nRFready™ Smart Remote 2
Reference Design
nRF6924, nRF6925

User Guide v1.0
1 Introduction

This reference design is the ideal platform for developing advanced remote controls for internet-enabled TVs and set-top boxes, as well as wireless multi-touch trackpads for computers. It includes a remote control base board (nRF2790) with a multi-touch trackpad, QWERTY keyboard, low power accelerometer, 6-axis motion sensor, and an nRF51822 radio module (PCA20002).

The trackpad on the remote control base board supports both point and click and multi-touch gestures, bringing a high degree of versatility to your design. Included in the remote control base board is a full QWERTY keyboard with media control buttons providing your design with text input and classic remote control functions. The ultra-low power accelerometer detects orientation meaning the trackpad can be used both horizontally and vertically. Using the accelerometer for intelligent power saving, power consumption is significantly reduced. Legacy consumer electronics with IR receivers are supported by an IR diode on the remote control base board.

By plugging the nRF51822 radio module into the remote control base board you have the radio components to complete the remote control design. The nRF51822 radio module uses the nRF51822 System On Chip (SoC) providing a high performance, ultra-low power, and cost efficient RF remote control design.

1.1 Requirements

Demo requirements:

- nRF6920 nRFready™ 2.4 GHz RF Smart Remote or nRF6921 nRFready μBlue™ Smart Remote (only valid for the nRF6925 Smart Remote 2 Upgrade Kit)
- Bluetooth® low energy:
  - Bluetooth Smart Ready host
  - Windows 8 or Windows RT
  - Linux with Bluetooth 4.0 support (for example BlueZ 5.0 or later)
- Gazell/2.4 GHz, using supplied 2.4 GHz dongle:
  - Windows XP/7/8/RT
  - Linux with USB HID support

Development requirements:

- Windows XP/7

1.2 Writing conventions

This User Guide follows a set of typographic rules that makes the document consistent and easy to read. The following writing conventions are used:

- Commands are written in Lucida Console.
- Pin names are written in Consolas.
- File names and User Interface components are written in bold.
- Internal cross references are italicized and written in semi-bold.
2 Kit content

The nRF51822 Smart Remote Reference Design consists of hardware and access to software components, reference design files, and documentation from www.nordicsemi.com.

2.1 nRFready Smart Remote 2 Reference Design hardware content

![Figure 1 Hardware content](image)

Smart Remote base board (nRF2790)

Smart Remote nRF51822 Module (PCA20002)

nRF24LU1+ USB dongle (nRF2728)

SEGGER J-Link Lite ARM® Cortex™ JTAG/SWD emulator

10 p cable

2 x AAA Batteries

USB cable

Bluetooth 4.0 USB dongle (GBU521)
2.2 Downloadable content

The nRFready Smart Remote 2 Reference Design Kit includes firmware source code, documentation, hardware schematics, and layout files. To access these files, log in to your My Page account, enter the nRFready Smart Remote 2 product key (found on the nRFready Smart Remote 2 Reference Design Kit sleeve), and download the following files.

2.2.1 nRFready Smart Remote 2 Reference Design firmware installer file
- Firmware for nRFready Smart Remote 2
  - Precompiled HEX files
  - Source code
  - Keil ARM® project files
- S110 nRF51822 SoftDevice

2.2.2 nRFready Smart Remote 2 Reference Design documentation
- This User Guide
- Firmware documentation (will be available in a folder on you computer after you have run the installer file)

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 nRFready Smart Remote 2 Reference Design Kit.
- Altium Designer files
- PCB layout files
- Production files
  - Assembly drawings
  - Drill files
  - Gerber files
  - Pick and Place files
  - Bill of Materials
- Schematics

2.2.4 Synaptics® Pointing device driver
Trackpad driver provided by Synaptics to enable advanced multi-touch gestures.

2.2.5 Other relevant nRF51822 documentation
- nRF51 Series Reference Manual
- nRF51822 PS
- S110 nRF51822 SoftDevice Specification
- nRF51822 PAN
3 Quick start

This section shows you how to set up the nRFready Smart Remote 2 Reference Design Kit and provides example applications to help you start programming your device.

Assemble the Smart Remote 2 hardware

1. Unpack your nRFready Smart Remote 2 kit.
2. Plug the nRF51822 radio module (PCA20002) into the base board (nRF2790).

**Note:** Be careful to not use a lot of force when plugging the PCA2002 module into the base board as this could break the connector. You should be able to see the antenna on the PCA2002 after it is plugged into the base board.

3. Insert the batteries into the battery compartment on the base board (nRF2790).
3.1 Turn on and pair the Smart Remote 2

The Smart Remote 2 can be connected to a host system in two ways, by using Bluetooth Low Energy or the proprietary Gazell 2.4 GHz wireless protocol.

*Bluetooth* low energy technology is an enhancement to the *Bluetooth* Core Specification and brings Bluetooth wireless connectivity to low cost, small battery operated devices that require months to years of battery lifetime.

