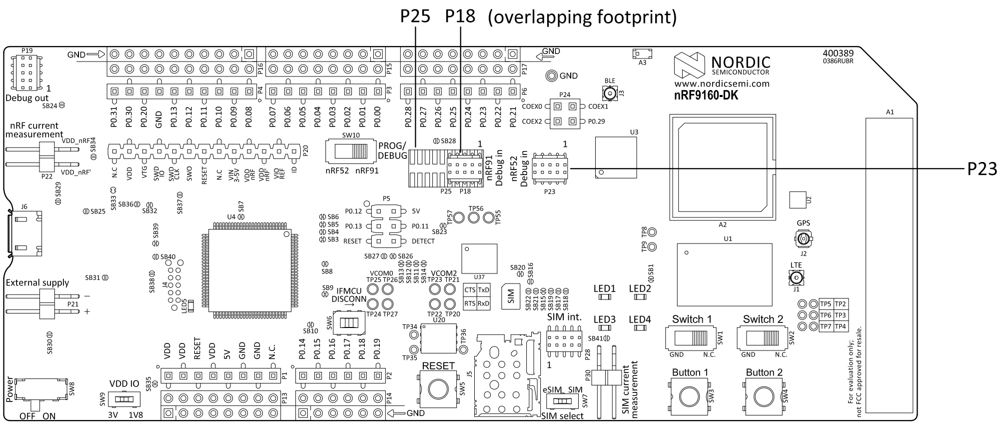


Figure 21: Debug input connector and trace footprint

To utilize the SW trace feature on nRF9160, 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 connector can be mounted. nRF9160 GPIOs used for the trace interface will not be available for application firmware use during trace.

Note: Connectors **P18** and **P25** overlap and share the same footprint.

GPIO	Trace
P0.21	TRACECLK
P0.22	TRACEDATA[0]
P0.23	TRACEDATA[1]
P0.24	TRACEDATA[2]
P0.25	TRACEDATA[3]

Table 9: nRF9160 trace interfaces

4.12 Debug output

The nRF9160 DK supports programming and debugging external boards with Nordic *SoC*s and *SiP*s.

To debug an external board with SEGGER J-Link OB IF, connect the Debug out connector **P19** to your target board with a 10-pin flat cable.

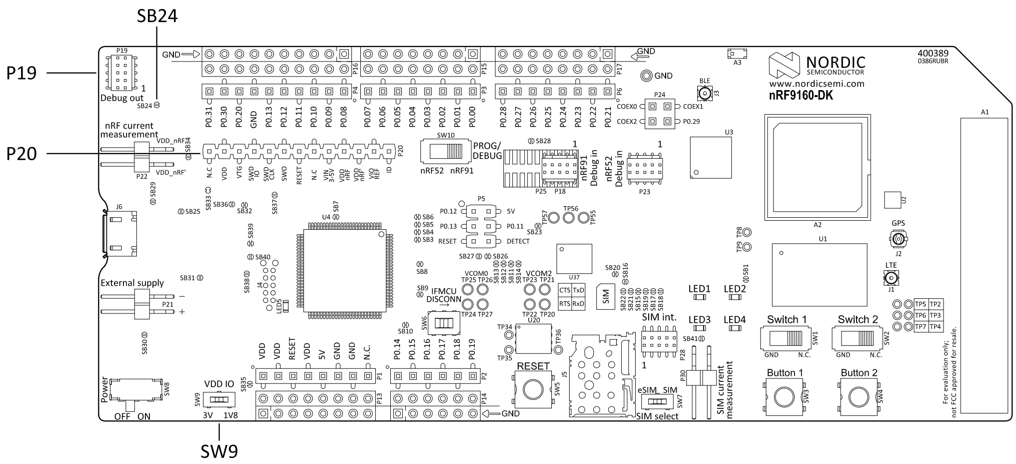


Figure 22: Debug output connector

When the external board is powered up, the interface MCU will detect the supply voltage of the board and program/debug the target chip on the external board instead of the onboard nRF9160 and/or nRF52840.

> **Note:** The voltage supported by external debugging/programming is the VDD voltage. This voltage can be selected to 1.8 V or 3 V using slide switch **SW9**. For optimal performance of the nRF9160 radio, it is recommended that only 1.8 V is used. Make sure the voltage level of the external board matches the VDD of the nRF9160 DK.

P20 can also be used as a debug out connector to program shield-mounted targets. For both **P19** and **P20**, the interface MCU will detect the supply voltage on the mounted shield and program/debug the target.

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

If it is inconvenient to have a separate power supply on the external board, the nRF9160 DK can supply power through the Debug out connector **P18**. To enable this, short solder bridge **SB24**.

4.12.1 Connectors for programming external boards

The voltage on the external board must match that of the *DK*.

Pin number	Signal	Description
1	SWD0_SELECT	Voltage supply from the external target, used as voltage detect input to the interface MCU
2	SWD0_SWDIO	Serial Wire Debug (SWD) data line
3	GND	Ground
4	SWD0_SWDCLK	SWD clock line
5	GND	Ground
6	SWD0_SWO	Serial Wire Output (SWO) line
7	N.C.	Not used
8	N.C.	Not used
9	N.C.	Not used
10	SWD0_RESET	Reset

Table 10: Pinout of connector P19 for programming external targets

Pin number	Signal	Description
1	N.C.	Not used
2	VDD	Used for VDD voltage
3	SWD1_SELECT	Voltage supply from the external target, used as voltage detect input to the interface MCU
4	SWD1_SWDIO	Serial Wire Debug (SWD) data line
5	SWD1_SWDCLK	SWD clock line
6	SWD1_SWO	Serial Wire Output (SWO) line
7	SWD1_RESET	Reset line
8	N.C.	Not used
9	VIN3-5V	Voltage supply
9	VDD_HV	Used for current measurement
10	VDD_HV’	Used for current measurement
12	VIO_REF	GPIO voltage reference input
13	ID	Board ID resistor

Table 11: Pinout of connector P20 for programming target on shields

4.13 Signal routing switches

A number of the GPIO signals of the nRF9160 are routed through analog switches for use for onboard functionality or having them available on the pin headers for external circuitry or Arduino type shields.

