nRF52 DK Hardware

v3.x.x

User Guide

4397_500 / 2023-04-11

NORDIC®
SEMICONDUCTOR

Contents

Revision history . . . iii

Environmental and safety notices . . . v

1 Introduction . . . 6

2 Kit content . . . 7

3 Interface MCU . . . 8

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

4 Hardware description . . . 11

4.1 Hardware drawings . . . 11
4.2 Block diagram . . . 12
4.3 Power supply . . . 12
4.4 Connector interface . . . 15
4.5 Buttons and LEDs . . . 16
4.5.1 I/O expander for buttons and LEDs . . . 17
4.6 32.768 kHz crystal . . . 19
4.7 Debug input . . . 20
4.8 Debug out for programming external boards . . . 21
4.8.1 Programming an external board . . . 22
4.8.2 Programming a board with custom connections . . . 23
4.9 NFC antenna interface . . . 25
4.10 Solder bridge and test point overview . . . 27

5 Current measurement . . . 30

5.1 Set up the DK . . . 30
5.2 Measure current profile with an oscilloscope . . . 31
5.3 Measure average current with an ampere meter . . . 32

6 RF measurements . . . 34

Glossary . . . 35

Recommended reading . . . 38

Legal notices . . . 39

Revision history

Date	Description

2023-04-11	• Updated the following chapters for nRF52 DK v3.x.x:
	  ◦ Interface MCU on page 8 - Removed Mbed OB interface firmware information
	  ◦ Reset button on page 8 - Updated button function
	  ◦ Virtual serial port on page 8 - Added information on two virtual serial ports
	  ◦ Dynamic hardware flow control on page 9 - Added information on UART without HWFC
	  ◦ Power supply on page 12 - Updated schematic and IF MCU current consumption information
	  ◦ Programming an external board on page 22 - Updated instructions and P19 connector pinout
	  ◦ Programming a board with custom connections on page 23 - Updated instructions and P20 connector pinout
	  ◦ Table 8: Solder bridge configuration for nRF52 DK (v3.0.0 and higher) on page 27 - Updated table contents
	• Updated drawings and schematics
	• Editorial changes

2023-01-05	• Updated Debug out for programming external boards on page 21
	• added:
	  ◦ Programming an external board on page 22
	  ◦ Programming a board with custom connections on page 23
	• Editorial changes

2022-10-19	• Updated content for nRF52 DK v2.2.0
	• Editorial changes
	• Removed:
	  ◦ Setting up the development kit
	  ◦ Software tools
	  ◦ Start developing

April 2020	Editorial changes
December 2018	Updated: • Introduction on page 6 • Software tools • Current measurement on page 30 • Set up the DK on page 30
February 2017	• Created PDF for nRF52 DK v1.1.x (valid for all DK versions) • Added Mass Storage Device on page 9 • Updated:

Date	Description
	• Interface MCU firmware • Solder bridge and test point overview on page 27

Previous versions

PDF files for relevant previous versions are available here:

- nRF52 DK User Guide v2.x.x
- nRF52 DK User Guide v1.3.1

Environmental and safety notices

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

> **Note:** The nRF52 DK must be powered by a PS1 class (IEC 62368-1) power supply with maximum power less than 15 W.

Skilled persons

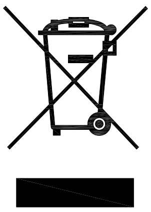
The nRF52 DK is intended for use only by skilled persons.

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 nRF52 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 nRF52 *Development Kit (DK)* is a hardware development platform used to design and develop application firmware on the nRF52832, nRF52810, or nRF52805 *System on Chip (SoC)*s.

Key features

- nRF52832 flash-based *Bluetooth*® Low Energy and ANT™/ANT+™ SoC solution
- Buttons and LEDs for user interaction
- I/O interface for Arduino form factor plug-in modules
- SEGGER J-Link OB Debugger with debug out functionality
- UART interface through a virtual serial port
- Drag-and-drop *Mass Storage Device (MSD)* programming
- Support for *NFC-A Listen Mode*
- Pins for measuring power consumption

To get started, see the Software development Getting Started Guides. For more detailed documentation regarding nRF52810 development, see Developing for nRF52810 in the SDK documentation.

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

2 Kit content

The nRF52 DK includes hardware, preprogrammed firmware, documentation, hardware schematics, and layout files.

The nRF52 DK (PCA10040) comes with an *Near Field Communication (NFC)* antenna.

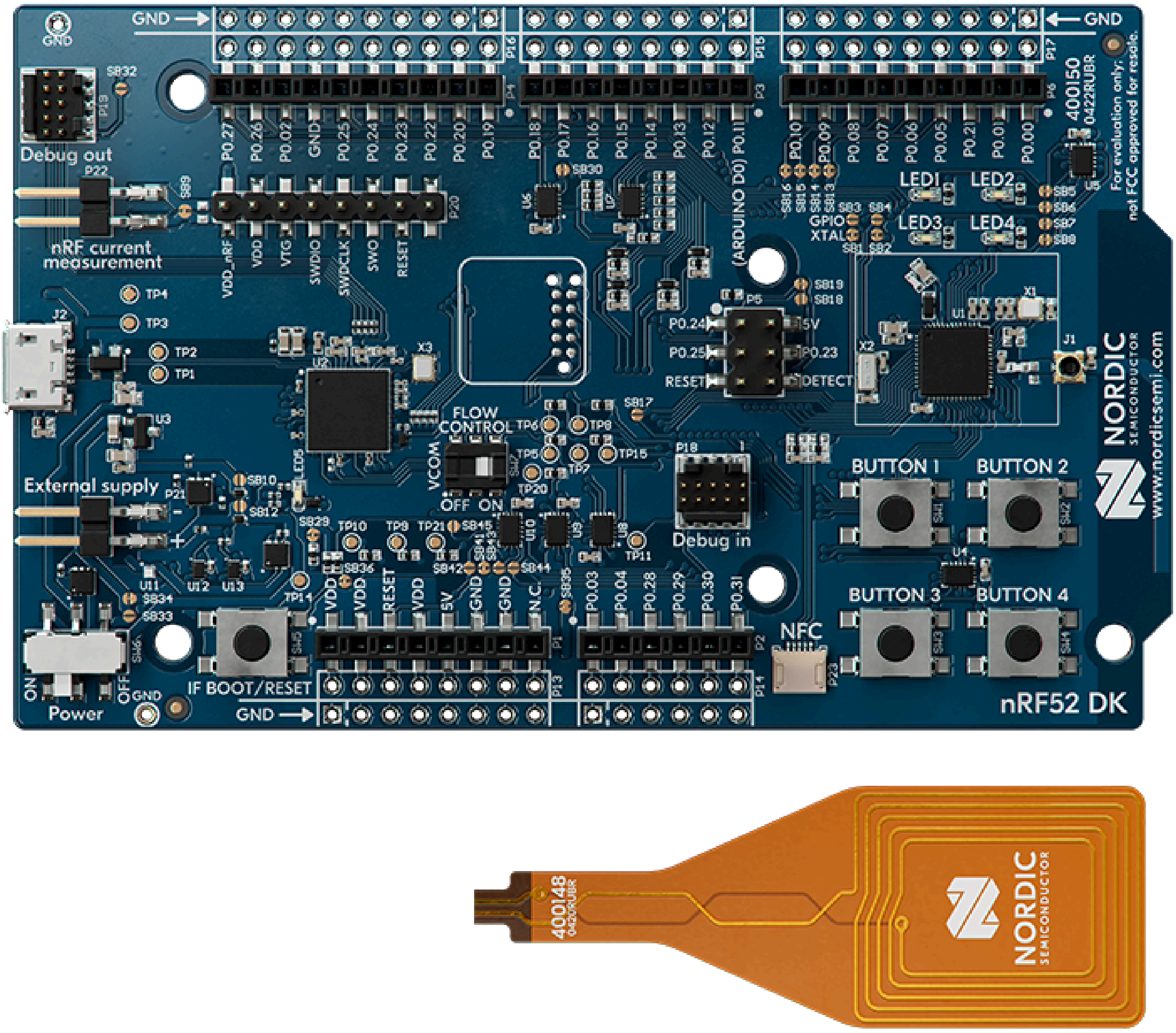


Figure 1: nRF52 DK (PCA10040) and NFC antenna

Hardware files

The hardware design files for the nRF52 DK are available on the nRF52832 product page. They include.

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

3 Interface MCU

The interface MCU on the nRF52 DK runs SEGGER J-Link *Onboard (OB)* interface firmware. It is used to program and debug the application firmware of the nRF52832 *SoC*.

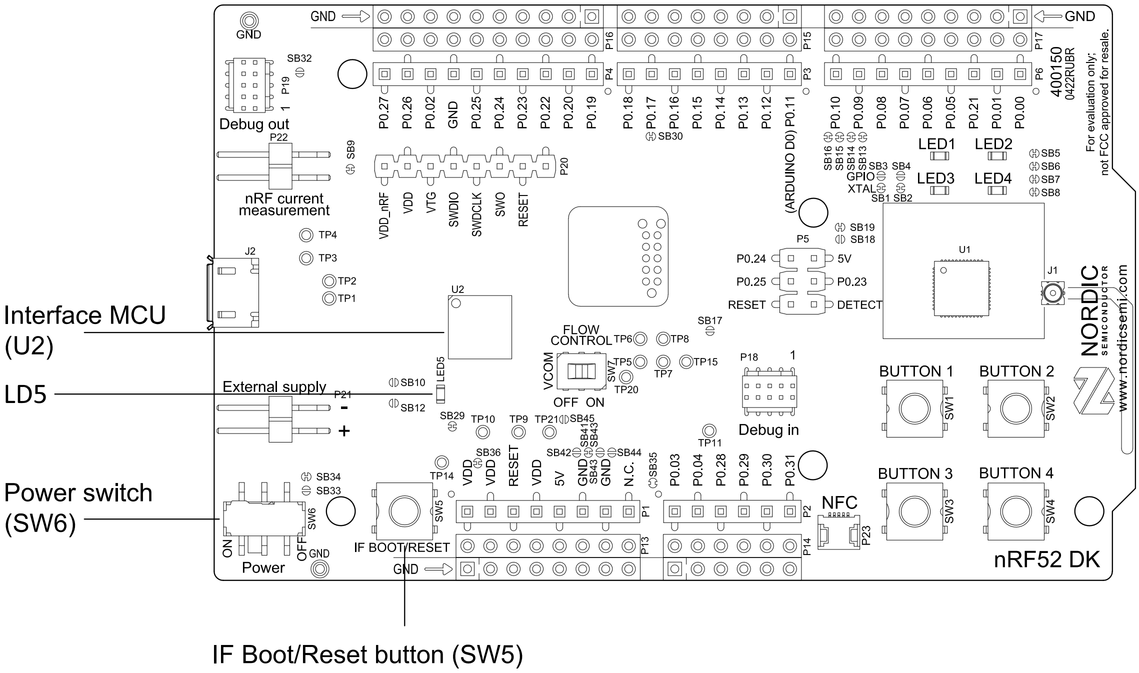


Figure 2: Interface MCU

3.1 Reset button

Reset button (**SW5**) is connected to the interface MCU on the *DK* and resets the nRF52832 *SoC* or any device connected to the external programming connectors.