Gazell is a lightweight, low power, and easy-to-use protocol with advanced coexistence features especially suited for power sensitive wireless applications.

3.1.1 Pair with *Bluetooth* 4.0.

nRFready Smart Remote will only pair and work with *Bluetooth* 4.0 and HID over GATT compliant host systems. The nRFready Smart Remote 2 functionality will vary depending on the supported features in these platforms.

If your computer is not *Bluetooth* 4.0 enabled, insert the *Bluetooth* dongle into your computer and ensure the dongle is recognized and the drivers install properly.

Windows 8

The following steps are an example of the pairing process using Windows 8:

1. Slide the **SW1** switch on the base board (nRF2790) to **ON** while pressing the **BTN** button underneath the base board. The **Connection** LED now blinks orange.
2. On your computer, navigate to the device settings menu (press Windows key + W, then type ‘devices’ in the search box). Click **Add a device**.
3. When discovered, you will see Smart Remote 2 in the New Device list. Select it to begin pairing.
4. After successfully pairing, the device will show up in the connected Devices list.

5. Once pairing is successful, the **Connection** LED on the Smart Remote blinks green three times.

**Ubuntu with BlueZ 5.0**

The following steps are an example of the pairing process using Ubuntu:

1. Slide the **SW1** switch on the base board (nRF2790) to **ON** while pressing the **BTN** button underneath the base board. The **Connection** LED now blinks orange.
2. On your computer, navigate to the **Bluetooth** icon and select **Bluetooth Settings**
3. To search for a new device, click on the **+** button in the **Bluetooth** window.
4. When discovered, you will see Smart Remote 2 in the Device list. Select it and press Continue to begin pairing.

5. After successfully pairing, the device will show up in the connected Devices list.

6. Once pairing is successful, the **Connection** LED on the Smart Remote blinks green three times.

### 3.1.2 In Gazell/2.4 GHz mode

Plug the 2.4 GHz dongle into your computer. The dongle will be recognized by the computer as an nRFready Smart Remote.

1. Make sure the power switch (**SW1**) is in **OFF** position.
2. Place the remote control into a 10 to 30 cm range of the USB dongle.
3. Slide the **SW1** switch on the base board (nRF2790) to **ON** while pressing the General purpose button (**BTN**) on the underside of the base board.
4. Once pairing is successful, the **Connection** LED blink green three times.
5. Start using the remote control to control and navigate content on the host system.
3.2 Installing the Synaptics Pointing Device driver

Note: The Synaptics Pointing Device driver will only work with the 2.4 GHz USB dongle.

1. Download the Synaptics Pointing device driver from the Nordic Semiconductor website.
2. To install the driver, insert the USB dongle in your computer and start the driver installation file.
3. Follow the instructions given by the driver installer.

![Figure 2 - The driver installer](image)

4. When the driver is successfully installed, a trackpad icon becomes visible in the bottom-right corner of your screen. This icon should always be visible when the USB dongle is plugged into your computer.

![Figure 3 - Trackpad icon in a Windows taskbar](image)
5. Click the icon and select **ClickPad Properties**. This opens a gesture overview and configuration window.

![Figure 4](image_url)

*Figure 4* The gesture overview and configuration window

6. In this window you can enable and configure movement gestures, mouse cursor speed, scrolling speed and so on. Gesture demonstrations are also shown in this window.

7. A complete feature list can be found on [www.synaptics.com](http://www.synaptics.com).
4 System overview

This chapter describes the functionality of the remote control including how it can be used for development purposes.

There are three main hardware components in the reference design:

- Smart Remote base board (nRF2790)
- nRF51822 Smart Remote module (PCA20002)
- nRF24LU1+ USB dongle (nRF2728)

When the nRF51822 Smart Remote module (PCA20002) is inserted into the Smart Remote base board (nRF2790), you have a functioning remote control (see chapter 3 "Quick start" for instructions). The term remote control refers to the Smart Remote base board (nRF2790) after the nRF51822 radio module (PCA20002) is inserted.

![The remote control with a connected TV](image)

*Figure 5 The remote control with a connected TV*

All the communication in the system is digital and packet-based, which means that data between the remote control and the host/USB dongle is exchanged as discrete packets of information. The radio module checks the status of the trackpad and the keyboard matrix on the base board before sending this information to the host/USB dongle. The driver on the computer decodes the packages allowing you to use the remote control as both a pointing device and keyboard.

**Note:** The term host refers to a Bluetooth Smart Ready compliant device which supports the HID over GATT profile. Windows 8, for instance, natively supports the HID over GATT profile provided there is Bluetooth Smart Ready hardware connected to the system. Linux-based systems which incorporates version 5.0 or later of BlueZ (the official Linux Bluetooth stack) also supports the HID over GATT profile and can be used as a host. Bluetooth 4.0 dual-mode (Smart Ready) dongles may also come with driver stacks providing HID over GATT support for an operating system.
For further details on the software and firmware included, please consult the firmware documentation, code API, or the code itself.

