nRF8001 Development Kit

User Guide v2.0

1 Introduction

The nRF8001 Development Kit v2.0 for Arduino is an easy-to-use *Bluetooth* low energy kit that is designed for use with an Arduino board or an Arduino compatible board. By using the development kit with our open source ble_sdk_arduino SDK and Arduino's hardware and software, you can start creating designs and speedily prototype with little expenditure.

1.1 Minimum requirements

Below are the minimum requirements for the nRF8001 Development Kit:

- Arduino UNO or an Arduino compatible board
- Computer with 2 USB ports
- Windows XP and Windows 7

1.1.1 Support for Linux and OS X

The Arduino development environment supports development on OS X and Linux and the ble_sdk_arduino SDK is supported on the OS X and Linux version of Arduino.

Note: The Master Control Panel is Windows based and is not supported on OS X and Linux.

1.2 Documentation

The following documentation is recommended reading:

Document	Description
nRF8001 Product Specification	Defines the nRF8001 hardware and electrical specifications and describes the ACI (Application Controller Interface).
nRFgo Studio help	nRFgo Studio help documents the functionality of the nRFgo Studio.

1.3 Required tools

The following list of software tools are necessary for using this development kit. Installation is covered in ***Chapter 3 "Quick start"*** on page 6 and ***Chapter 4 "Typical setup"*** on page 7.

Nordic Tools	Description
nRFgo Studio	nRFgo Studio is our tool to program and configure devices. It is the visual editor for GATT (Generic Attribute Profile) server, GATT Client and GAP (Generic Access Profile) settings. nRFgo Studio is also used for *Bluetooth* Direct Test Mode (DTM). For more information, see the help in nRFgo Studio.
ble_sdk_arduino	The ble_sdk_arduino Software Development Kit (SDK) provides source code of examples and libraries forming the base of your application development. The ble_sdk_arduino SDK includes: • Example code • *Bluetooth* profile examples • Drivers • Libraries For more information, see the documentation packaged with the ble_sdk_arduino SDK.
Master Control Panel	The Master Control Panel is the software tool that is used with the Master Emulator Development Dongle (PCA10000) to act as a *Bluetooth* low energy peer device. You can test your application's wireless connection with this tool. For more information, see the help files in the Master Control Panel.

Third party tools	Description
Arduino hardware and software	Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. See ***http://arduino.cc/*** for purchasing/downloading hardware and software components.

2 Kit content

The nRF8001 Development Kit consists of hardware and software components, firmware libraries, and example applications.