For the reset button to work, pin reset on `P0.21` must be enabled in the embedded firmware of the nRF52832*SoC*.

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)
- Optional *Hardware Flow Control (HWFC)*
- Tri-stated UART lines when no terminal is connected

The following table lists the nRF52832 *SoC* UART GPIO pins and their signals.

GPIO nRF52832	nRF52832 UART	Interface MCU UART
`P0.05`	RTS	CTS
`P0.06`	TXD	RXD
`P0.07`	CTS	RTS
`P0.08`	RXD	TXD

Table 1: Relationship of UART connections on nRF52832 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.

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 nRF52 DK (PCA10040) 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

nRF52 DK hardware drawings show both sides of the PCA10040.

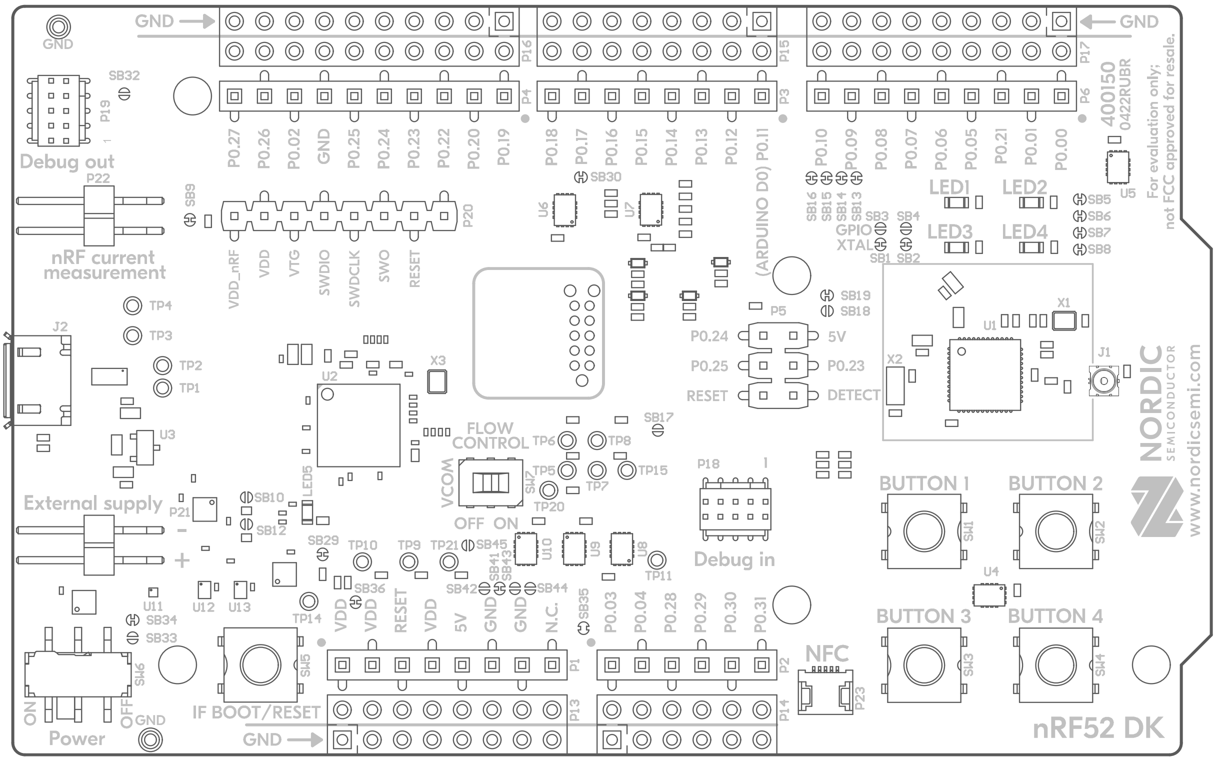


Figure 3: nRF52 DK front view

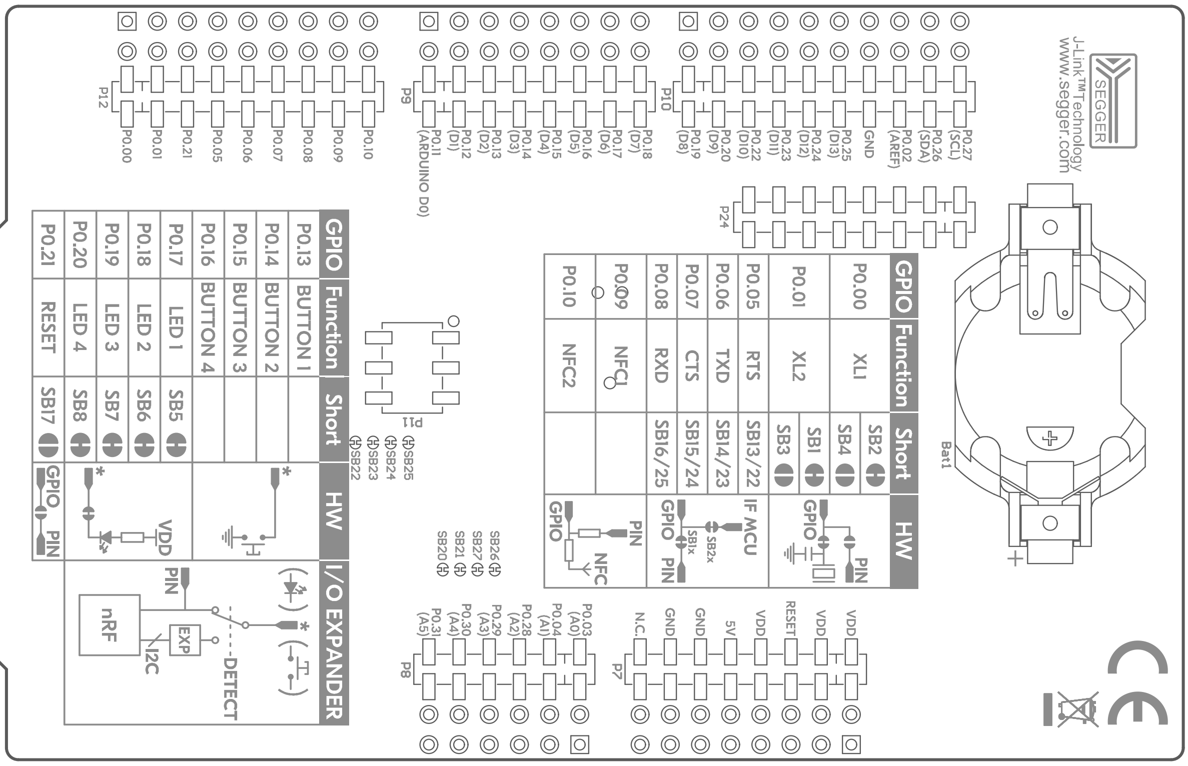


Figure 4: nRF52 DK back view

4.2 Block diagram

The block diagram illustrates the nRF52 DK functional architecture.

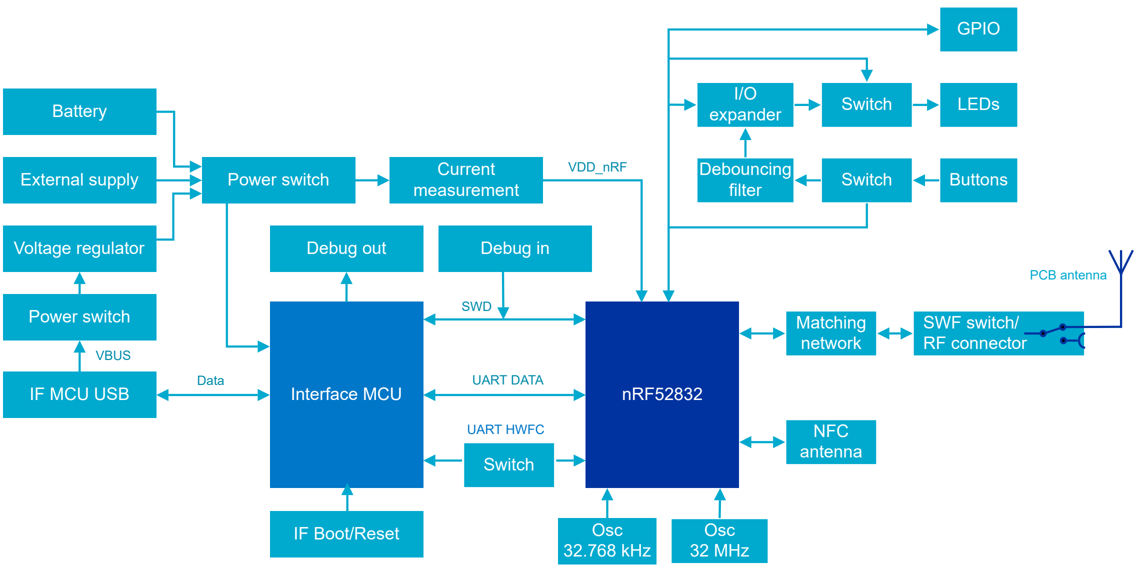


Figure 5: Block diagram

4.3 Power supply

The nRF52 DK has three power options: USB (5 V), external power supply (1.7 V to 3.6 V), and coin cell battery.