4.13.1 Interface MCU disconnect switches

The IF MCU DISCONNECT mode can be enabled using the switches shown in the schematics.

For more information, see Performance measurement mode on page 10.

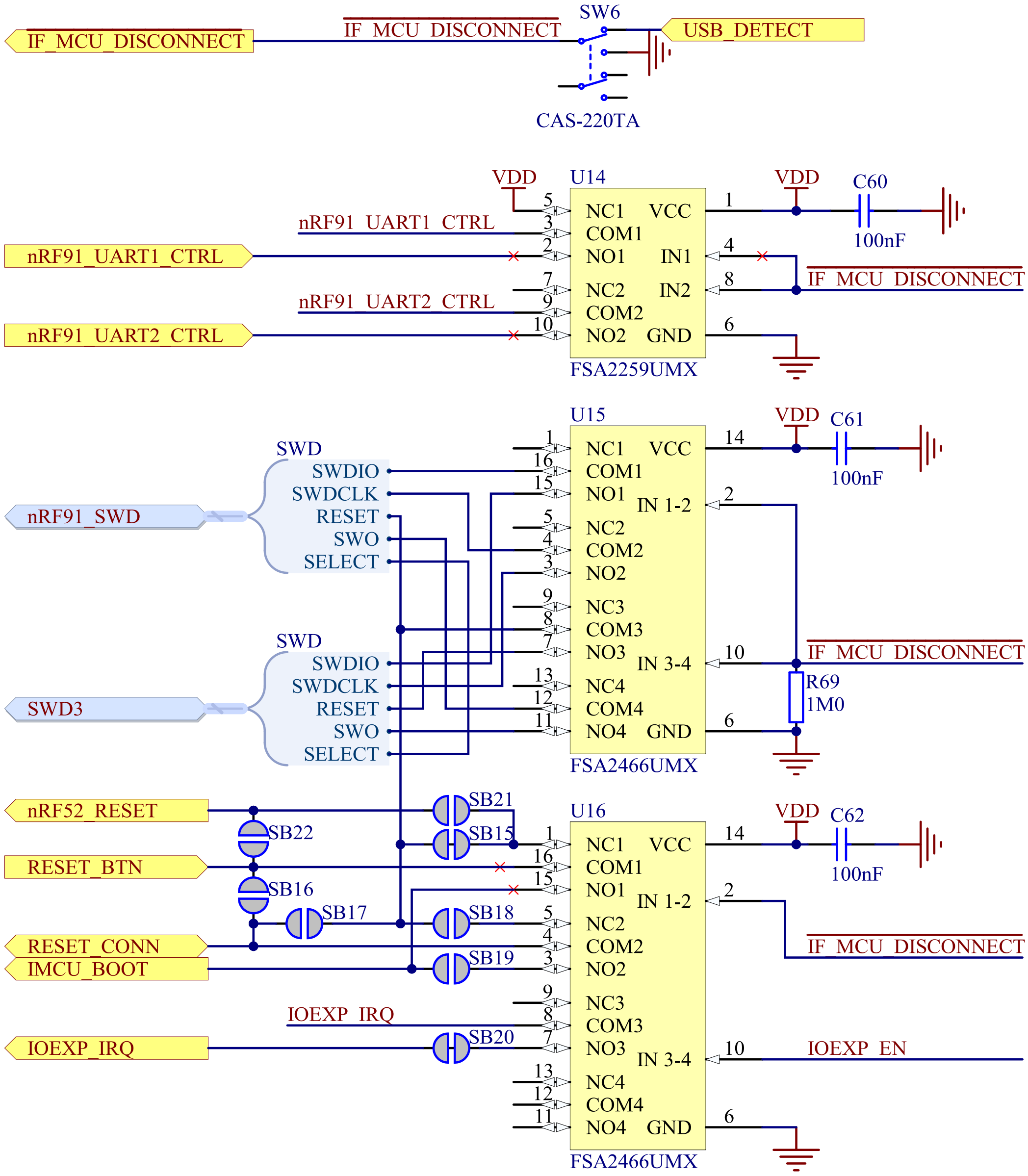


Figure 23: Signal interface switches

4.13.2 Switches for UART interface

Two UART interfaces are routed between the interface MCU and the nRF9160 SiP. These can be controlled individually.