2.1 Hardware components

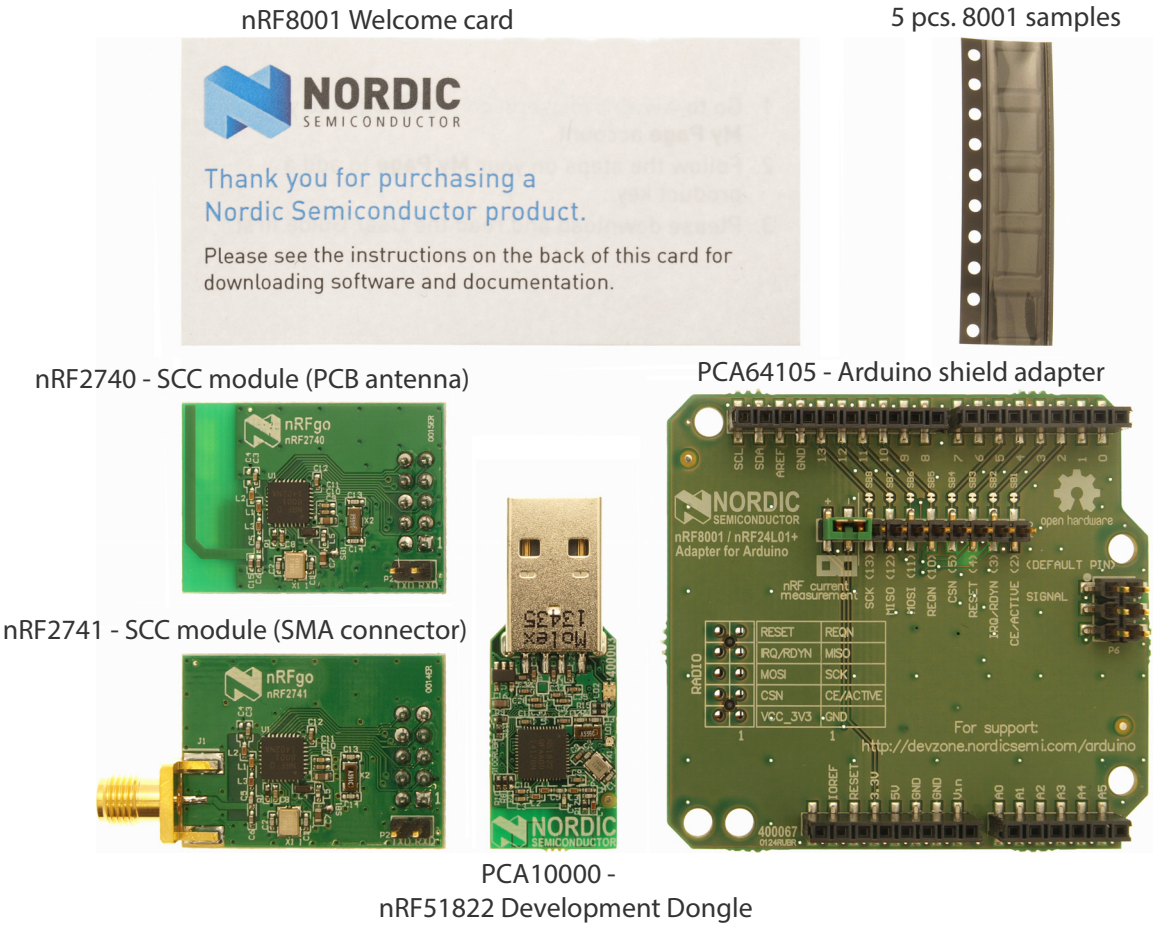


Figure 1 *Hardware components*

For more information on the hardware modules see ***Chapter 6 "Hardware description"*** on page 11.

2.2 Downloadable content

The nRF8001 Development Kit includes firmware source code, documentation, hardware schematics, and layout files. To access these files, log in to your My Page account, enter your product key, and download the files. Instructions can be found in ***Chapter 3 "Quick start"*** on page 6.

2.2.1 nRF8001 DK software content

- nRFgo Studio
- ble_sdk_arduino Software Development Kit (SDK)
  - Precompiled HEX files
  - Source code
  - Keil ARM project files
- Master Control Panel
- nRF Sniffer

2.2.2 nRF8001 DK documentation

- *nRF8001 Development Kit User Guide*
- *nRF8001 Product Specification*

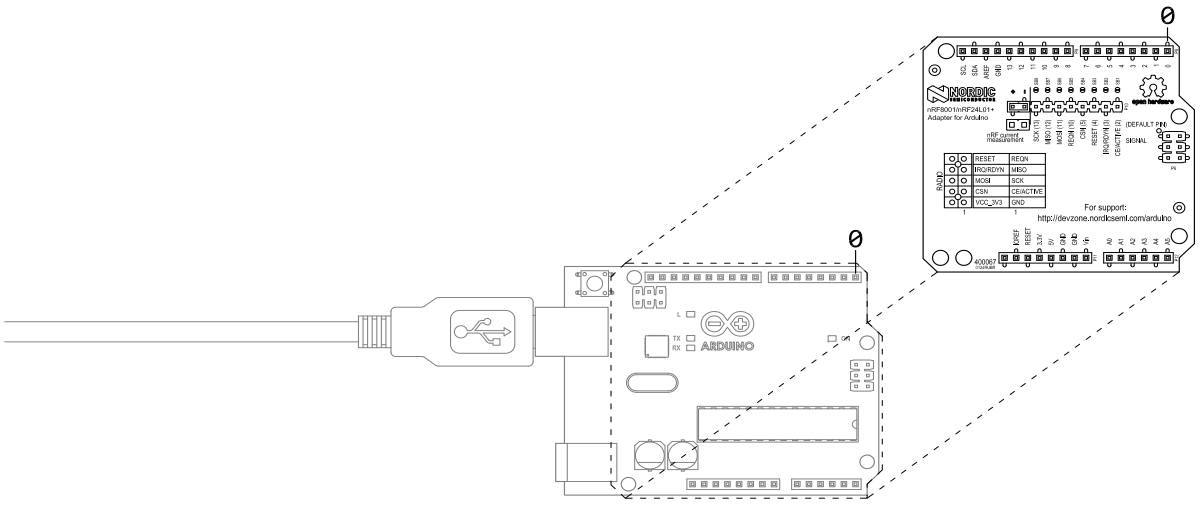
2.2.3 Schematics, Bill of Materials, PCB layout files, and production files

The ZIP file and its subdirectories contain the hardware design files for the nRF8001 DK.

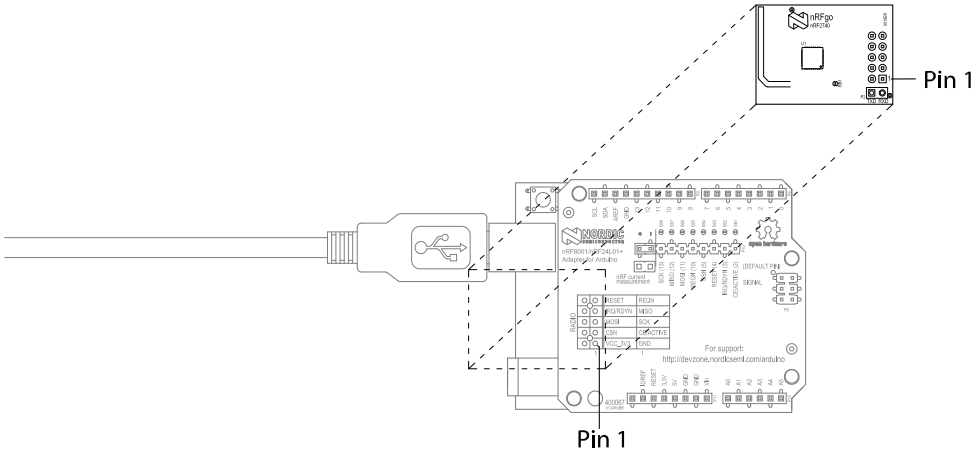
- Altium Designer files
- Schematics
- PCB layout files
- Production files
  - Assembly drawings
  - Drill files
  - Gerber files
  - Pick and Place files
  - Bill of Materials