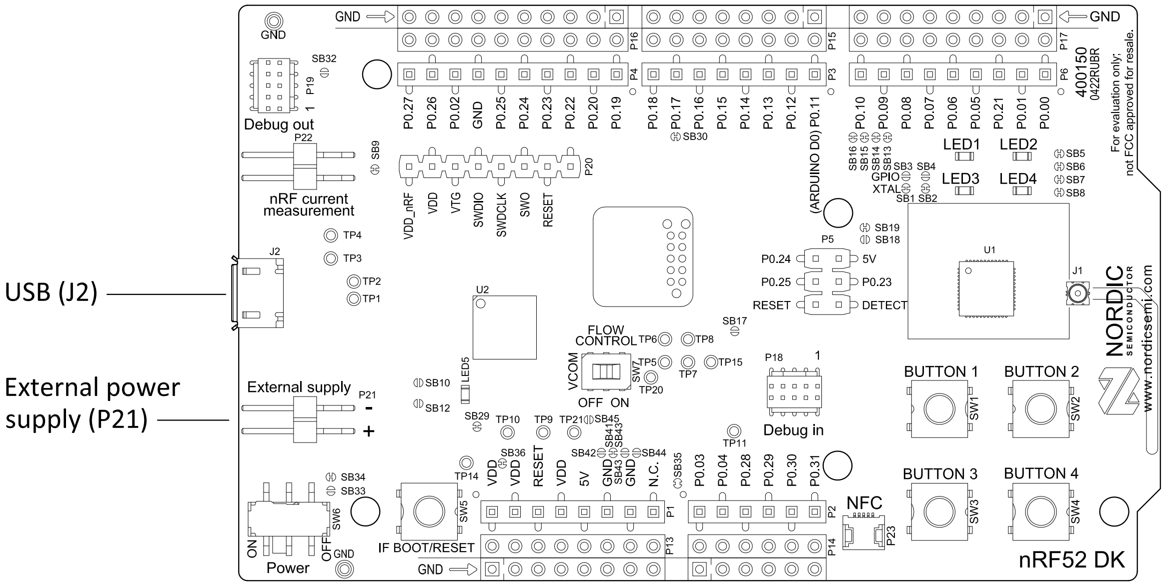


Figure 6: Power supply options (front)

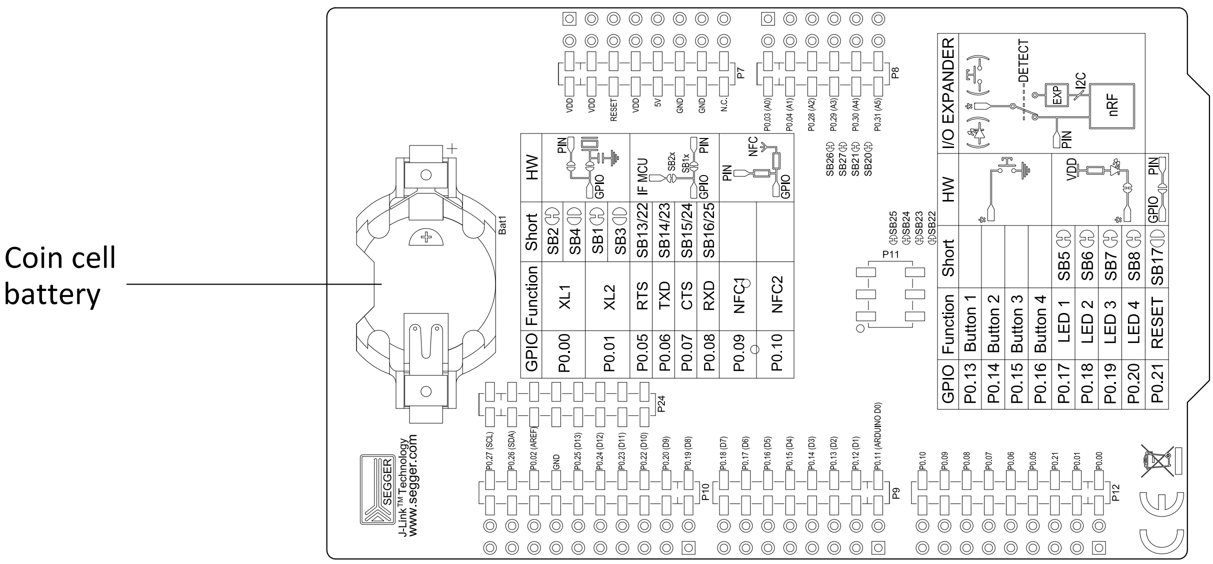


Figure 7: Power supply options (back)

The 5 V from the USB is regulated down to 3.3 V through an onboard voltage regulator. The battery and external power supply are not regulated. The power from the regulator and external supply is routed through diodes for reverse voltage protection (**D1** and **D7**), where the circuit is supplied from the source with the highest voltage. The power from the battery is routed through a load switch and a transistor. These are controlled by the regulator voltage or external voltage, so if any of the two other sources are present, the battery is disconnected from the circuit.

> **Note:** When USB is not powered, the interface MCU is in dormant state and draws an additional current of approximately 20 mA to maintain the reset button functionality. This affects current consumption, but not the nRF52832 current measurements, as described in Current measurement on page 30.

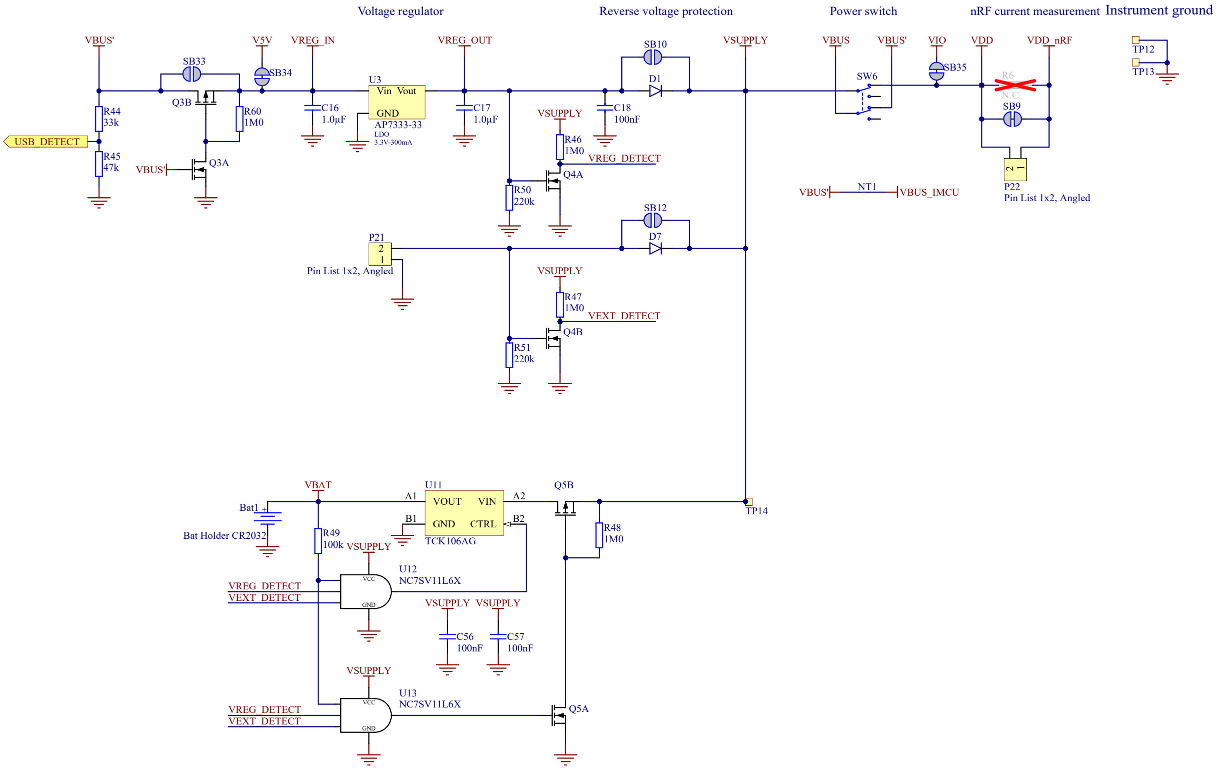


Figure 8: Power supply circuitry

The reverse voltage protection diodes for the regulator and external supply add a voltage drop to the supply voltage of the circuit. To avoid this voltage drop, the diodes can be bypassed by shorting one or more solder bridges.

Power source	Protection bypass	Voltage level
USB	**SB10**	3.3 V
External supply	**SB12**	1.7 V to 3.6 V

Table 2: Protection diode bypass solder bridges

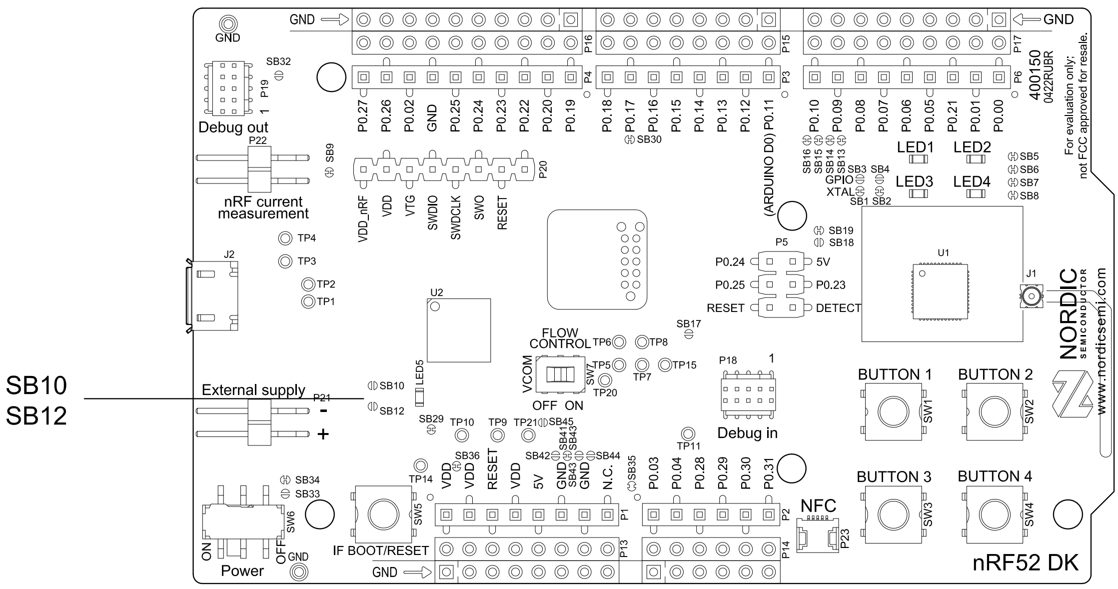


Figure 9: Protection diode bypass solder bridges

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

4.4 Connector interface

Access to the nRF52832 *General-Purpose Input/Output (GPIO)*s is available from connectors **P2**, **P3**, **P4**, **P5**, and **P6**.