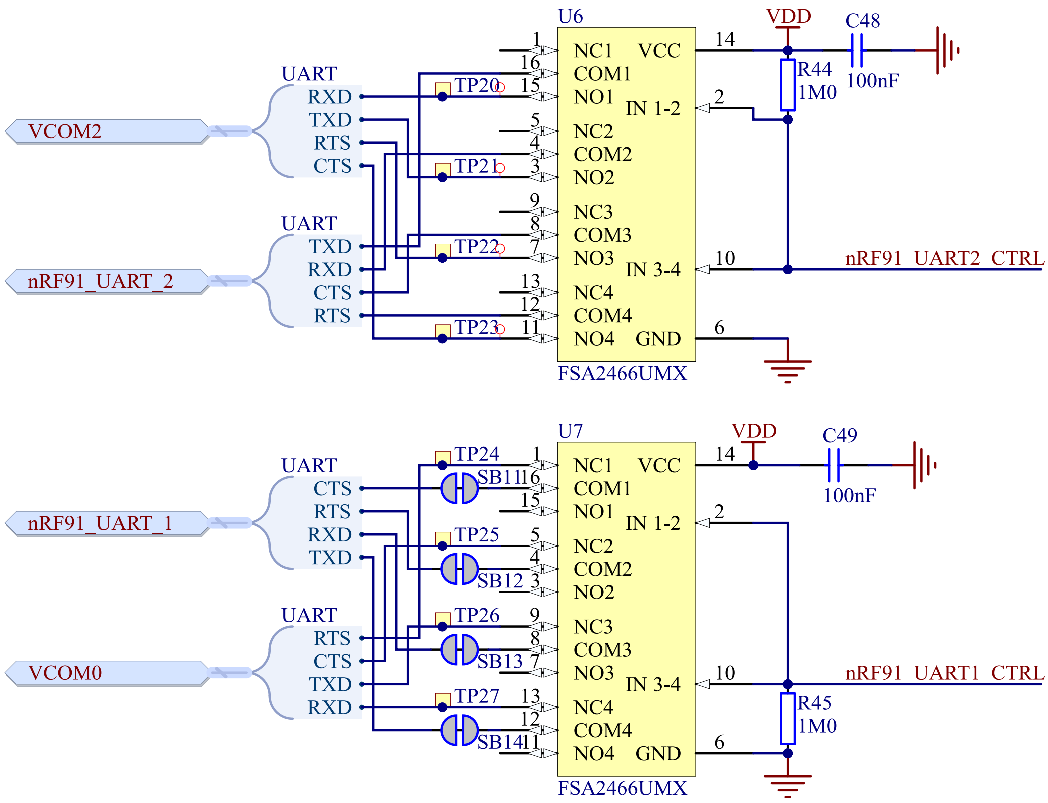


Figure 24: UART interface switches

4.13.3 Switches for buttons and LEDs

On the nRF9160 DK, there are a few analog switches that are used to connect and disconnect signals to control buttons, switches, and LEDs.

The analog switches select if the LEDs, buttons, and switches should be connected directly to the nRF9160 GPIO or to an I/O expander. See I/O expander on page 25 for more information. The switches are controlled by the nRF52840, which contains firmware from Nordic Semiconductor.

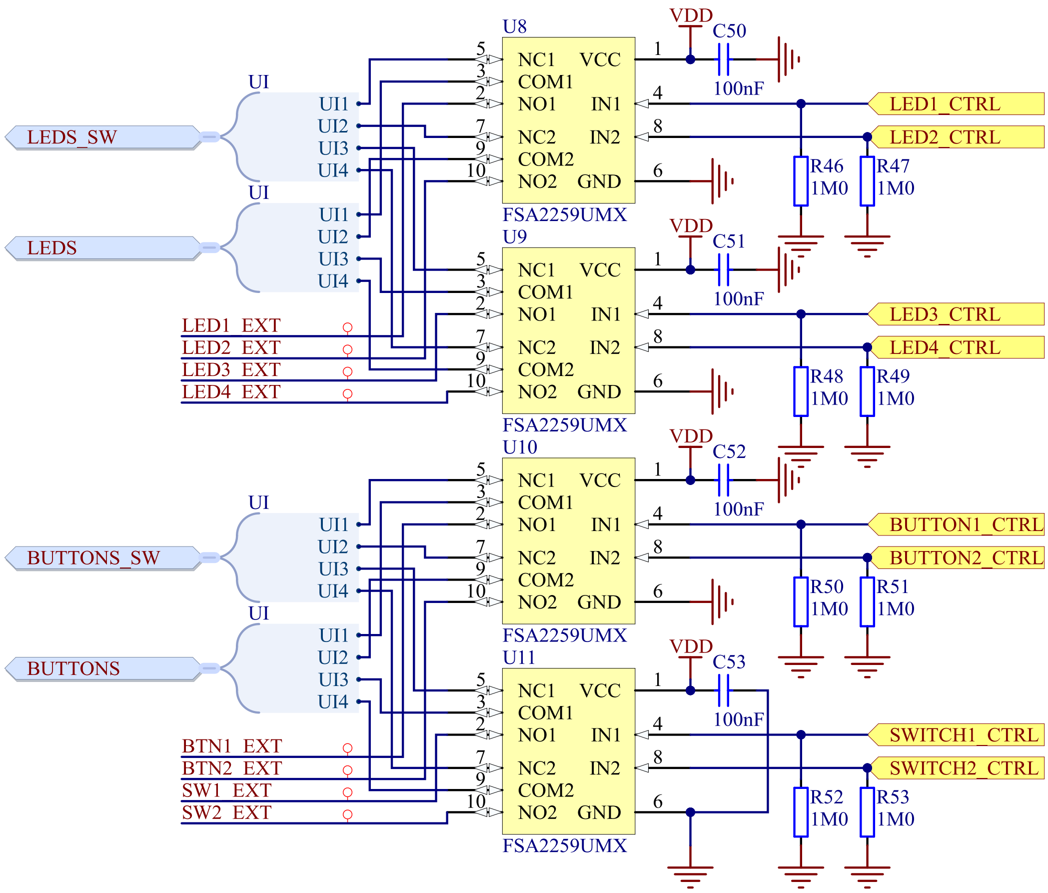


Figure 25: LED switch control

4.13.4 Switches for nRF52840 interface

A total of four analog switches controls the routing of signals between the nRF9160 *SiP* and the nRF52840 *SoC*. There are a total of ten signals that can be routed between the devices, seven GPIOs and three COEX pins.