3 Quick start

1. Connect the Arduino shield adapter to the Arduino motherboard or Arduino compatible board.



2. Connect the nRF8001 radio module to the Arduino shield adapter board.



3. Follow the instructions at http://arduino.cc/en/Guide/HomePage to download and program a test program for the Arduino board.
4. Run the test program to verify that the Arduino board has been successfully programmed and is working.
5. Download the ble_sdk_arduino SDK from https://github.com/NordicSemiconductor/ble-sdk-arduino/releases.
6. Unzip the file.
7. In the Arduino IDE, click **Sketch** and choose **Import library** and **Add library**.
8. From the **ble-sdk-arduino** folder, choose to import the libraries/BLE folder.
9. Test the installation by running the **ble_Hello_World_Program** from File/Examples/BLE.

4 Typical setup

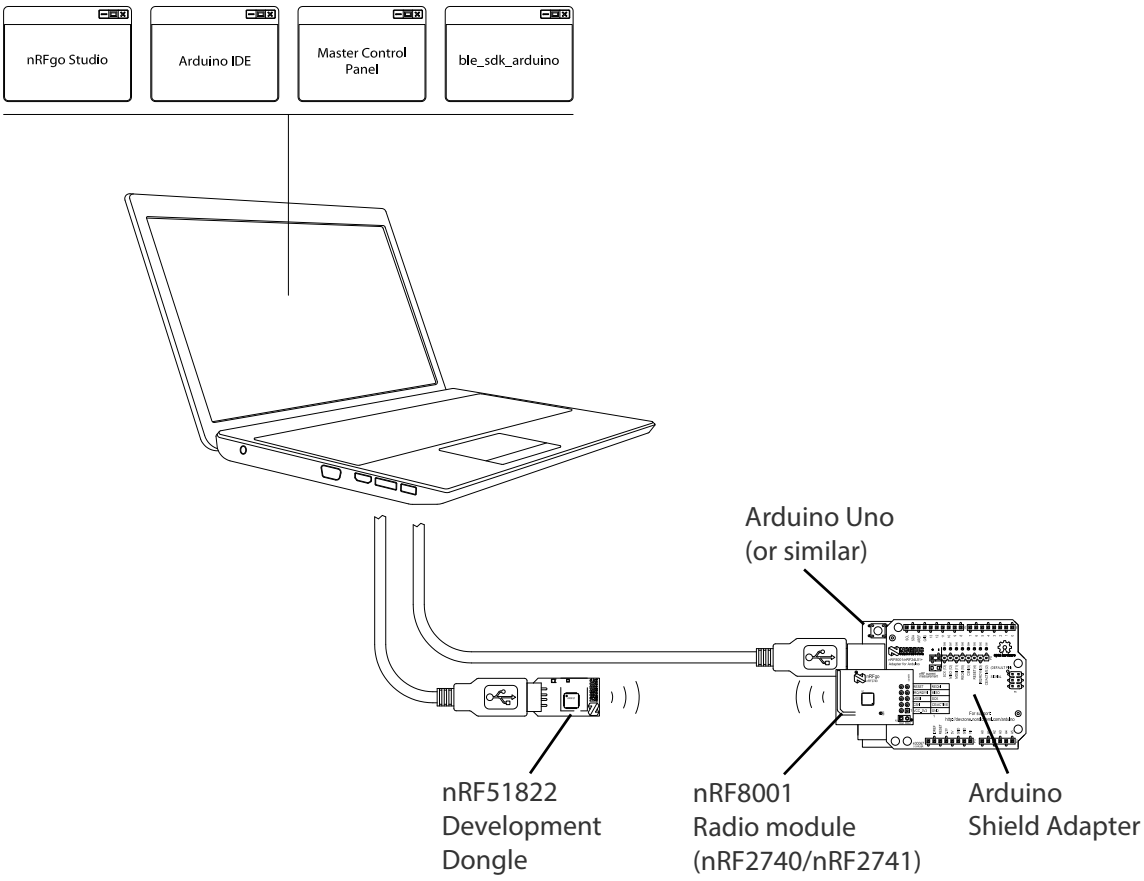


Figure 2 *Typical setup*

4.1 nRFgo Studio

Before you can start writing any application code for your *Bluetooth* low energy product you need to decide what kind of user data your product needs to transfer. For interoperability with products from other vendors you need to follow set formats specified by *Bluetooth*. These formats are defined in *Bluetooth* low energy services and profiles and decide how, for instance, a temperature sensor reports its data. From the nRF8001 configuration menu in nRFgo Studio you can choose and combine the necessary services for your product. You can also choose to use proprietary formats when your product only works with your application(s) and *Bluetooth* profile compliance is not required:

- Setting the static parameters for nRF8001:
  - **Setup of GATT services, profiles:** lets you decide which *Bluetooth* low energy profiles you want to include. The **GATT Services** view allows you to select the services you want to include in your application. You can also define your own services and characteristics.
  - **GAP setup:** lets you tune the parameters of the Generic Access Profile (GAP), for example, timing parameters and advertisement data.
  - **Hardware setup:** lets you select and configure the nRF8001 hardware features you want to use. Make the appropriate choices according to your product design.