The **P1** connector provides access to ground and power on the nRF52 DK.

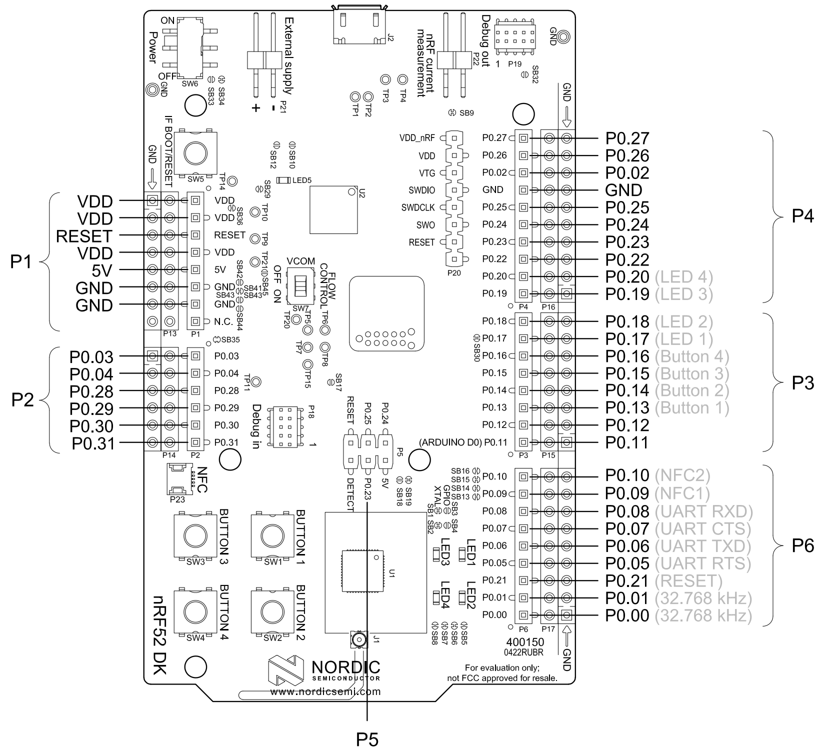


Figure 10: nRF52 DK connectors

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 nRF52 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 19 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 8 for more information.
- `P0.09` and `P0.10` are by default used by signals NFC1 and NFC2. See NFC antenna interface on page 25 for more information.
- `P0.13`–`P0.20` are by default connected to the buttons and LEDs. See Buttons and LEDs on page 16 for more information.

When the nRF52 DK is used as a shield together with an Arduino standard motherboard, the Arduino signals are routed as shown in the following figure.

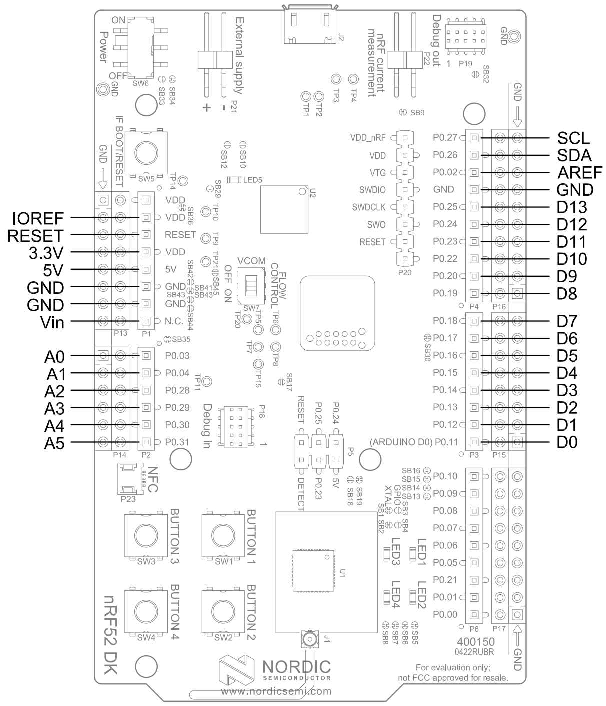


Figure 11: Arduino signals routing on the nRF52 DK

4.5 Buttons and LEDs

The four buttons and four LEDs on the nRF52 DK are connected to dedicated GPIOs on the nRF52832 *SoC*.

Part	GPIO	Short
BUTTON 1	`P0.13`	-
BUTTON 2	`P0.14`	-
BUTTON 3	`P0.15`	-
BUTTON 4	`P0.16`	-
LED1	`P0.17`	**SB5**
LED2	`P0.18`	**SB6**
LED3	`P0.19`	**SB7**
LED4	`P0.20`	**SB8**

Table 3: Button and LED connections

If `P0.17`–`P0.20` are needed elsewhere, the LEDs can be disconnected by cutting the short on **SB5–SB8**. See the following figure for more information.

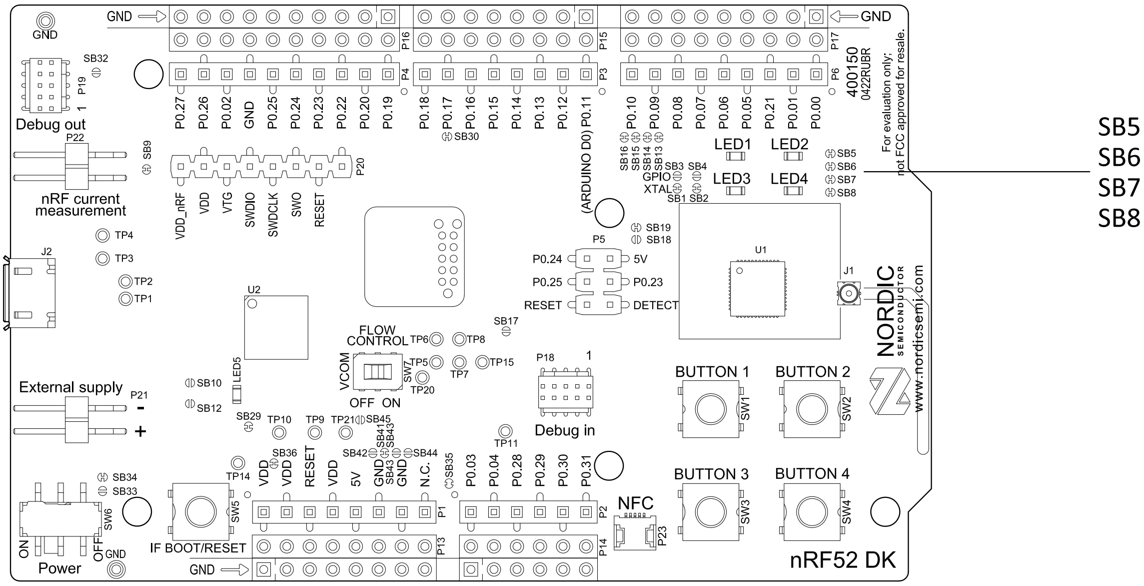


Figure 12: Disconnecting the LEDs

The LEDs and buttons can also be disconnected by using the I/O expander as described in I/O expander for buttons and LEDs on page 17.

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 **P0.13—P0.16** 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.

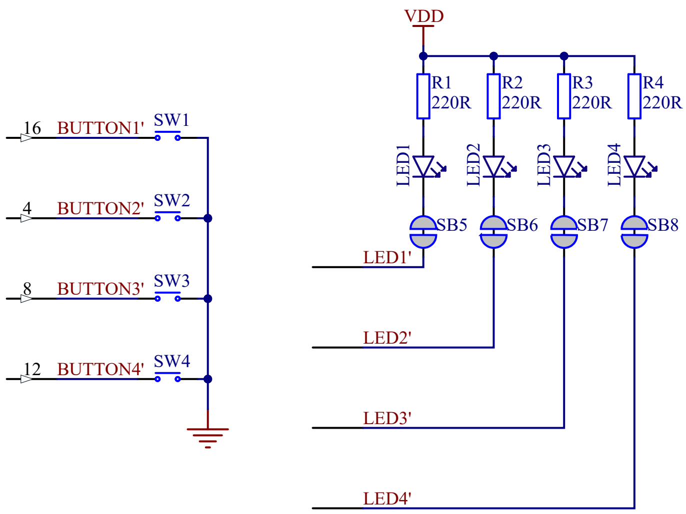


Figure 13: Button and LED configuration

4.5.1 I/O expander for buttons and LEDs

The onboard *GPIOs* for the buttons and LEDs can conflict with boards that follow the Arduino standard. To avoid such conflicts, the nRF52 DK has an I/O expander.

GPIO	Part	Arduino signal
`P0.13`	**BUTTON 1**	D2
`P0.14`	**BUTTON 2**	D3
`P0.15`	**BUTTON 3**	D4
`P0.16`	**BUTTON 4**	D5
`P0.17`	**LED1**	D6
`P0.18`	**LED2**	D7
`P0.19`	**LED3**	D8
`P0.20`	**LED4**	D9

Table 4: GPIO connection

The I/O expander releases these GPIOs for general use when the nRF52 DK is used together with boards that follow the Arduino standard. The I/O expander can be permanently enabled by shorting solder bridge **SB18**, or permanently disabled by cutting the shorting track on **SB19**. You must also short **SB18** when cutting **SB19** for full compatibility with the Arduino standard.

The I/O expander can be temporarily enabled by connecting SHIELD **`DETECT`** to ground.