4.1 USB dongle

The USB dongle is a stand-alone device that connects to a USB port and communicates with a USB host. It functions as a bridge between the remote control and the USB host. The USB dongle must remain connected to the USB port at all times for the remote control to work properly.

The USB dongle complies with the USB Human Interface Device (HID) specification, which has out-of-the-box support from most major operating systems (see section 4.4 on page 22 for OS support). This allows you to use the remote control without the use of additional software. However, in order to fully take advantage of its multi-touch capabilities, a Windows device driver must be installed on your computer, see section 3.2 on page 10.

4.2 Remote control

The flexibility of the remote control allows you to experiment with your own firmware and functionality. After pairing, the trackpad, accelerometer, and free space navigation can be enabled and calibrated.

4.2.1 Pairing

When the remote control is turned on, it will attempt to pair with the 2.4 GHz USB dongle. If no connection is established with the 2.4 GHz dongle, the remote control goes automatically into bond mode to pair with a Bluetooth Smart Ready compatible master.

Bluetooth low energy, HID over GATT

The Smart Remote 2 features the HID over GATT profile and can connect to any Bluetooth Smart Ready host system supporting this profile. The HID over GATT profile is a direct mapping from the USB HID standard. In addition to controlling how the HID data is transferred through the wireless link, the profile requires the use of Bluetooth device security. Security and data encryption are handled by the Bluetooth Security Manager, which is a Bluetooth Protocol Layer handled by nRF51822. On the master side of the Bluetooth Link, the Security Manager is handled in the Bluetooth driver stack.

When the remote control is turned on it starts to advertise to Smart Ready devices for pairing. This state is indicated on the board by the Connection LED blinking, see section 4.2.2.2 on page 15. When Smart Remote 2 is in this advertising state, the Bluetooth Host platform should be instructed to start scanning for and then to connect to it. During this initial connection an encrypted link is configured by nRF51822 and the host transmits HID commands to the Smart Remote 2 according to the HID over GATT profile. The Connection LED on the Smart Remote blinks to indicate that a connection has been made.

The remote control will stay in bond mode for 180 seconds waiting for a Bluetooth connection. If no connection is established, the remote control will enter deep sleep. Bonding mode will be resumed upon user activity.

Although the specifics of the pairing process may differ between platforms, the main steps remain the same:

1. Scan for the Smart Remote 2.
2. Connect to the Smart Remote 2.
3. Bond/pair with the Smart Remote 2.

Read more about Bluetooth 4.0 and HID over GATT at http://developer.bluetooth.org.
Gazell/2.4 GHz

When the remote control is first turned on (power switch SW1 set to ON) it is unpaired and cannot successfully transmit any information. By following the steps outlined in Chapter 3 on page 6 you will pair the remote control with the USB dongle.

**Note:** The Smart Remote will retain the pairing information in persistent memory when turned off. Use the pairing procedure in Chapter 3 on page 6 when you wish to establish a new pairing with a new host/protocol.

Pairing the remote control with the USB dongle ensures that data is sent to the correct USB dongle and that it does not interfere with similar devices in the area. Furthermore, during the pairing process security keys are automatically exchanged between the USB dongle and the remote control. Exchange of these keys allows encryption and secure transmission of key presses from the keyboard.

### 4.2.2 LED indicators

The remote control features two bi-color LEDs that indicate the voltage level of the battery and the connection status. The LEDs do not blink when the remote control is sleeping.

**Figure 6** The location of the LED indicators on the remote control

#### 4.2.2.1 Battery LED

When you turn on the remote control it automatically measures the battery level. The **Battery** LED indicates the voltage level of the battery:

- Blinking green (three times) indicates the battery is OK.
- Blinking orange (three times) indicates the battery is getting low.
- Blinking red (three times) indicates the battery is critically low.

After the initial battery measurement, the **Battery** LED does not blink unless it needs to indicate that the battery is low.

<table>
<thead>
<tr>
<th>Interval</th>
<th>Indication</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 2.4 V</td>
<td>Green</td>
<td>Battery is OK</td>
</tr>
<tr>
<td>2.4 to 2.0 V</td>
<td>Orange</td>
<td>Battery is getting low</td>
</tr>
<tr>
<td>&lt; 2.0 V</td>
<td>Red</td>
<td>Battery is critically low</td>
</tr>
</tbody>
</table>

**Table 1 Battery level thresholds**
**Note:** When the battery voltage is measured to be below the safe operating level for alkaline batteries (typically 0.8 V) the Smart Remote 2 will disable all functionality and enter its lowest power-state.

### 4.2.2.2 Connection LED

The **Connection** LED indicates the connection status:

- Blinking red means the remote control has lost the connection. This could be because the master is out of range or that the master has disconnected the radio link.
- Blinking orange means the remote control is in Bluetooth bond mode and ready for bonding with a Bluetooth low energy master.
- Blinking green (three times) means the remote control has successfully connected to a master device.