When you have the setup you want, you can generate source files containing the setup code as #defines. The files generated by nRFgo Studio are used by the application project examples in the SDK to configure nRF8001 before use. This allows you to focus on the application code rather than spending time ensuring the *Bluetooth* setup details are correct.

The GAP and GATT settings captured on the nRFgo studio are translated to nRF8001 Setup messages. These messages are generated as #defines in the services.h and services_lock.h. Include one of these header files in your build environment (services.h places the nRF8001 Setup in the RAM of the nRF8001 and services_lock.h places the Setup in the NVRAM of the nRF8001). Once the Setup messages are in your build environment you can send them over the SPI to the nRF8001 for configuration.

You can also conduct *Bluetooth* low energy tests and program the hardware from the *Bluetooth* menu in the nRFgo Studio:

- Testing the RF PHY using Direct Test Mode.

Please see the nRFgo Studio's online help for detailed instructions.

4.2 SDK

The SDK contains source code for developing applications on the application processor (Arduino).
It includes working examples, and contains libraries that are portable to other microcontroller platforms.
The examples are written in C and can be programmed on to the Arduino using the Arduino IDE.

4.2.1 ACI modules

ACI is the interface for nRF8001. The SDK has two modules that handle the ACI communication. One handles the physical transport and the other implements the protocol.

4.2.1.1 ACI library (lib_aci)

This library implements the ACI protocol and lets you send commands, and receive events. Please refer to the nRF8001 Preliminary Product Specification for protocol details. The ACI library gives you:

- An API for ACI commands
- Decoding of ACI events

The ACI library does not have hardware dependencies, and can be used by any microcontroller with a C compiler. The ACI library requires the ACI Transport Layer, described in the following section.

4.2.1.2 ACI transport layer (hal_aci_tl)

The ACI Transport Layer module is a Hardware Abstraction Layer (HAL) for the ACI physical transport. It handles the SPI communication, and the hand-shake signals. Because of the hardware dependencies this module must be ported to your target microcontroller.

4.2.1.3 Hardware dependencies

The dependencies are related to power down mode. The code for entering power down mode is hardware dependent and must be ported when using a third-party application processor.

4.3 Master Control Panel

The Master Control Panel is used with the nRF51822 Development Dongle (PCA10000) to create a *Bluetooth* low energy master for nRF8001 and to test the application using nRF8001. With the Master Control Panel you can:

- Scan for advertisers
- Connect with nRF8001
- Send/receive data

Two additional components are installed in conjunction with the Master Control Panel:

- Visual C++ 2008 Redistributables: Files needed for the Master Control Panel, unless .Net v3.5 has already been installed
- Drivers for the master emulator (Segger)

4.4 Compiling for the Arduino platform

The source code is written in C and developed using the Arduino IDE and toolchain and Atmel studio. The Arduino IDE is available for download from http://arduino.cc/en/Main/Software.

To debug software for the application processor (Arduino) you can use simple Serial.print() function calls to print debug messages through the UART or use a JTAG based debugger that works with the Arduino board.

5 Testing the physical layer with Direct Test Mode

To test the RF performance of the physical (PHY) layer of the *Bluetooth* low energy stack in nRF8001 you can use the Direct Test Mode (DTM) interface. This interface is compliant with the description in the *Bluetooth* Specification, Version 4.0, Volume 6, Part F. It can be used for performance testing, tuning your prototypes and compliance testing.

The DTM is accessed through a dedicated UART interface on nRF8001 and through the ACI (SPI) interface. This interface is only available when the nRF8001 is in test mode. Please see the *nRF8001 Product Specification* for information on the interface and how to enter test mode.

The DTM is designed for use with *Bluetooth* test equipment. If you don't have a *Bluetooth* tester you can access the interface using this Development Kit and using nRFgo Studio to run the tests. The nRFgo Studio DTM requires that the **ble_set_in_test_mode_ACI** program must be flashed/uploaded to the Arduino board.

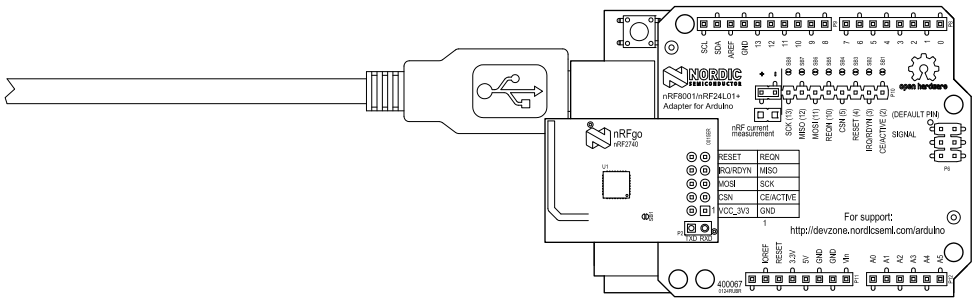


Figure 3 *Hardware setup for testing with Direct Test Mode*

5.1 Testing the physical layer on your own product

The DTM (Direct Test Mode) must be available on your product so that it can be qualified as a *Bluetooth* low energy product. This means that your application must have a way of setting the nRF8001 in test mode and have connections to the UART interface. **ble_set_in_test_mode_UART** is used to set the nRF8001 in test mode over UART.

ble_set_in_test_mode_ACI is a simple demo example showing how nRF8001 is put into the test state. From the source code you can see how the ACI command `Test` is used to set the device in test mode.