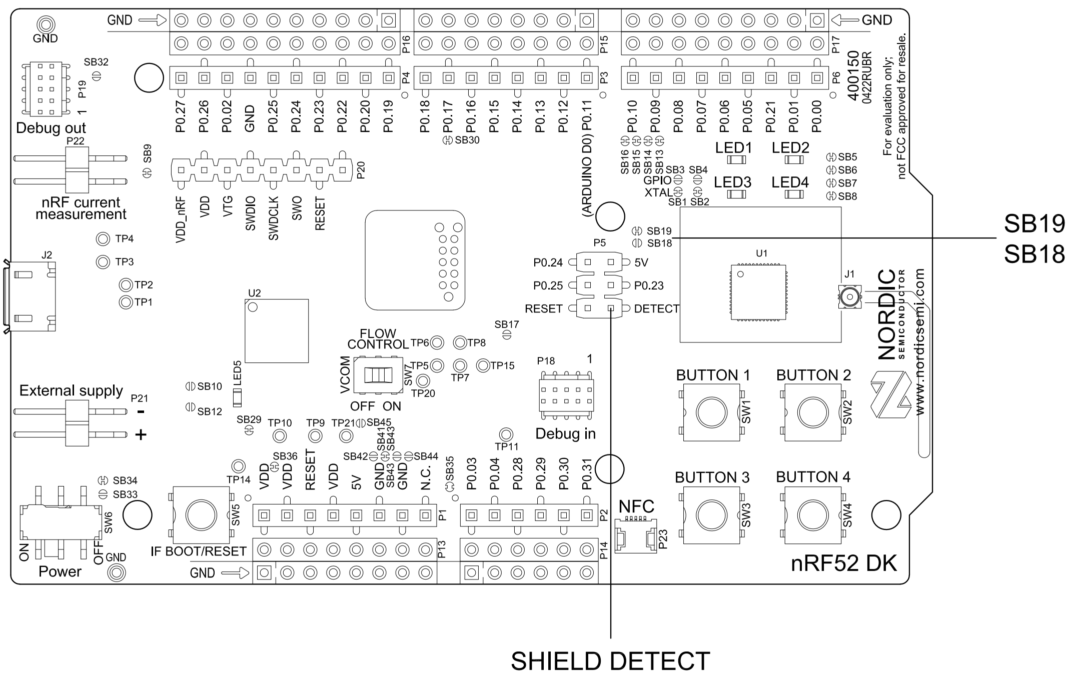


Figure 14: Enable or disable I/Os for Arduino standard

In addition to the buttons and LEDs, the following GPIOs are used for the I/O expander:

I/O expander signal	GPIO
/INT	`P0.17`
SDA	`P0.26`
SCL	`P0.27`

Table 5: I/O expander connection

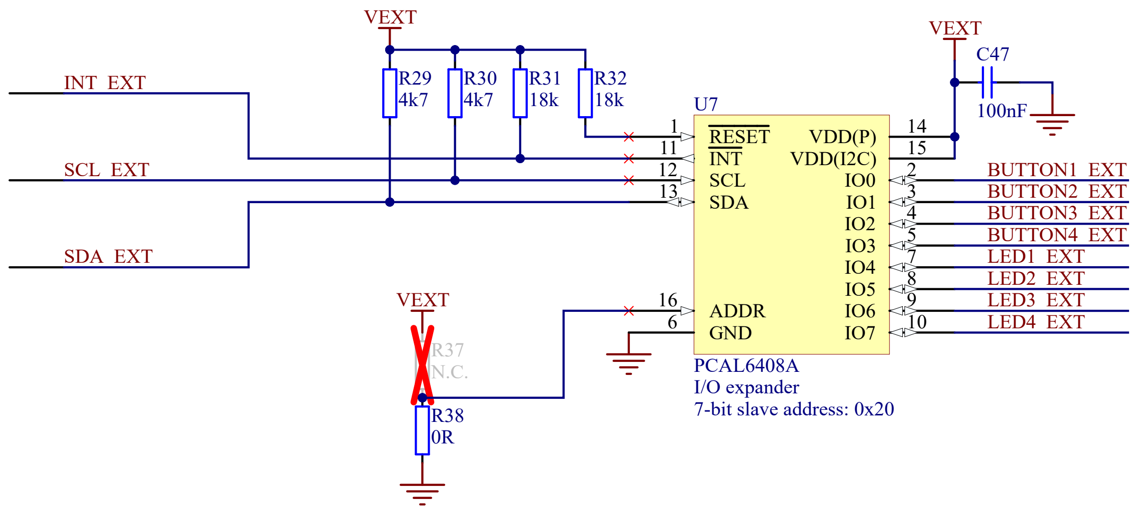


Figure 15: I/O expander schematic

Note: Software debouncing is not needed when using the I/O expander. Each button on the nRF52 DK has a debouncing filter.

4.6 32.768 kHz crystal

The nRF52832 *SoC* can use an optional 32.768 kHz crystal (**X2**) for higher accuracy and lower average power consumption.

On the nRF52 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.

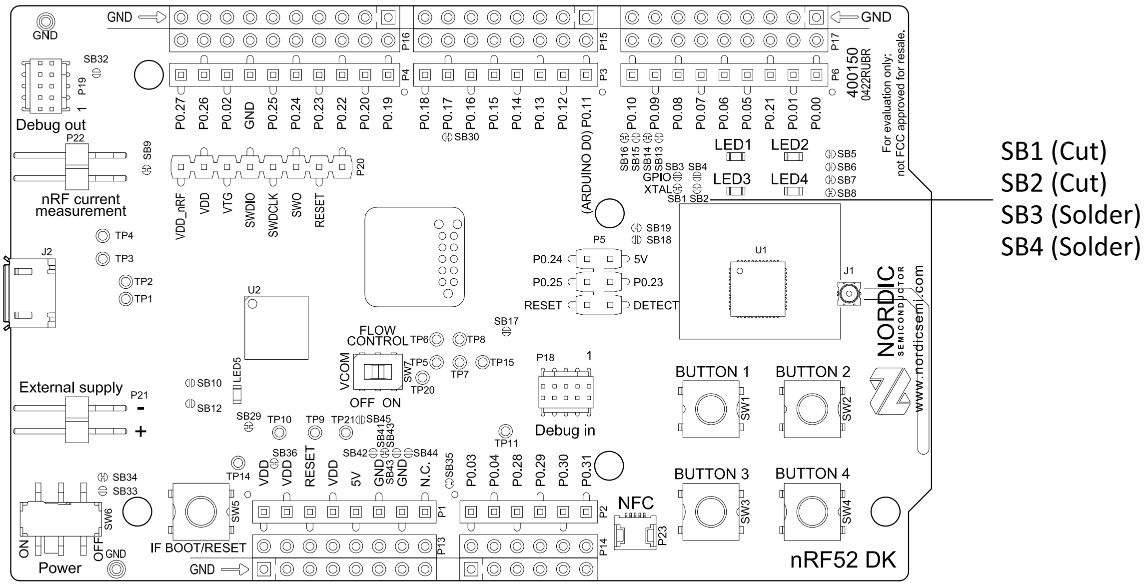


*Figure 16: Configuring **P0.00** and **P0.01***

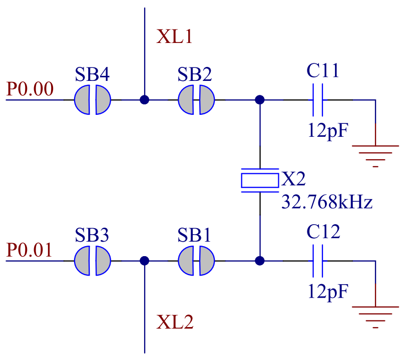


*Figure 17: 32.768 kHz crystal and **SB1–SB4***

4.7 Debug input

The Debug in connector (**P18**) makes it possible to connect external debuggers while running on a battery or an external power supply.

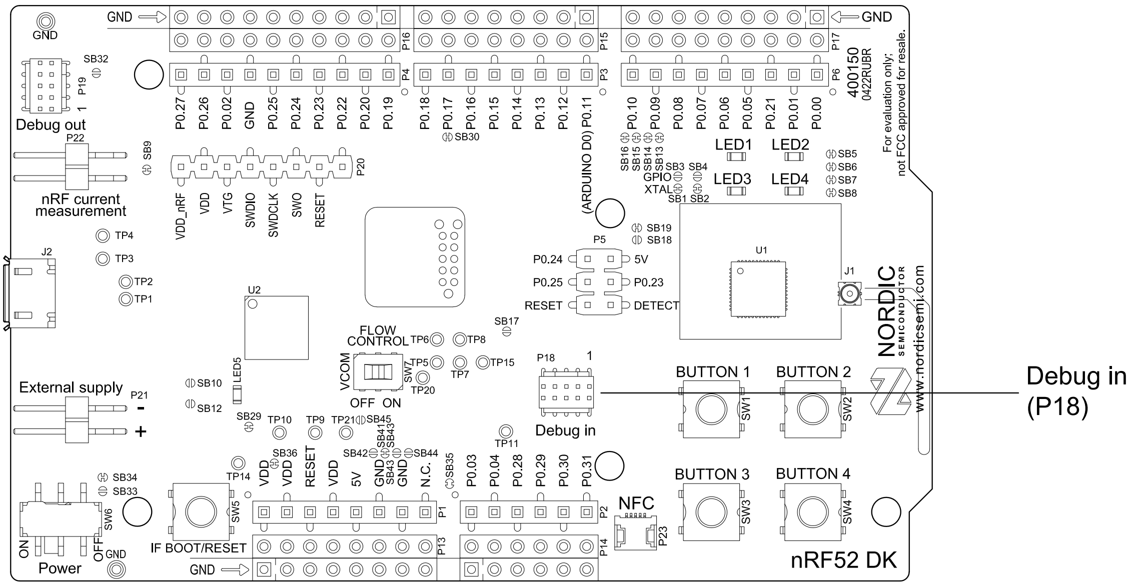


Figure 18: Debug input

4.8 Debug out for programming external boards

The nRF52 DK supports programming and debugging external boards with an nRF51, nRF52, and nRF53 Series *SoC*, or the nRF91 Series *System in Package (SiP)*.

The interface MCU on the nRF52 DK runs SEGGER J-Link *OB* interface firmware. It is used to program and debug the application firmware of the nRF52832 *SoC*, by default.

To program/debug an external board instead, connect to the Debug out connector (**P19**) using a 10-pin cable or use **P20** for custom connection.

Note: It is recommended to power the external board separately from the DK. The voltage on the external board must match that of the DK. When the DK is powered through the USB connector, the voltage is 3V.