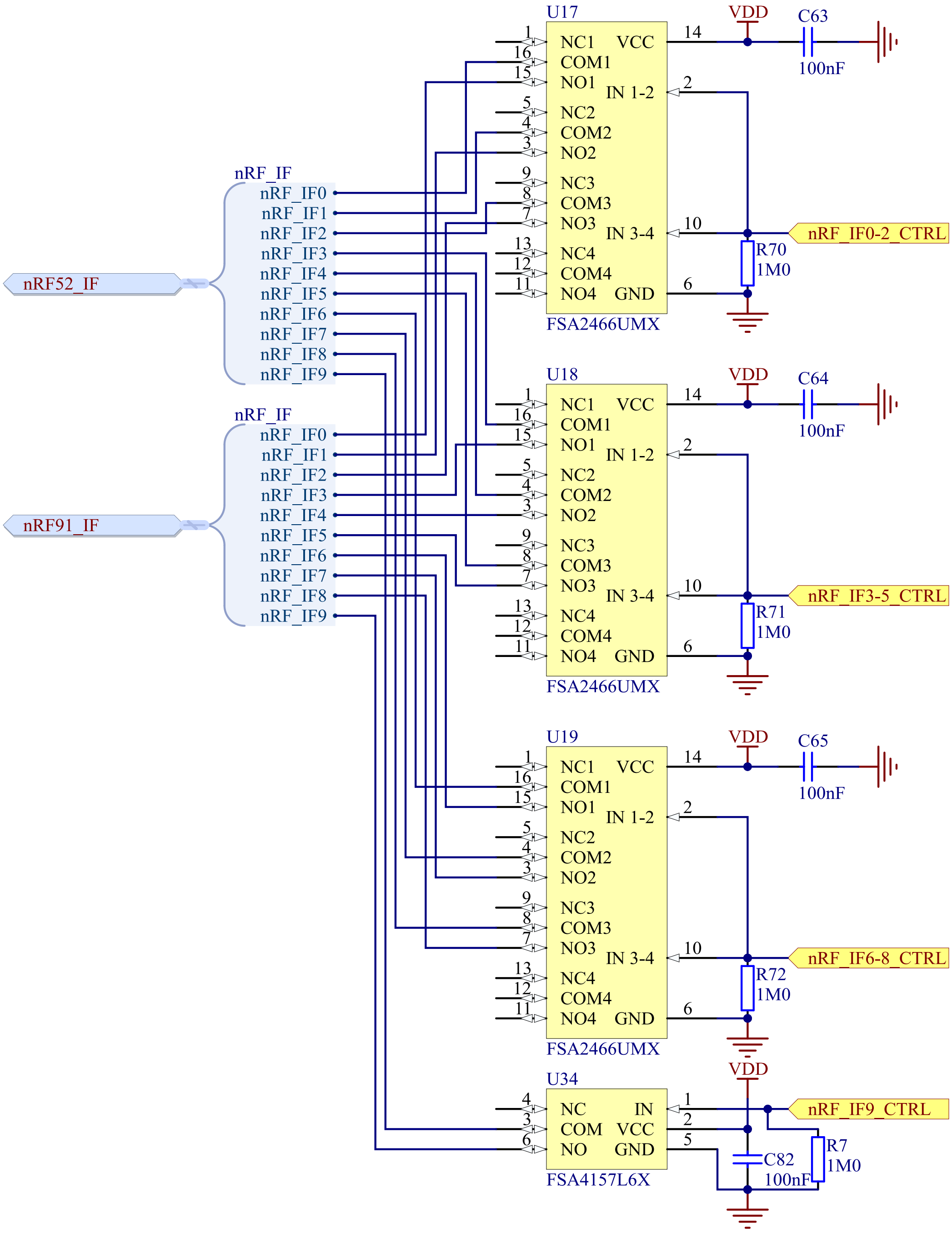


Figure 26: nRF52840 interface switches

4.13.5 Switches for external memory

Two analog switches are used to control the signals between the nRF9160 and the external memory.

The switches are controlled by **P0.19** of nRF52840.

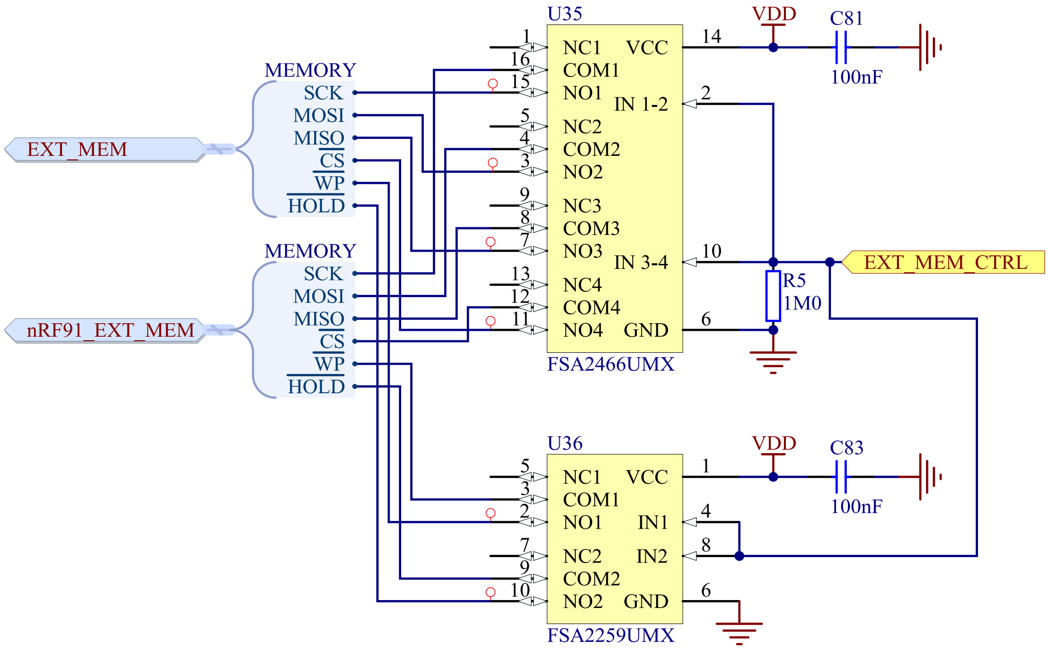


Figure 27: External memory switches

4.13.6 Switches for I/O expander

An analog switch is used to control the signals between the nRF9160 and the I/O expander.