6 Hardware description

This chapter describes the:

- SCC (Single Chip Connectivity) modules (nRF2740 and nRF2741)
- Arduino shield adapter (PCA64105)
- nRF51822 Development Dongle (PCA10000)

6.1 SCC modules (nRF2740 and nRF2741)

The two SCC modules (nRF2740 and nRF2741) are identical except nRF2740 has an PCB antenna and nRF2741 has an SMA connector:

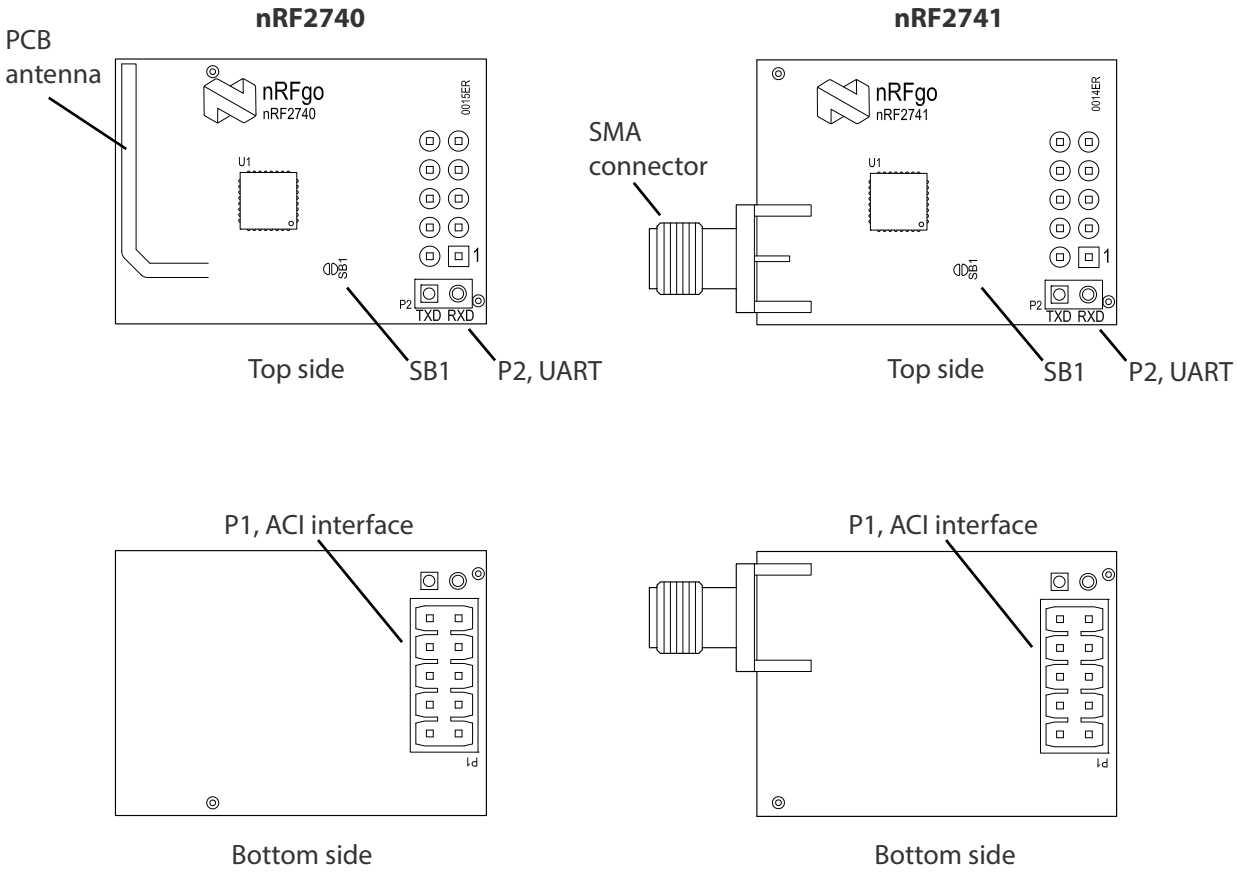


Figure 4 *nRF2740 and nRF2741 radio modules*

6.1.1 Solder bridge SB1

By default, solder bridge SB1 is shorted. Opening the SB1 solder bridge enables use of the nRF8001 internal step-down DC/DC converter. This feature is enabled from the *Bluetooth* menu in the nRFgo Studio.

6.1.2 Connector P1

For further details about signal description please see the *nRF8001 Product Specification*.