When the pairing process is complete, the **Connection** LED does not blink again unless indicating connection loss or that it is entering bond mode.

### 4.2.3 Trackpad

The trackpad used on the remote control is a Synaptics ClickPad ([http://www.synaptics.com](http://www.synaptics.com)). Five-finger multi-touch functionality and advanced gesture recognition make this a versatile interface device for the remote control. The trackpad is identified as a standard mouse by your computer and does not require any special software application to work.

However, in order to fully take advantage of its multi-touch capabilities, a Windows device driver must be installed on your computer, see section 3.2 on page 10. But section 4.2.3.1 on page 16 and section 4.2.3.2 on page 17 cover trackpad use regardless of whether the driver is installed or not.

**Note:** The current version of the Smart Remote will only use the driver when in 2.4 GHz mode.
4.2.3.1 Basic use

To use the trackpad, place one of your fingers on the surface of the pad as shown in Figure 7. As you move your finger along the trackpad surface, you should see the mouse cursor on your computer screen moving according to the movement of your finger.

![Figure 7 One-finger movement](image)

To perform the equivalent to a left-click on your mouse, you can either press down the left side of the trackpad until you feel a button-like click or tap the surface of the trackpad anywhere with your finger.

Various gestures with two or more fingers like described above and in section 4.2.3.2 on page 17 can be performed as well. Some of these gestures require that you install a software driver on your computer. Please see Figure 4 on page 11 for more detailed information about the different gestures. This driver can be downloaded from the Nordic Semiconductor website.
4.2.3.2 Common gestures

*Figure 8* and *Figure 9* depict trackpad gestures that can be performed regardless of whether the driver is installed or not.

*Figure 8* Two-finger horizontal scroll  
*Figure 9* Two-finger vertical scroll
4.2.4 Accelerometer

When there is no user interaction, the remote control is put into low power sleep mode. The LIS3DH 3-axis ultra-low power accelerometer from ST Microelectronics (www.st.com) is used to detect user interaction and wake up the MCU, which in turn wakes up the rest of the system. This provides intelligent power-saving (see section 4.2.6 on page 21) and auto rotation of trackpad input.

4.2.4.1 Trackpad auto rotation

When the remote control is held in a stable vertical position for a few seconds, the accelerometer will provide a notification to the MCU of the change in orientation and the remote control will automatically accommodate the trackpad to the new orientation.

To change the orientation of the trackpad, hold the remote control with the trackpad vertical for a few seconds. The trackpad axes will be re-aligned and you can use the trackpad with your preferred orientation. *Figure 10* shows the three different supported orientations.

*Figure 10* All supported orientations of the trackpad (the arrow indicates the vertical mouse cursor movement)
4.2.5 Free-space navigation

The remote control includes the MPU-6050, a powerful 3-axis gyro and 3-axis accelerometer combo circuit from InvenSense® (http://invensense.com) that can be used as a free space navigation sensor, enabling the user to move a mouse cursor through gesturing with the remote control.

The gyro and accelerometer circuit is used as an input device for a SmartMotion® firmware library from Movea® that is integrated into the Smart Remote 2 firmware. Output data from the Movea library is fitted into a HID mouse report and sent to the host.

**Note:**
- On versions of nRF2790 prior to rev. 1.0.2, the MPU-6050 is physically disconnected from the Two-Wire bus by use of solder bridges. To use the circuit, solder the solder bridges **SB1**, **SB2**, **SB3**, and **SB4** on nRF2790. See **section 5.4.4 on page 28** for more details.
- The in-air pointing functionality is only provided as precompiled HEX files. To use the Smart-Motion library in your design, please contact Movea at [http://www.movea.com](http://www.movea.com).
4.2.5.1 Basic use

To enable the free space navigation feature, the button BTN on the bottom layer of the base board must be pressed, see Figure 11. The remote control will then switch from touchpad to free space navigation mode. In this mode, acceleration and rotational data from the gyro is processed by the SmartMotion firmware library. When operating in the free space navigation mode, the Up arrow key on the keyboard will work as a left mouse button and the Backspace key will work as a right mouse button, enabling point and click functionality. To deactivate free space mode and go back to normal keyboard and touchpad mode, press the button (BTN) again.

![Figure 11 Button to enable the free space navigation feature.](image)

Movement of the remote control is translated into two dimensional mouse cursor movement. Rotation around the X axis (upward and downward movements) of the remote control will lead to vertical mouse cursor movement and rotation around the Z axis (side to side movements) will lead to horizontal mouse cursor movement. See Figure 12 as reference.

![Figure 12 Smart remote coordinate system for free space movement](image)
4.2.5.2 Trackpad functionality in free space mode

Once free space navigation mode is enabled, the trackpad functionality changes. Moving a finger on the trackpad does not cause the cursor to move, but results in a “scroll” motion (similarly to the scroll wheel on a mouse).