The switch is controlled by **`P0.14`** of nRF52840.

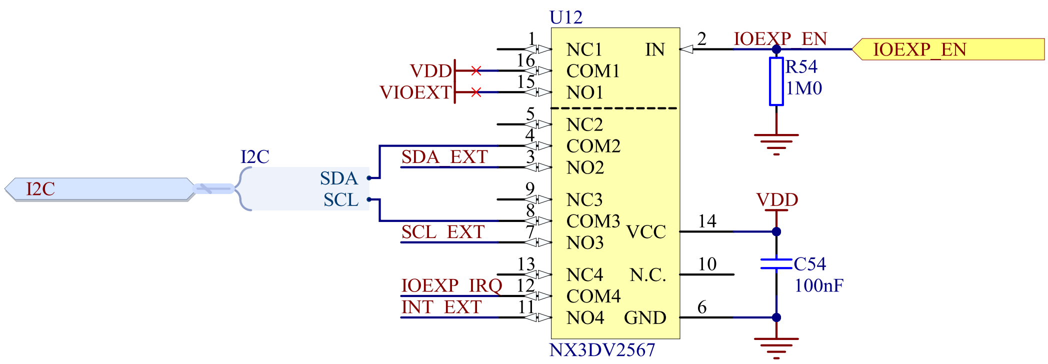


Figure 28: Switches for I/O expander

4.14 SIM and eSIM

The nRF9160 DK is designed to support both regular and embedded SIM (eSIM). For this purpose, it has a pluggable SIM card socket (**J5**) that takes a nano-sized SIM (4FF) and a non-populated footprint for an eSIM (MFF2).

Using the SIM socket is the default. If an eSIM is soldered on to the *DK*, switch **SW7** can be used to select the eSIM. Connector **P28** can be used to connect and monitor the traffic on the SIM interface.

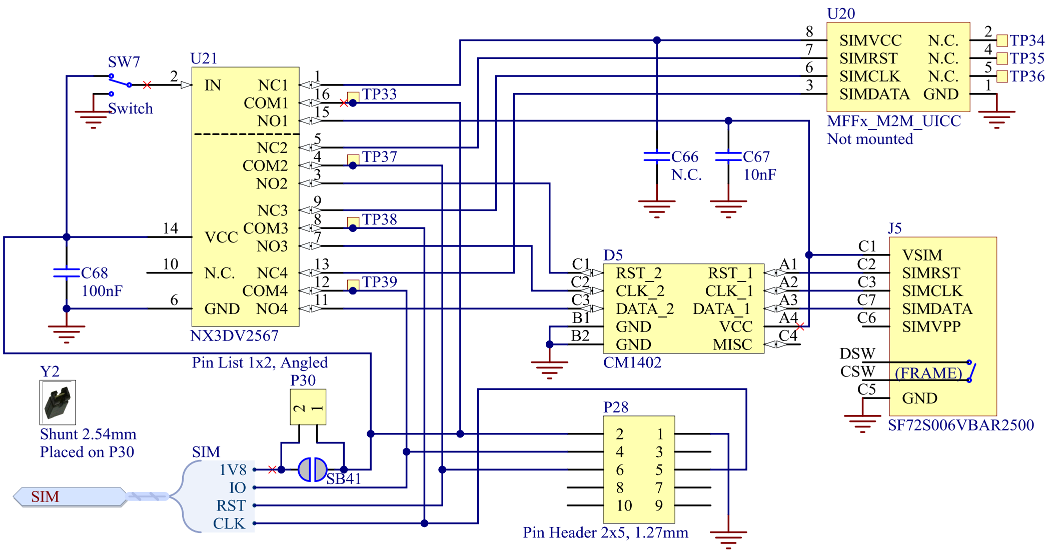


Figure 29: SIM card connector, eSIM, and selection switch

Note: The nano-SIM card is inserted with the electrical interface down in the SIM card holder (**J5**).

4.15 Additional interfaces

The nRF9160 DK supports a few dedicated interfaces.

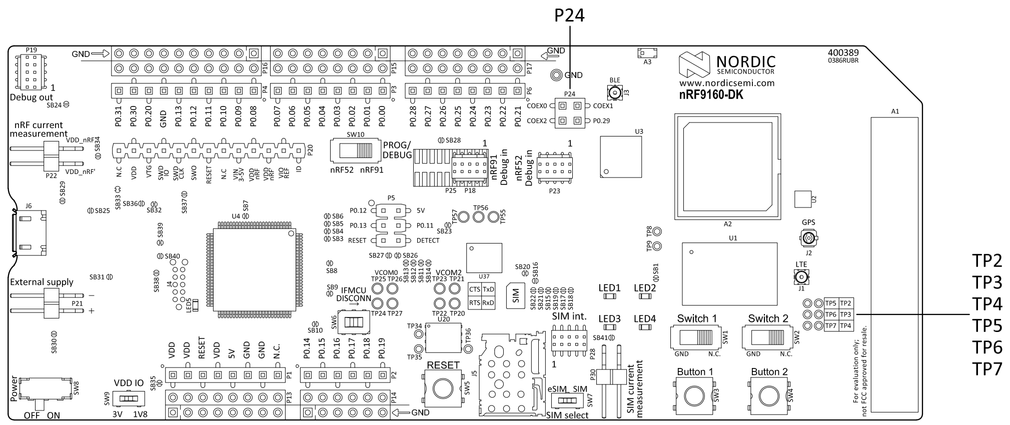


Figure 30: SiP external interfaces

Coexistence interface

On the nRF9160 DK, there are three COEX pins enabling coexistence handling between multiple wireless devices. These pins are routed to the nRF52840 for use when nRF52840 is used as a network processor. These signals are also available on connector **P24**. In addition, COEX0 is used to enable the *GPS LNA* of the nRF9160 DK.