Pin number	Connector P1
1	GND
2	VCC_nRF
3	ACTIVE
4	Not in use
5	SCK
6	MOSI
7	MISO
8	RDYN
9	REQN
10	RESET

Table 1 *Pinout for connector P1*

6.1.3 Connector P2

This connector is used for the Direct Test Mode interface, which is treated in more detail in ***Chapter 5 "Testing the physical layer with Direct Test Mode"*** on page 10.

Pin number	Connector P2
1	TXD
2	RXD

Table 2 *Pinout for connector P2*

6.2 Arduino shield adapter (PCA64105)

The PCA64105 adapter shield for Arduino is designed to plug onto an Arduino or ChipKIT board (UNO32, UNO, Mega2560, Leonardo or Due), or similar. The module contains headers for the Arduino interface and a level translator circuit for the communication between a 5 V platform such as Arduino and the nRF8001 running at 3.3 V.

6.2.1 Key features

- Arduino compatible headers with pass through for stacking
- Connector for a nRF8001 development kit module
- Configurable pin mapping
- Radio current measurement

6.2.2 Hardware pictures

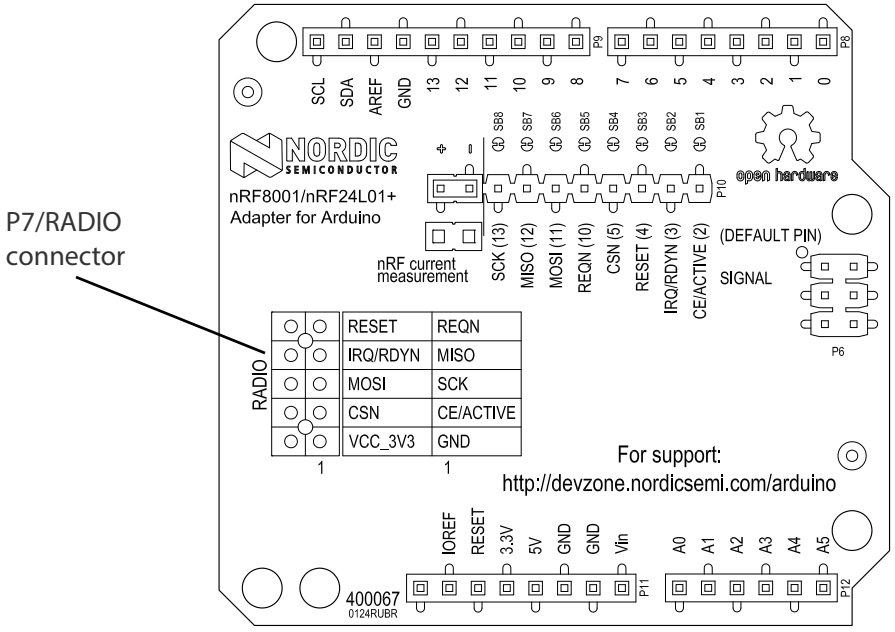


Figure 5 *Arduino shield adapter, top side*

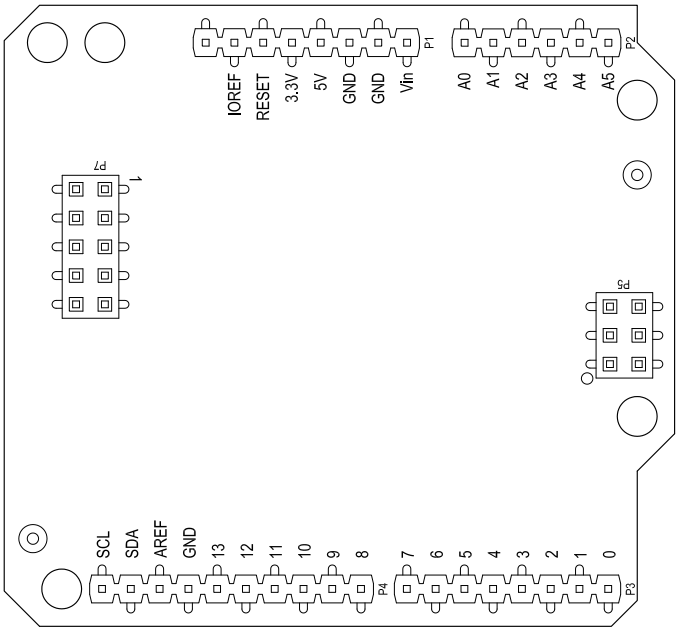


Figure 6 *Arduino shield adapter, bottom side*

6.2.3 Connector P7/RADIO

Connector P7/RADIO in ***Figure 5*** on page 13 contains the ACI interface of nRF8001 and connects the SCC module (either nRF2740 or nRF2741) to the application processor on the Arduino board.
For detailed signal descriptions please see the *nRF8001Product Specification*.

Pin number	Connector P5
1	GND
2	VCC_nRF
3	ACTIVE
4	CSN
5	SCK
6	MOSI
7	MISO
8	RDYN
9	REQN
10	RESET

Table 3 *Pinout for connector P7/RADIO*

6.2.4 Jumper nRF Current Measurement on PCA64105

This jumper supplies voltage to the connected SCC module (nRF2740/nRF2741). By replacing this jumper with an ampere meter it is possible to measure the current drawn by the nRF8001 device on the SCC module (nRF2740/nRF2741) in any operating mode.