Vertical movement on the trackpad causes vertical scroll while horizontal movement results in horizontal scroll motion. This is illustrated in Figure 13.

Tapping or clicking the trackpad results in left-click gestures, which is not any different from the regular trackpad use.

4.2.5.3 Gyro calibration

When you start using the free space navigation you might experience that the cursor is moving involuntarily. This is due to gyro wandering, a trait inherent to this kind of motion sensor. The SmartMotion firmware library will automatically compensate for this movement. Simply leave the Smart Remote on a flat surface with free space navigation enabled. After a few seconds the cursor will stop moving and you can pick up the Smart Remote and start using the free space navigation feature.

4.2.6 Intelligent power-saving

When the remote control has not been used for a few seconds most of its functions are powered down to conserve energy, including the trackpad, which means it will not react to user input. In this powered down state the remote control relies on the accelerometer for notification of user activity. When the remote control is picked up or nudged lightly it will resume normal operation.
4.2.7 Keyboard

The remote control includes a QWERTY keyboard and a selection of media keys including volume control, play/pause, and track selection.

As the keyboard can be used to transmit sensitive data (passwords, personal information, and so on) an encrypted link between the remote control and USB dongle is used to transmit key strokes. This link is set up during the pairing process and prevents other devices from eavesdropping on your remote control.

**Note:** Media key strokes (play/pause, stop, and so on) are not transmitted through the secure link.

4.2.8 Infrared LED

To support control of legacy electronic devices that are only fitted with infrared (IR) remote control receivers, the remote control is fitted with an IR LED. The IR LED and IR protocols are handled by the MCU.

**Note:** There is no IR LED functionality implemented in the Smart Remote 2 firmware.

![Figure 14 Location of the infrared LED on the remote control](image)

4.3 Firmware upgrading

Both the Smart Remote 2 and the USB dongle firmware can be freely modified and upgraded. See Chapter 8 on page 44 for how to upgrade the firmware for the Smart Remote 2. Firmware upgrading for the USB dongle requires the nRF24LU1+ Development Kit and the upgrading process is described in its documentation.

4.4 Supported operating systems

**Bluetooth low energy:**

The nRFready Smart Remote 2 complies with the Bluetooth 4.0 HID over GATT profile and is compatible with all host systems incorporating this profile.

**Gazell/2.4 GHz, using supplied 2.4 GHz dongle:**

The 2.4 GHz USB dongle complies with the USB HID specification which is supported by Windows, Linux, Apple's OS X and any system that supports USB HID. However, to utilize the Synaptics driver, special requirements may apply. Please refer the Synaptics driver for supported operating systems.
5 nRF2790 Smart Remote base board hardware description

The base board contains all the hardware necessary for user interaction, including batteries. It has a trackpad and a QWERTY keyboard matrix as main input devices and there are two bi-color LEDs for feedback. The base board has an infrared LED, a low power accelerometer, and a gyro/accelerometer combo chip so that you can implement customized features. Use the 40 pin connector on the bottom side of the base board to connect to an nRFready Smart Remote radio module.

5.1 Key features

The base board has the following key features:

- Full QWERTY keyboard matrix with media keys
- Multi-touch trackpad (supports touch of up to five fingers)
- Gyro and accelerometers
- Infrared LED for legacy support
- Bi-color LEDs for user feedback
- DC-DC boost regulator to ensure a stable voltage and extend battery life time
- Powered by two AAA batteries, alkaline or NiMH
- Supports plugin modules for Nordic Semiconductor radios

5.2 Hardware figures

![Figure 15 nRF2790 top view](image-url)
5.3 Block diagram

Figure 16 nRF2790 bottom view

Figure 17 nRF2790 block diagram
5.4 Design description
This chapter contains details about the hardware blocks on the base board.

5.4.1 Trackpad
The trackpad is mounted onto the nRF2790 PCB and connected to the P2 connector, see Figure 18.

![Figure 18 Trackpad interface connector](image)

The trackpad is interfaced through the two-wire bus interface. See Table 2 for the pin description.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NC</td>
<td>Not Connected</td>
</tr>
<tr>
<td>2</td>
<td>WU_IRQ_0</td>
<td>Interrupt from trackpad</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>4</td>
<td>SCL</td>
<td>Two-wire serial clock</td>
</tr>
<tr>
<td>5</td>
<td>SDA</td>
<td>Two-wire serial data</td>
</tr>
<tr>
<td>6</td>
<td>VCC_PAD</td>
<td>Power supply for trackpad</td>
</tr>
</tbody>
</table>

*Table 2 Trackpad pin description*
5.4.2 Keyboard matrix

The keyboard on the nRF2790 base board is a 60 button matrix from Raytac. The matrix has 8 rows and 8 columns using 16 I/Os on the MCU, see Figure 19.