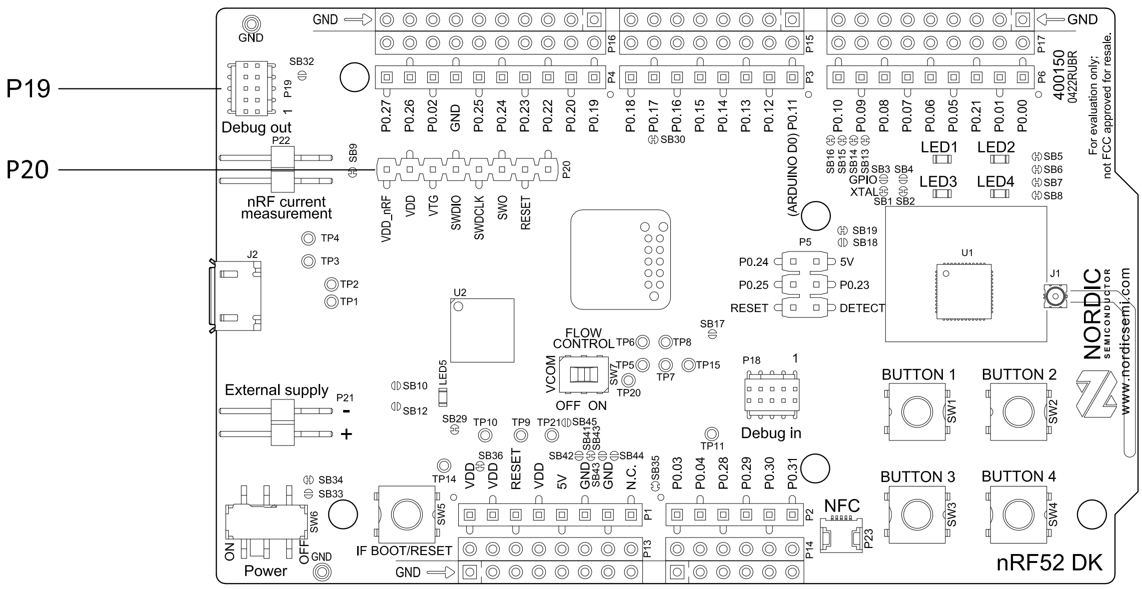


Figure 19: Debug output connectors

4.8.1 Programming an external board

For boards with a standard 10-pin *Serial Wire Debug (SWD)* connector, or a connector that supports a standard 10-pin flat cable, it is recommended to connect to **P19**.

Connect the boards as shown in the following figure.

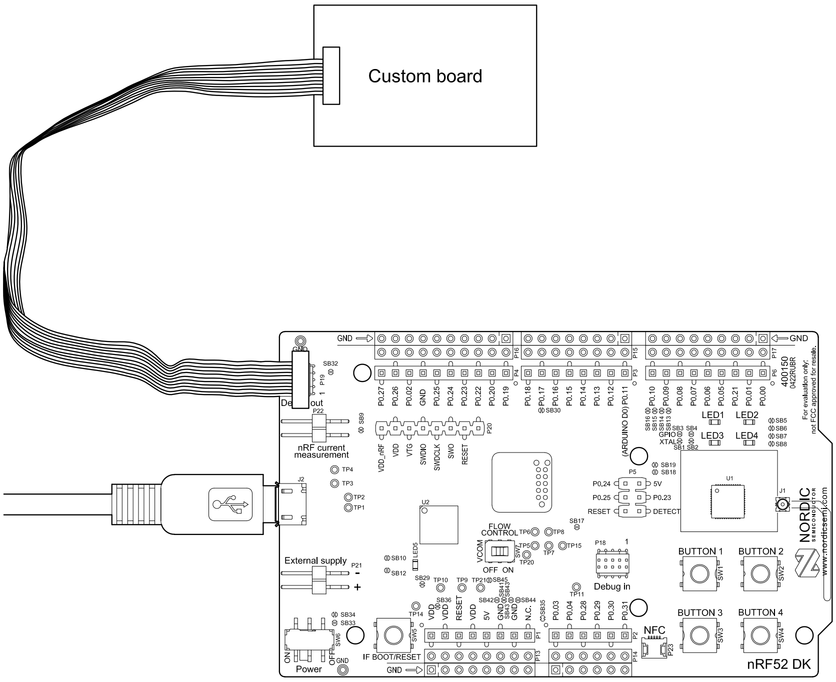


Figure 20: Connecting an external board to P19

It is recommended to power the external board separately from the DK. The voltage on the external board must match that of the DK. When the DK is powered through the USB connector, the voltage is 3V.

When **pin 3** (**`SWD0_SELECT`**) of **P19** is connected to **GND** through the 10-pin flat cable, the interface MCU programs or debugs the target chip on the external board instead of the onboard nRF52832 *SoC*.

If it is inconvenient to have a separate power supply on the external board, the nRF52 DK can supply power through the Debug out connector **P19**. To enable this, short solder bridge **SB32**.

CAUTION: To avoid damaging your board, do not connect a separate power supply to the external board when **SB32** is shorted.

CAUTION: To avoid overloading the power supply and damaging the DK, use VDD and keep the supply below 100 mA. Do not use a *Lithium-polymer (Li-Poly)* source.

The following section includes an illustration of the **P19** connector pinout with a description table.

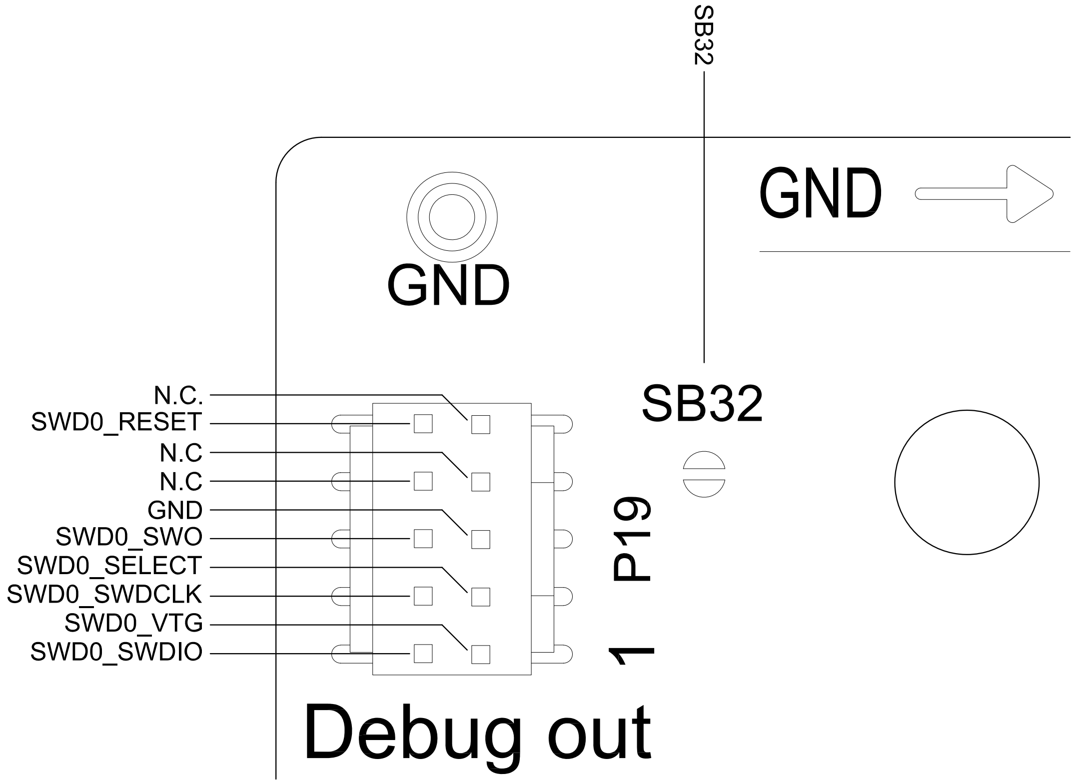


*Figure 21: Debug output connector **P19***

Pin number	Signal	Description
1	SWD0_VTG	Optional voltage supply to the external target
2	SWD0_SWDIO	*SWD* Data Input/Output
3	SWD0_SELECT	Debug out select signal, connect to ground on external board
4	SWD0_SWDCLK	Serial Wire Clock line
5	GND	Ground
6	SWD0_SW0	The *Serial Wire Output (SWO)* line is not used for programming and debugging over SWD
7	N.C.	Not used
8	N.C.	Not used
9	N.C.	Not used
10	SWD0_RESET	Reset line

Table 6: Connector P19 pinout for programming external targets

4.8.2 Programming a board with custom connections

For boards with custom pin connections for programming and debugging, use debug output on connector **P20**.

Connect the boards as shown in the following figure.

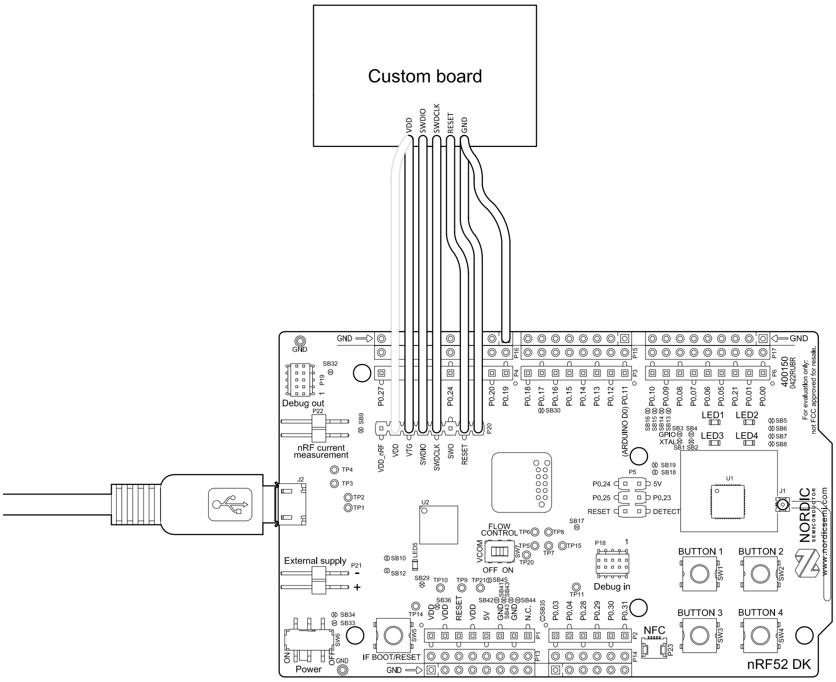


Figure 22: Connecting an external board to P20

It is recommended to power the external board separately from the DK. The voltage on the external board must match that of the DK. When the DK is powered through the USB connector, the voltage is 3V.