Details on the nRF8001 static and dynamic current consumption can be found in the *nRF8001 Product Specification*.

For details on how to perform dynamic current consumption measurements, please read the white paper *RF Performance Test Guidelines*, available from www.nordicsemi.com.

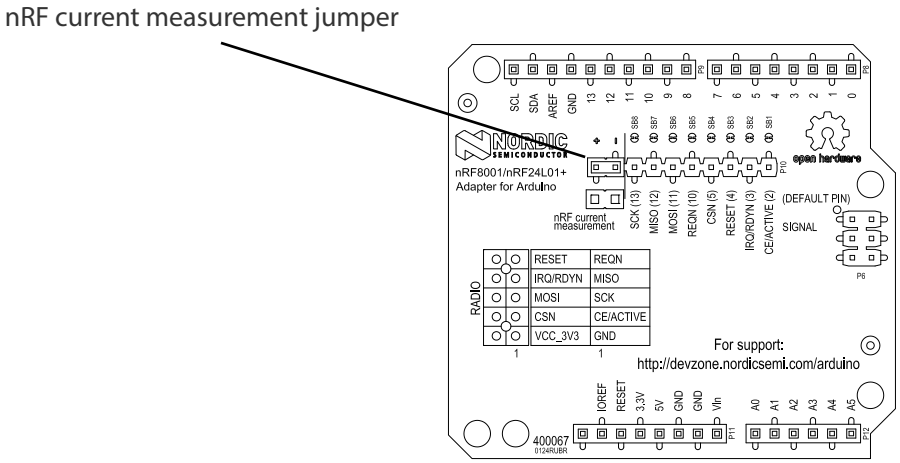


Figure 7 *nRF current measurement jumper*

6.3 nRF51822 Development Dongle (PCA10000)

The nRF51822 Development Dongle (PCA10000) can be used as a peer device or a *Bluetooth* sniffer. When PCA10000 is loaded with Master Emulator firmware and combined with the Master Control Panel, it gives you a peer device for nRF8001 that you can use to test the wireless connection.

PCA10000 can also be used as a peer device when developing a *Bluetooth* application using the nRF8001 device. It features an on-board programming and debugging solution from SEGGER. In addition to radio communication, the nRF51822 device can communicate with a computer through a virtual COM port provided by the SEGGER chip.

Note: PCA10000 can be reprogrammed if overwritten.

6.3.1 Key features

The PCA10000 has the following key features:

- nRF51822 flash based SoC solution
- *Bluetooth* low energy compatible
- 2.4 GHz compatible with nRF24L devices
- USB to UART bridge
- SEGGER J-Link OB programming and debugging capabilities

6.3.2 Hardware pictures

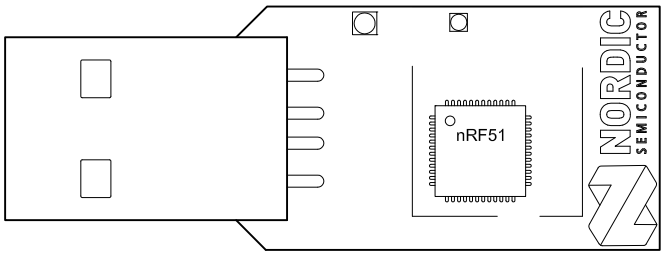


Figure 8 *PCA10000 top side*

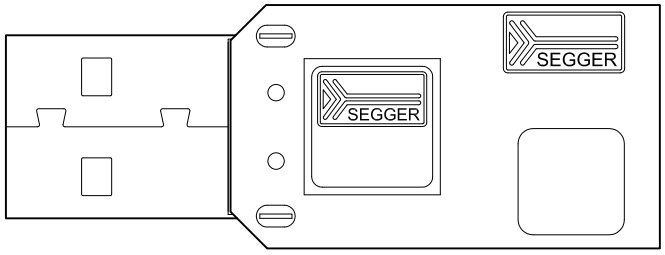


Figure 9 *PCA10000 bottom side*

6.3.3 Block diagram

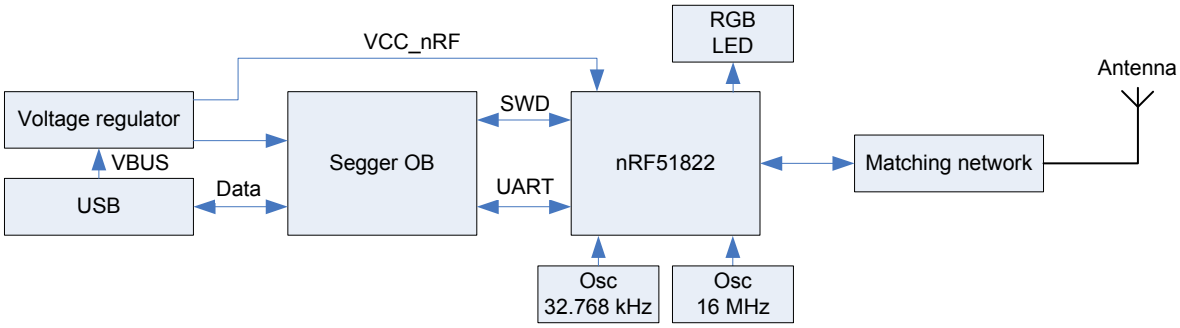


Figure 10 *PCA10000 block diagram*

6.3.4 Multicolor LED

The Development Dongle (PCA10000) is equipped with a multicolor RGB LED. The LED is connected to dedicated I/Os on the nRF51822 chip. The connections are shown in ***Table 4***.