RF control interface

nRF9160 has an interface allowing the modem to control external RF applications, such as antenna tuner devices. There are three MIPI RFFE interface pins, namely, VIO, SCLK, SDATA and three MAGPIO interface pins, namely, MAGPIO0, MAGPIO1, MAGPIO2. The MAGPIOs and MIPI RFFE pins are not in use in this *DK*. The user can access the signals through test points. See the following table:

Signal	Testpoint
SDATA	TP2
SCLK	TP3
VIO	TP4
MAGPIO0	TP5
MAGPIO1	TP6
MAGPIO2	TP7

Table 12: RF control interface testpoints

4.16 SiP enable

The *SiP* can be enabled or disabled by pulling pin **101** high and low.

By default, the enable signal is pulled high by resistor **R1**.

4.17 Solder bridge configuration

The nRF9160 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 a complete overview of the solder bridges on the nRF9160 DK.

Solder bridge	Default	Function
SB1	Closed	Cut to disconnect VDD_GPIO supply to the nRF9160 SiP
SB2	Open	Short to connect DBG_CMD to the SD card reader **J8** (not populated by default)
SB3	Closed	Cut to disconnect RTS signal from the nRF52840 SoC
SB4	Closed	Cut to disconnect CTS signal from the nRF52840 SoC
SB5	Closed	Cut to disconnect RXD signal from the nRF52840 SoC
SB6	Closed	Cut to disconnect TXD signal from the nRF52840 SoC
SB7	Closed	Cut to disconnect the 3.0 V USB supply from the interface MCU (**U4**)
SB8	Closed	Cut to disconnect SWDIO from the nRF52840 SoC
SB9	Closed	Cut to disconnect SWDCLK from the nRF52840 SoC
SB10	Closed	Cut to disconnect the RESET signal from the nRF52840 SoC
SB11	Closed	Cut to disconnect CTS from the nRF9160 SiP
SB12	Closed	Cut to disconnect RTS from the nRF9160 SiP
SB13	Closed	Cut to disconnect RXD from the nRF9160 SiP
SB14	Closed	Cut to disconnect TXD from the nRF9160 SiP
SB15	Closed	Cut to disconnect the RESET button from the nRF9160 reset pin when the interface MCU is disconnected

Solder bridge	Default	Function
SB16	Open	Short to connect the RESET button to the RESET pin on the Arduino interface
SB17	Open	Short to connect the RESET pin on the Arduino interface to the nRF9160 reset pin
SB18	Open	Short to connect the RESET pin on the Arduino interface to the nRF9160 reset pin when the interface MCU is disconnected
SB19	Open	Short to connect the RESET pin on the Arduino interface to the interface MCU Boot when the interface MCU is disconnected
SB20	Closed	Cut to disconnect IOEXP_IRQ from nRF9160 GPIO when the I/O expander is enabled
SB21	Open	Short to connect the RESET button to the nRF52840 reset pin when the interface MCU is disconnected
SB22	Open	Short to connect the RESET button to the RESET pin of the nRF52840
SB23	Closed	Cut to disconnect external memory chip select from P0.25
SB24	Open	Short to enable power supply of the external device when using the Debug out connector (**P19**)
SB25	Open	Short to bypass the USB detect switch (**Q7**)
SB26	Closed	Cut to permanently disable SHIELD DETECT feature
SB27	Open	Short to permanently enable the SHIELD DETECT feature
SB28	Open	Short to supply VDD to bottom UART debug connector
SB29	Open	Short to bypass the USB supply switch (**U24**)
SB30	Open	Short to bypass the external supply switch **U25**
SB31	Open	Short to bypass the VIN3–5 V switch **U26**
SB32	Closed	Cut to disconnect supply to LDO (**U28**)
SB33	Closed	Cut to disconnect the VSUPPLY and VDD_nRF'
SB34	Open	Short to connect VDD_nRF and VDD_nRF'
SB35	Closed	Cut to disconnect VDD and VIO
SB36	Open	Short to bypass the VDD_IMCU switch (**U30**)
SB37	Open	Short to bypass the VDD_IMCU_3V3 switch (**U31**)
SB38	Open	Short to bypass switch **U33**
SB39	Closed	Cut to disconnect supply from Arduino interface
SB40	Open	Short to bypass Arduino 5 V reverse protection diode **D11**
SB41	Open	Short to permanently connect SIM VDD, bypass Sim current measurement connector

Table 13: Solder bridge configuration

5 Measuring current

The current drawn by the nRF9160 can be monitored on the nRF9160 DK.

Current can be measured using various test instruments. Examples of test equipment are the following:

- Power analyzer
- Oscilloscope
- Ampere-meter

Connector **P22** can be used for measuring current consumption or monitoring voltage levels to the nRF9160.

The use of a USB connector is not recommended for powering the *DK* during current measurements due to potential noise from the USB power supply. Instead, the *DK* should be powered by connecting a power supply to connector **P21**.