The 8 x 8 matrix gives room for 64 buttons in firmware, leaving 4 open spaces in the table. One space is used for the button (BTN) on the bottom layer of the base board. See Figure 16 on page 24. All the rows of the matrix are connected to a diode array giving a logical OR of all the lines. This signal is connected to a wake up interrupt line going to the MCU.
5.4.3 Accelerometer circuit

To obtain low power consumption and long battery lifetime, a low power 3-axis accelerometer (U3) has been added to the base board, see Figure 20. See also section 4.2.6 on page 21, Intelligent power-saving.

The accelerometer has digital SPI outputs and can detect motion on 3 axes. The sensitivity is configurable to \( \pm 2 \text{ g/} \pm 4 \text{ g/} \pm 8 \text{ g/} \pm 16 \text{ g} \).
5.4.4 Gyro/Accelerometer circuit

For advanced features, the base board has a 3-axis gyro integrated with a 3-axis accelerometer (U2), see Figure 21.

Figure 21 Gyro/accelerometer circuit

To connect the circuit to the MCU, four solder bridges must be soldered (SB1-SB4). On versions of the nRF2790 prior to v1.0.2 the circuit has been disconnected by default so that you can choose whether to enable this chip. This circuit is a ±250 dps/±500 dps/±1000 dps/±2000 dps selectable 3-axis gyro and a ±2 g/±4 g/±8 g/±16 g selectable 3-axis accelerometer. The chip has digital outputs and is interfaced through two-wire communication.

The MPU-6050 has an auxiliary two-wire master built in that could be used to interface a digital magnetometer/compass. This interface is available on the P5 connector on the nRF2790.
5.4.5 Power supply

The remote control is powered by two AAA batteries. These can be alkaline (2 x 1.5 V) or rechargeable NiMH (2 x 1.2 V) batteries. The voltage from the batteries is boosted to a level suitable for the devices on the remote control, see Figure 22.

![Figure 22 Voltage regulator circuit](image)

VCC is regulated to 3.276 V with the adjustable voltage regulator U1. The equation for VCC is shown below:

\[ V_{out} = 0.45 \times \frac{(R_1 + R_2)}{R_2} \Rightarrow VCC = 0.45 \times \frac{(270k\Omega + 43k\Omega)}{43k\Omega} = 3.276V \]

The supply voltage used in this circuitry is set to 3.276 V which is required by the trackpad.

**Note:** Since this is a boost only converter, the battery voltage should never exceed the output voltage of 3.276 V.

5.4.6 Infrared LED and driver

To support legacy products the base board has an infrared LED with a driver circuit. The IR LED is driven by a transistor (Q1) to offer higher current than the MCU I/O can offer.

![Figure 23 The infrared LED and driver circuit](image)

The control signal is active high meaning that when the IR_LED signal has logic high level, the LED emits infrared light.
5.4.7 LEDs and drivers

There are two bi-color LEDs on the base board. These LEDs are driven by transistors (Q2 and Q3) to offer higher current than the MCU I/Os can offer.

![LED drivers circuit](image1)

The control signals to the transistors are active low. The LEDs are able to emit three different colors, LED LD1 and LED LD2 have the same configuration, and the truth table is described in Table 3.

<table>
<thead>
<tr>
<th>LED_Gx</th>
<th>LED_Rx</th>
<th>LDx color</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>High</td>
<td>Off</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>Green</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>Red</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>Orange/Yellow</td>
</tr>
</tbody>
</table>