When the interface MCU detects the voltage of the external board on pin 3 (`SWD1_VTG`) of **P20** it programs or debugs the target chip on the external board instead of the onboard nRF52832 *SoC*.

If it is inconvenient to have a separate power supply on the external board, the nRF52 DK can supply power through pin 2 (`VDD`) of **P20**. The connection is shown with a grey outline in Figure 22: Connecting an external board to P20 on page 24. If the interface MCU detects boards connected to both **P19** and **P20**, it programs or debugs the target connected to **P19** by default.

CAUTION: To avoid overloading the power supply and damaging the DK, use VDD and keep the supply below 100 mA. Do not use a *Li-Poly* source.

CAUTION: To avoid damaging your board, do not connect a separate power supply to the external board when VDD of nRF52 DK is connected to the external board.

The following section includes an illustration of the **P20** connector pinout with a description table.

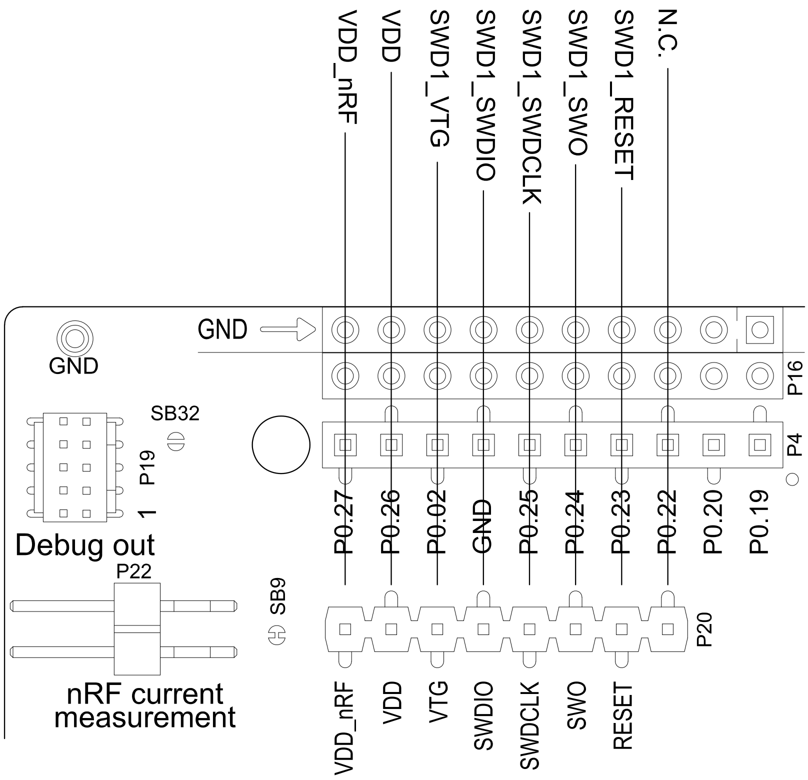


*Figure 23: Debug output connector **P20***

Pin number	Signal	Description
1	VDD_nRF	nRF52832 SoC power domain
2	VDD	Main nRF52 DK power domain
3	SWD1_VTG	Voltage supply from the external target. Used as an enable signal for activating SWD1
4	SWD1_SWDIO	*SWD* data line
5	SWD1_SWDCLK	SWD clock line
6	SWD1_SWO	The *SWO* line is not needed for programming and debugging over SWD
7	SWD1_RESET	Reset line
8	N.C.	Not used

Table 7: Pinout of connector P20 for programming external targets

4.9 NFC antenna interface

The nRF52 DK supports an *NFC* tag.

NFC-A Listen Mode operation is supported on the nRF52832 *SoC*. The NFC antenna input is available on connector **P23** on the nRF52 DK.

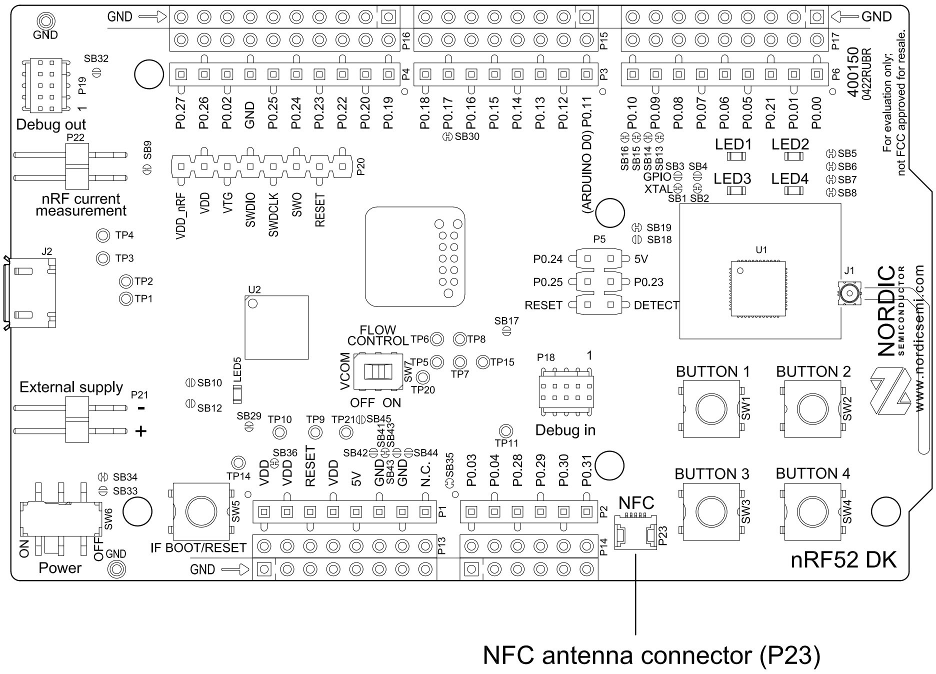


Figure 24: NFC antenna connector

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 *Integrated Development Environment (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 **P0.09** and **P0.10** are by default configured to use the NFC antenna. To use these pins as GPIO, **R25** and **R26** must be not connected (NC) and **R27** and **R28** must be shorted with an OR resistor.

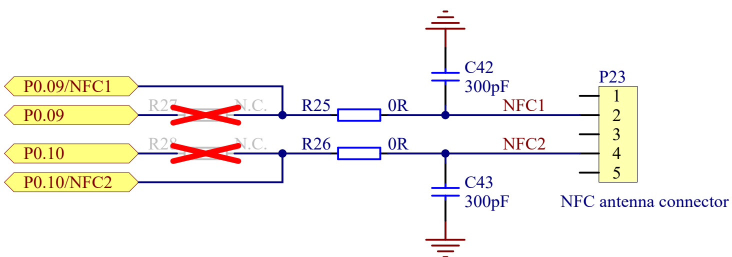


Figure 25: NFC input

4.10 Solder bridge and test point overview

The nRF52 DK has a range of solder bridges for enabling or disabling functionality on the *DK*. It also has test points available for use. Changes to these are not needed for normal use of the *DK*.

The following table is an overview of the solder bridges on the nRF52 DK.

Solder bridge	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	Closed	Cut for current measurements
SB10	Open	Short to bypass the reverse voltage protection diode on the USB power
SB12	Open	Short to bypass the reverse voltage protection diode on the external supply power
SB13	Closed	Cut to disconnect **P0.05** from the connector interface
SB14	Closed	Cut to disconnect **P0.06** from the connector interface
SB15	Closed	Cut to disconnect **P0.07** from the connector interface
SB16	Closed	Cut to disconnect **P0.08** from the connector interface
SB17	Open	Short to connect **P0.21** to the connector interface **RESET**
SB18	Open	Short to permanently enable the I/O expander
SB19	Closed	Cut to permanently disable the I/O expander
SB20	Closed	Cut to isolate **SWDIO** from nRF52832 to the interface MCU
SB21	Closed	Cut to isolate **SWDCLK** from nRF52832 to the interface MCU
SB22	Closed	Cut to isolate **P0.05** from nRF52832 to the interface MCU
SB23	Closed	Cut to isolate **P0.06** from nRF52832 to the interface MCU
SB24	Closed	Cut to isolate **P0.07** from nRF52832 to the interface MCU
SB25	Closed	Cut to isolate **P0.08** from nRF52832 to the interface MCU
SB26	Closed	Cut to isolate **P0.21** from nRF52832 to the interface MCU
SB27	Closed	Cut to isolate **P0.18** from nRF52832 to the interface MCU
SB29	Closed	Cut to disable power for interface MCU
SB30	Closed	Cut to isolate **P0.17** from I/O expander interrupt line

Solder bridge	Default	Function
SB32	Open	Short to supply power through the **Debug out** connector
SB33	Open	Short to connect the **5V** supply from the connector interface to VBUS
SB34	Closed	Cut to disconnect the **5V** supply to the connector interface
SB35	Closed	Cut to disconnect **VDD** supply to the connector interface
SB36	Closed	Cut to disconnect the **IF BOOT/RESET** button from the IMCU
SB41	Closed	No function on the current BOM
SB42	Open	Short to connect **IF BOOT/RESET** button to **RESET** pin on the Arduino interface
SB43	Open	Short to connect the **RESET** pin on the Arduino interface to the nRF52832 **P0.21/RESET** pin
SB44	Open	No function on the current BOM
SB45	Open	No function on the current BOM

Table 8: Solder bridge configuration for nRF52 DK (v3.0.0 and higher)

The following table is a complete overview of the test points on the nRF52 DK.