For more information on measuring, see sections Using an oscilloscope for current profile measurement on page 40 and Using a current meter for current measurement on page 40.

5.1 Preparing the development kit for current measurements

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

1. Remove jumper from **P22**.

   Removing the jumper disconnects the nRF9160 from the power management circuitries on the *DK*.

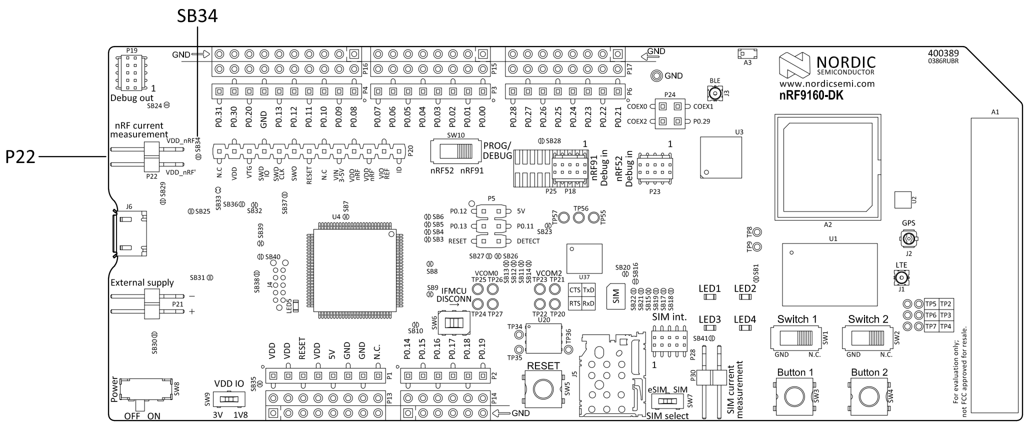


Figure 31: Solder bridge SB34 and P22 on the nRF9160 DK

2. To restore normal kit functionality after measurement, apply the jumper on **P22** or short **SB34**.

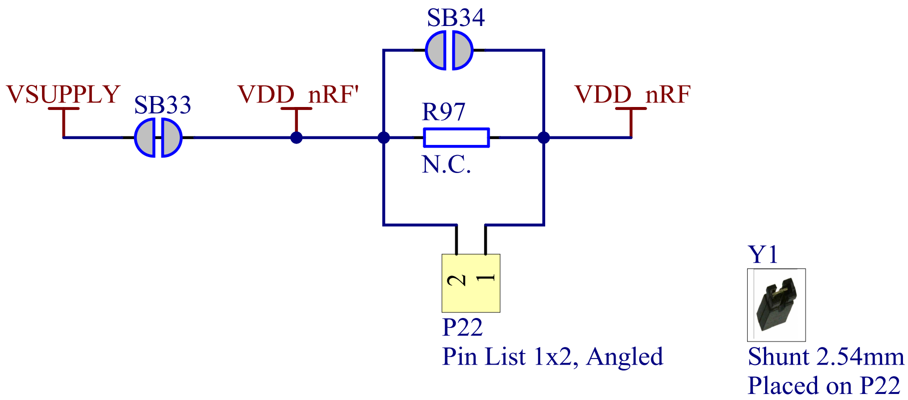


Figure 32: nRF power source

5.2 Using an oscilloscope for current profile measurement

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

Before you start, make sure you have prepared the *DK* as described in section Preparing the development kit for current measurements on page 39.

Follow the steps below to measure the current profile of the nRF9160:

1. Solder a 0.5 Ω resistor to R97.

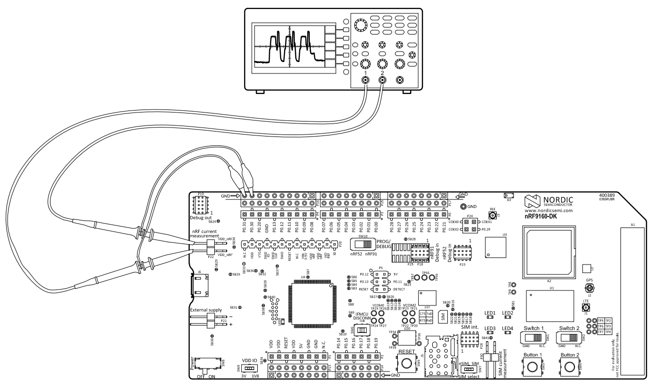


Figure 33: Current measurement with an oscilloscope

2. Connect probes to the pins on connector **P22**.
3. Measure the current power profile by measuring the voltage drop over the 0.5 Ω resistor.

5.3 Using a current meter for current measurement

The current drawn by the nRF9160 can be measured using a current meter.

Before you start, make sure you have prepared the *DK* as described in section Preparing the development kit for current measurements on page 39.

To connect the current meter in series with the nRF9160, connect the current meter to the pins on connector **P22**.

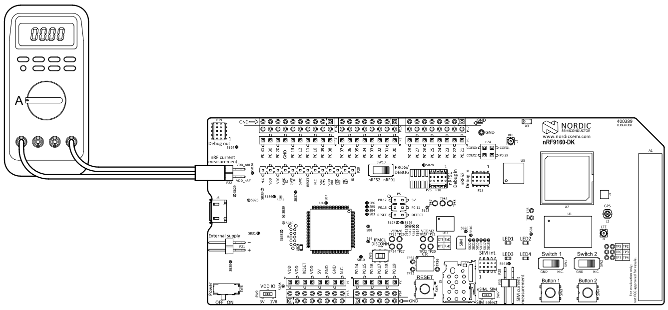


Figure 34: Current measurement with a current meter

Note: Use a high-speed, high-dynamic-range ampere meter for the best and most reliable measurements. Current range switching in the current meter may affect the power supply to the nRF9160. High speed and bandwidth are required to detect rapid changes in the current consumption in the nRF9160.