Table 3 Truth table for LED LD1 and LED LD2

5.4.8 Module interface

A 40 pin PCB edge connector is used to connect a radio module to the base board. Through this connector all the functionality of the base board can be accessed by a radio module. See Table 4 for pin information.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KEYIN_0</td>
<td>Input to the MCU from keyboard matrix row 0</td>
</tr>
<tr>
<td>2</td>
<td>KEYIN_1</td>
<td>Input to the MCU from keyboard matrix row 1</td>
</tr>
<tr>
<td>3</td>
<td>KEYIN_2</td>
<td>Input to the MCU from keyboard matrix row 2</td>
</tr>
<tr>
<td>4</td>
<td>KEYIN_3</td>
<td>Input to the MCU from keyboard matrix row 3</td>
</tr>
<tr>
<td>5</td>
<td>KEYIN_4</td>
<td>Input to the MCU from keyboard matrix row 4</td>
</tr>
<tr>
<td>6</td>
<td>KEYIN_5</td>
<td>Input to the MCU from keyboard matrix row 5</td>
</tr>
<tr>
<td>7</td>
<td>KEYIN_6</td>
<td>Input to the MCU from keyboard matrix row 6</td>
</tr>
<tr>
<td>8</td>
<td>KEYIN_7</td>
<td>Input to the MCU from keyboard matrix row 7</td>
</tr>
<tr>
<td>9</td>
<td>KEYOUT_1</td>
<td>Output from the MCU to keyboard matrix column 1</td>
</tr>
<tr>
<td>10</td>
<td>KEYOUT_2</td>
<td>Output from the MCU to keyboard matrix column 2</td>
</tr>
<tr>
<td>11</td>
<td>KEYOUT_3</td>
<td>Output from the MCU to keyboard matrix column 3</td>
</tr>
<tr>
<td>Pin</td>
<td>Label</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>12</td>
<td>KEYOUT_4</td>
<td>Output from the MCU to keyboard matrix column 4</td>
</tr>
<tr>
<td>13</td>
<td>KEYOUT_5</td>
<td>Output from the MCU to keyboard matrix column 5</td>
</tr>
<tr>
<td>14</td>
<td>KEYOUT_6</td>
<td>Output from the MCU to keyboard matrix column 6</td>
</tr>
<tr>
<td>15</td>
<td>KEYOUT_7</td>
<td>Output from the MCU to keyboard matrix column 7</td>
</tr>
<tr>
<td>16</td>
<td>KEYOUT_8</td>
<td>Output from the MCU to keyboard matrix column 8</td>
</tr>
<tr>
<td>17</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>18</td>
<td>SDA</td>
<td>Two-wire serial data</td>
</tr>
<tr>
<td>19</td>
<td>SCL</td>
<td>Two-wire serial clock</td>
</tr>
<tr>
<td>20</td>
<td>SPI_SCK</td>
<td>SPI serial clock</td>
</tr>
<tr>
<td>21</td>
<td>SPI_MOSI</td>
<td>SPI serial master output data</td>
</tr>
<tr>
<td>22</td>
<td>SPI_MISO</td>
<td>SPI serial master input data</td>
</tr>
<tr>
<td>23</td>
<td>CSN_1</td>
<td>SPI serial slave select 1</td>
</tr>
<tr>
<td>24</td>
<td>CSN_0</td>
<td>SPI serial slave select 0</td>
</tr>
<tr>
<td>25</td>
<td>GP_IRQ_1</td>
<td>General purpose interrupt 1</td>
</tr>
<tr>
<td>26</td>
<td>GP_IRQ_0</td>
<td>General purpose interrupt 0</td>
</tr>
<tr>
<td>27</td>
<td>WU_IRQ_2</td>
<td>Wake-up interrupt 2</td>
</tr>
<tr>
<td>28</td>
<td>WU_IRQ_1</td>
<td>Wake-up interrupt 1</td>
</tr>
<tr>
<td>29</td>
<td>WU_IRQ_0</td>
<td>Wake-up interrupt 0</td>
</tr>
<tr>
<td>30</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>31</td>
<td>NC</td>
<td>No connection</td>
</tr>
<tr>
<td>32</td>
<td>IR_LED</td>
<td>Infrared LED control signal input</td>
</tr>
<tr>
<td>33</td>
<td>LED_G2</td>
<td>Battery LED green color control signal input</td>
</tr>
<tr>
<td>34</td>
<td>LED_R2</td>
<td>Battery LED red color control signal input</td>
</tr>
<tr>
<td>35</td>
<td>LED_G1</td>
<td>Connection LED green color control signal input</td>
</tr>
<tr>
<td>36</td>
<td>LED_R1</td>
<td>Connection LED red color control signal input</td>
</tr>
<tr>
<td>37</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>38</td>
<td>VBAT</td>
<td>Battery voltage</td>
</tr>
<tr>
<td>39</td>
<td>VCC_MOD</td>
<td>Power supply for module</td>
</tr>
<tr>
<td>40</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>

*Table 4* Module connector pin information
5.4.9 Current measurement

On the base board there are test pads available for current measurement of individual components in the system. By default, 0 Ω resistors are mounted shorting the line between the test points. If performing current measurement on a specific unit in the system, remove the 0 Ω resistor for that part to measure the current between the two test pads. See Table 5 for information on resistors and test pads.

![Current measurement circuit](image)

**Figure 25** Current measurement circuit

<table>
<thead>
<tr>
<th>Unit</th>
<th>Resistor</th>
<th>Test pad supply side</th>
<th>Test pad device side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio module</td>
<td>R14</td>
<td>TP7</td>
<td>TP1</td>
</tr>
<tr>
<td>Infrared</td>
<td>R15</td>
<td>TP8</td>
<td>TP2</td>
</tr>
<tr>
<td>LEDs</td>
<td>R16</td>
<td>TP9</td>
<td>TP3</td>
</tr>
<tr>
<td>LIS3DH Accelerometer</td>
<td>R17</td>
<td>TP10</td>
<td>TP4</td>
</tr>
<tr>
<td>Trackpad</td>
<td>R18</td>
<td>TP11</td>
<td>TP5</td>
</tr>
<tr>
<td>MPU-6050 Gyro/Accelerometer</td>
<td>R19</td>
<td>TP12</td>
<td>TP6</td>
</tr>
</tbody>
</table>