Designator	Signal	Description	Size	Layer
TP1	IMCU_P	USB connector D+	1.0 mm	Top
TP2	IMCU_N	USB connector D-	1.0 mm	Top
TP3	VBUS	USB voltage before power switch	1.0 mm	Top
TP4	GND	Ground	1.0 mm	Top
TP5	VCOM0_CTS	UART CTS	1.0 mm	Top
TP6	VCOM0_RxD	UART RXD	1.0 mm	Top
TP7	VCOM0_RTS	UART RTS	1.0 mm	Top
TP8	VCOM0_TxD	UART TXD	1.0 mm	Top
TP9	SWD3_SWDIO	nRF52832 SWDIO signal, IMCU side of **SB20**	1.0 mm	Top
TP10	SWD3_SWDCLK	nRF52832 SWDCLK signal, IMCU side of **SB21**	1.0 mm	Top
TP11	P0.21	nRF52832 **`P0.21/RESET`**	1.0 mm	Top
TP12	GND	Ground	Through hole	Top
TP13	GND	Ground	Through hole	Top
TP14	VSUPPLY	Board supply power before power switch	1.0 mm	Top
TP15	SHIELD_DETECT	Detect signal for Arduino compatible shield	1.0 mm	Top
TP20	SWD3_RESET	nRF52832 RESET signal, IMCU side of **SB26**	1.0 mm	Top
TP21	SWD3_SWO	nRF52832 SWO signal, IMCU side of **SB27**	1.0 mm	Top

Table 9: Test point overview

5 Current measurement

The current drawn by the nRF52832 *SoC* can be monitored on the nRF52 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.

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. It is recommended to power the DK from a coin cell battery or external power supply on connector **P21** (1.7 V to 3.6 V).

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

After programming the nRF52832 SoC, the USB must be disconnected and the DK power cycled to reset the debugger chip before current measurement.

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 nRF52832 *SoC* and the rest of the DK, and bypass protection components in the power supply chain.

- To put **P22** in series with the load, cut the PCB track shorting solder bridge **SB9**.
- If using an external power supply, short solder bridge **SB12** to bypass the protection diode, which would otherwise give a voltage drop.

  > **Note:** While **SB12** is shorted, the DK must not be powered from the USB if there is an external supply connected because the protection diode has been bypassed.

- To restore normal DK function after measurement:

  1. Solder **SB9** or apply a jumper on **P22**.
  2. Cut or desolder **SB12** to reconnect the protection diode.

- To reprogram the nRF52832 SoC while the DK is prepared for current measurement, disconnect external supply, ensure there is no battery inserted, remove measurement devices from **P22**, add a jumper to **P22**, and then connect the USB cable.

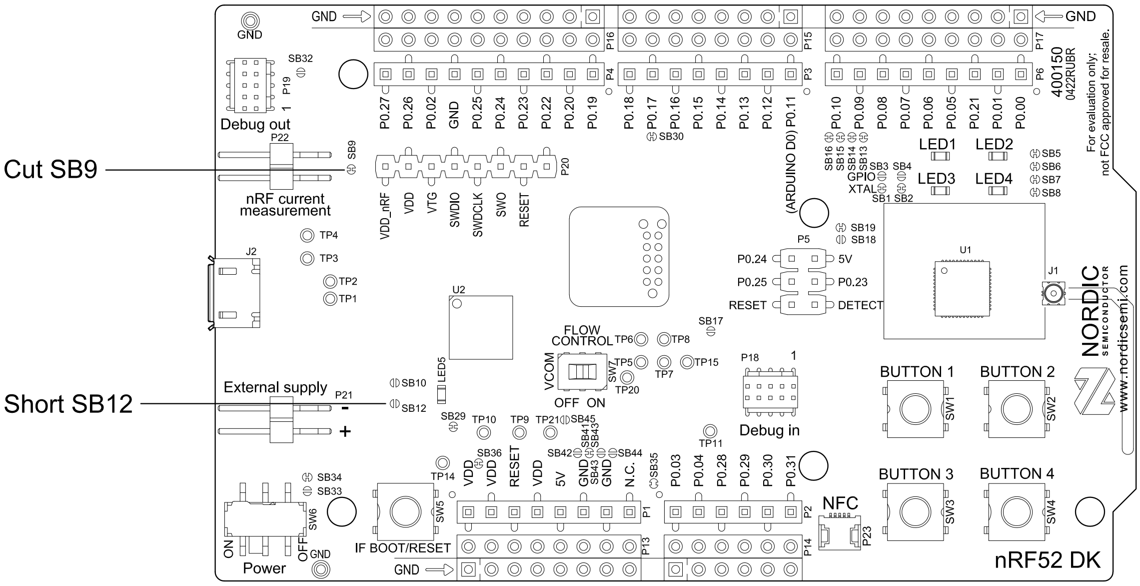


Figure 26: Preparing the DK for current measurements

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 30.

1. Mount a 10 Ω resistor on the footprint for **R6**.
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.

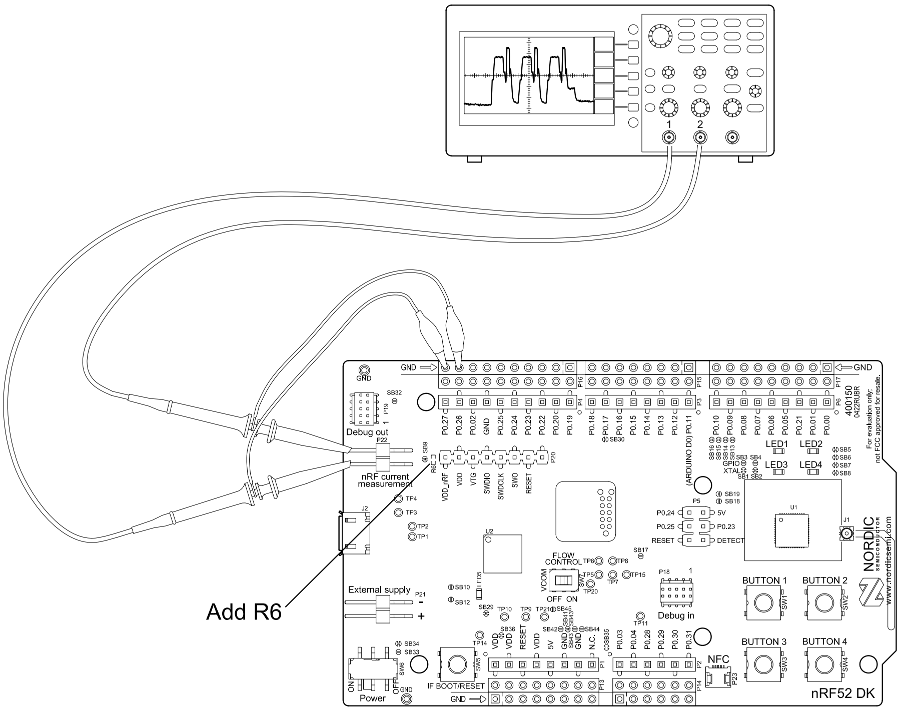


Figure 27: Current measurement with an oscilloscope

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 nRF52832 *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 30.

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

To measure the average current drawn by the nRF52832 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.

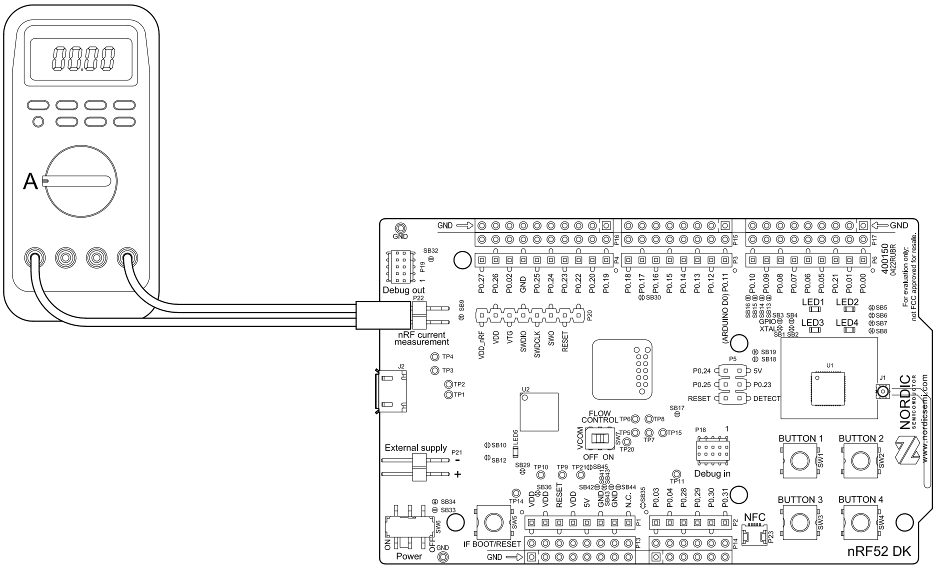


Figure 28: Current measurement with an ampere meter

6 RF measurements

The nRF52 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 nRF52832 *SoC* to the test probe.

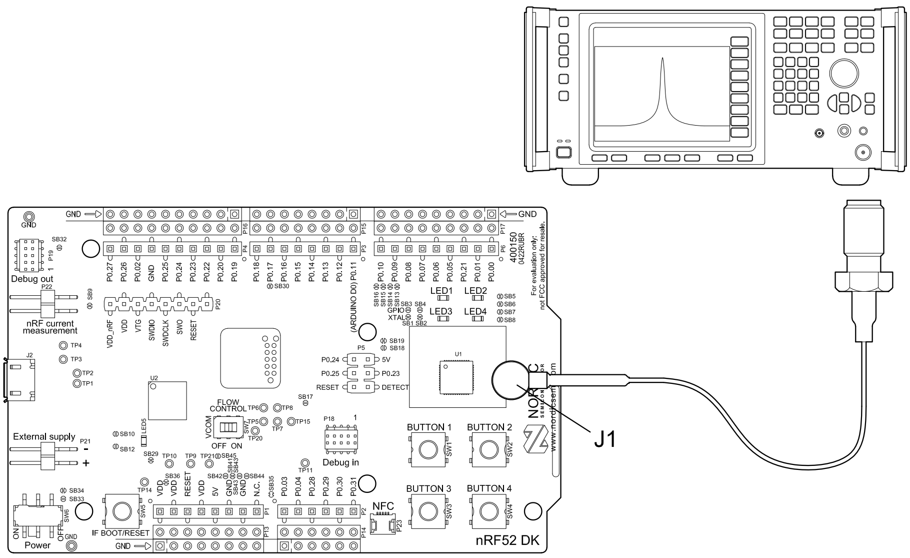


Figure 29: Connecting a spectrum analyzer

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 10: 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 35.

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

- nRF52832 Product Specification
- nRF52832 Compatibility Matrix
- nRF52832 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