6 RF measurements

The nRF9160 DK is equipped with a small size coaxial connectors **J1** and **J3** for measuring the RF signal for LTE and 2.4 GHz.

The connectors are 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 onboard antenna. The insertion loss in the adapter cable is approximately 0.5–1 dB.

An adapter is available (Murata part no. MXHS83QE3000) with a standard SMA connection on the other end for connecting instruments (the adapter is not included in the kit). When connecting the adapter, the internal switch in the SWF connector will disconnect the onboard antenna and connect the RF signal from the nRF9160 or nRF52840 to the adapter.

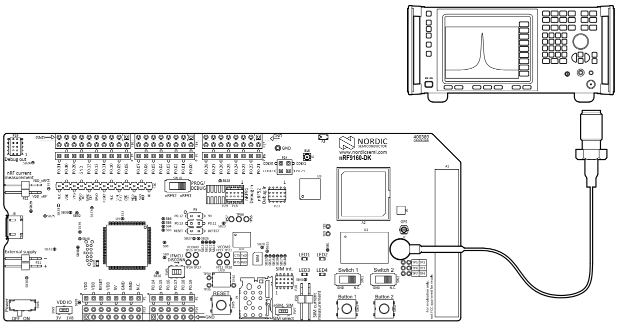


Figure 35: Connecting a spectrum analyzer

7 Radiated performance of nRF9160 DK

The LTE antenna on the *DK* is optimized for global operation, supporting all LTE frequency bands in the region of 698–960 MHz and 1710– 2200 Mhz.

All antennas on the nRF9160 DK have fixed matching networks, which means that no matching configuration is needed to switch between the frequency bands. The average performance of the LTE antenna mounted on the nRF9160 DK is shown in the table below. The LTE antenna also supports other frequency ranges but it is not optimized for operating on these frequency bands.

Band	Average return loss [dB]	Average efficiency [%]
3	-13.5	62.0
4	-7.1	47.1
13	-8.5	58.5
20	-10.5	73.8

Table 14: Antenna performance versus radio band

The 2.4 GHz antenna on the nRF9160 DK is optimized for the frequency range of 2.4–2.48 GHz. It has an average return loss of -15 dB, average efficiency being 80 %.

Glossary

Band-Pass Filter (BPF)

An electronic device or circuit that passes frequencies within a certain range and rejects frequencies outside that range.

Cat-M1

LTE-M User Equipment (UE) category with a single RX antenna, specified in 3GPP Release 13.

Cat-NB1

Narrowband Internet of Things (NB-IoT) User Equipment (UE) category with 200 kHz UE bandwidth and a single RX antenna, specified in 3GPP Release 13.

Clear to Send (CTS)

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

Development Kit (DK)

A development platform used for application development.

Fast Identity Online (FIDO)

A bundle of hardware and software modules serving as a tracing interface between traced device(s) and user application.

Global Positioning System (GPS)

A satellite-based radio navigation system that provides its users with accurate location and time information over the globe.

Hardware Flow Control (HWFC)

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

Inter-integrated Circuit (I^2C)

A multi-master, multi-slave, packet-switched, single-ended, serial computer bus.

Low-Dropout Regulator (LDO)

A linear voltage regulator that can operate even when the supply voltage is very close to the desired output voltage.

Low-Noise Amplifier (LNA)

In a radio receiving system, an electronic amplifier that amplifies a very low-power signal without significantly degrading its signal-to-noise ratio.

nRF Cloud

Nordic Semiconductor's platform for connecting IoT devices to the cloud, viewing and analyzing device message data, prototyping ideas that use Nordic Semiconductor chips, and more. It includes a public REST API that can be used for building IoT solutions. See nRF Cloud (nrfcloud.com).

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.

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.

SAW filter

A high-performing filter using Surface Acoustic Wave (SAW) technology. This technology employs piezoelectric transducers, which, when excited, produce waves that are used to filter out desired frequencies.

Surface Acoustic Wave (SAW)

An acoustic wave traveling along the surface of a material exhibiting elasticity, with an amplitude that typically decays exponentially with depth into the substrate.

Subscriber Identity Module (SIM)

A card used in *User Equipment (UE)* containing data for subscriber identification.

System in Package (SiP)

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

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.

Transmit Data (TXD)

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

User Equipment (UE)

Any device used by an end-user to communicate. The UE consists of the Mobile Equipment (ME) and the Universal Integrated Circuit Card (UICC).

Acronyms and abbreviations

These acronyms and abbreviations are used in this document.

BPF
: Band-Pass Filter

Cat-M1

Cat-NB1

CTS
: Clear to Send

DK
: Development Kit

FIDO
: Fast Identity Online

GPS
: Global Positioning System

HWFC
: Hardware Flow Control

I^{2}C
: Inter-integrated Circuit

LDO
: Low-Dropout (Regulator)

LNA
: Low-Noise Amplifier

PCB
: Printed Circuit Board

RTS
: Request to Send

RXD
: Receive Data

SAW
: Surface Acoustic Wave

SIM
: Subscriber Identity Module

SiP
: System in Package

SoC
: System on Chip

TXD
Transmit Data

UE
User Equipment

Recommended reading

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

Nordic documentation

- nRF9160 DK Getting Started Guide
- nRF9160 Product Specification
- nRF9160 Modem Firmware Release Notes (included in the latest nRF9160 modem firmware)
- nRF Connect SDK documentation
- nRF91 AT Commands Reference Guide
- nRF Connect Programmer

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