Color	GPIO
Red	P0.21
Green	P0.22
Blue	P0.23

Table 4 *LED connection*

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

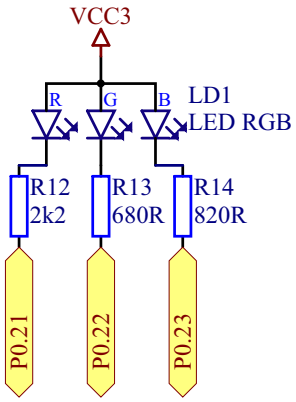


Figure 11 *LED configuration*

6.3.5 UART configuration

The Development Dongle v2.0 UART lines are connected to pins P0.08 to P0.11 as shown in ***Table 5***.

nRF51822		SEGGER IC
GPIO	**UART**	**UART**
P0.08	RTS	CTS
P0.09	TXD	RXD
P0.10	CTS	RTS
P0.11	RXD	TXD

Table 5 *nRF51822 Development Dongle UART configuration*

Note: The UART signals are routed directly to the SEGGER chip. The pins should only be used for UART. In order to use the USB to UART bridge, the software on the nRF51822 has to enable flow control. For details on how to set up the UART with flow control, see the *nRF51 Series Reference Manual*.

6.4 Schematics and PCB layouts

You can download nRF8001 Development Kit hardware schematics and PCB layout files from www.nordicsemi.com.

7 Porting the SDK

The SDK is designed for easy porting to any application controller. This chapter gives more information about how to do the porting and the minimum necessary steps.

7.1 Hardware Abstraction Layer (HAL) modules

To port the SDK to an application controller, you must re-implement the HAL modules you need for your application. However, we recommend that you do not modify the API itself, but only re-implement the functions that are already defined. This will avoid modifications to the application code.

The functionalities you need to port are:

- I/O configurations, in particular for the SPI lines, RDYN line, and REQN line.
- SPI master module: you need to configure this according to the Application Controller Interface (ACI) specification, see the nRF8001 Product Specification, chapter 7.1. for a description of this interface.

Note: Make sure that the connections are good if you are using the SCC module (nRF2740 and nRF2741) with a third party application processor development kit.

7.1.1 hal_platform.h

This file gathers all includes and register access that are specific to the hardware. This file has to be re-implemented for a new application controller using equivalent functionality on the new application controller.

7.1.2 hal_aci_tl

To communicate with nRF8001 you need to re-implement at least the hal_aci_tl (ACI transport Layer). This module implements the communication interface between the application processor and the nRF8001, the ACI transport layer. To verify your implementation, you can use the ble_aci_transport_layer_verification project which checks bi-directional data transfer and integrity of the data. See the aci_tl_demo project for more information.

A more detailed porting guide for the SDK is available at https://github.com/NordicSemiconductor/ble-sdk-arduino/blob/master/documentation/libraries/BLE/nRF8001-Porting-ACI-Library.md.

7.2 Other modules

The 'LIB' modules are designed to be as portable as possible, so you should be able to use them as is. The same is true for the 'services' modules.

8 Troubleshooting

The nRF8001 on the SCC module (nRF2740 and nRF2741) does not respond when I try to contact it. What has happened?

- Verify that the jumper **nRF current measurement** on the PCA64105 is connected.

The drop-down menu in the Master Control Panel displays no serial number. What has happened?

- Verify that the Master Control Panel software and the driver for the board have been installed and that the nRF51822 Development Dongle (PCA10000) has been plugged into a USB port on your computer.

Liability disclaimer

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

Life support applications

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

Contact details

For your nearest distributor, please visit www.nordicsemi.com.
Information regarding product updates, downloads, and technical support can be accessed through your My Page account on our home page.

Main office: Otto Nielsens veg 12
7052 Trondheim
Norway

Phone: +47 72 89 89 00
Fax: +47 72 89 89 89

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

RoHS and REACH statement

Nordic Semiconductor's products meet the requirements of Directive 2002/95/EC of the European Parliament and of the Council on the Restriction of Hazardous Substances (RoHS) and the requirements of the REACH regulation (EC 1907/2006) on Registration, Evaluation, Authorization and Restriction of Chemicals. The SVHC (Substances of Very High Concern) candidate list is continually being updated. Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website www.nordicsemi.com.

Revision History

Date	Version	Description
July 2014	2.0	• Entire User Guide has been updated to reflect the new Development Kit for Arduino.
December 2011	1.1	• Added chapter ***7 on page 19*** • Removed section 8.1.3 “Porting ACI Transport Layer to another application processor”
December 2010	1.0	

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Nordic Semiconductor:

nRF8001-DK