**Table 5** Current measurement test points

5.5 Schematics, Bill of Materials, PCB layout files, production files

The hardware files for the base board will be located in the following folder in the hardware files zip package: \nRFready Smart Remote 2 x_x_x\Hardware files\nRF2790-Smart Remote Base Board x_x_x. In this folder you can find Bill of Materials, schematics and pcb-layout files in pdf-format, Altium Designer files, and production files (assembly drawings, gerber files, drill files, pick & place files).
6 PCA20002 nRF51822 radio module hardware description

The PCA20002 radio module is designed to plug into the P1 connector on the nRF2790 Smart Remote base board. The module contains an nRF51822 multiprotocol Bluetooth low energy and 2.4 GHz proprietary SoC from Nordic Semiconductor and includes a programming interface for firmware upgrade and development. The module has a header that can be used for current measurements of the nRF51822. Headers for the UART interface are available. The UART interface can be used to print debugging outputs. The module has a 40 pin PCB edge connector to ensure good electrical connection and mechanical stability when connected to the base board.

6.1 Key features

The PCA20002 radio module has the following key features:

- nRF51822 flash based SoC solution
- Bluetooth low energy compatible
- 2.4 GHz compatible with nRF24L devices
- SWD interface connector for programming and debugging
- UART interface
- Current measurement header
- Interface to the Smart Remote base board

6.2 Hardware figures

![PCA20002 radio module top view](Figure 26)

![PCA20002 radio module bottom view](Figure 27)
6.3 Block diagram

Figure 28 PCA20002 radio module block diagram

6.4 Design description

This chapter contains information about the hardware block on the radio module.

6.4.1 SWD interface and firmware upgrade/debugging

A connector for the ARM SWD interface is included on the module for easy firmware upgrade purposes. When this interface is connected to a compatible programmer, firmware upgrades can be made directly on the module. The interface is found on the 10 pin connector P1, see Figure 29.

Chapter 8 on page 44 describes the programming adapter used for firmware upgrades and debugging of the nRF51822 module.

Note: Make sure that pin 1 on the PCA20002 P1 connector is connected to pin 1 of the connector on the programmer unit.

Figure 29 Location of SWD interface connector P1
6.4.2 I/O usage

The nRF51822-QFAA has 31 generic I/Os available. All I/Os are used in this design and are organized as shown in Table 7.

### Table 6 nRF51822 radio module SWD interface connector P1 pin description

<table>
<thead>
<tr>
<th>Pin</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCC</td>
<td>Reference voltage for programmer</td>
</tr>
<tr>
<td>2</td>
<td>SWDIO</td>
<td>Serial wire debug data</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>4</td>
<td>SWCLK</td>
<td>Serial wire debug clock</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>6</td>
<td>NC</td>
<td>No connection</td>
</tr>
<tr>
<td>7</td>
<td>NC</td>
<td>Not present</td>
</tr>
<tr>
<td>8</td>
<td>NC</td>
<td>No connection</td>
</tr>
<tr>
<td>9</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>10</td>
<td>RESET</td>
<td>MCU reset</td>
</tr>
</tbody>
</table>

### Table 7 nRF51822-QFAA general purpose I/Os

<table>
<thead>
<tr>
<th>I/O</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0.00</td>
<td>KEYOUT_3</td>
<td>Output to keyboard column 3</td>
</tr>
<tr>
<td>P0.01</td>
<td>KEYOUT_4</td>
<td>Output to keyboard column 4</td>
</tr>
<tr>
<td>P0.02</td>
<td>KEYOUT_5</td>
<td>Output to keyboard column 5</td>
</tr>
<tr>
<td>P0.03</td>
<td>KEYOUT_6</td>
<td>Output to keyboard column 6</td>
</tr>
<tr>
<td>P0.04</td>
<td>KEYOUT_7</td>
<td>Output to keyboard column 7</td>
</tr>
<tr>
<td>P0.05</td>
<td>KEYOUT_8</td>
<td>Output to keyboard column 8</td>
</tr>
<tr>
<td>P0.06</td>
<td>BAT_SENSE</td>
<td>Battery monitoring input signal</td>
</tr>
<tr>
<td>P0.07</td>
<td>SDA</td>
<td>Two-wire master data</td>
</tr>
<tr>
<td>P0.08</td>
<td>SCL</td>
<td>Two-wire master clock</td>
</tr>
<tr>
<td>P0.09</td>
<td>SPI_SCK</td>
<td>SPI master clock</td>
</tr>
<tr>
<td>P0.10</td>
<td>SPI_MOSI</td>
<td>SPI master output data</td>
</tr>
<tr>
<td>P0.11</td>
<td>SPI_MISO</td>
<td>SPI master input data</td>
</tr>
<tr>
<td>P0.12</td>
<td>CSN_1</td>
<td>SPI master slave select 1</td>
</tr>
<tr>
<td>P0.13</td>
<td>CSN_0</td>
<td>SPI master slave select 0</td>
</tr>
<tr>
<td>P0.14</td>
<td>GP_IRQ_1</td>
<td>General purpose interrupt 1</td>
</tr>
<tr>
<td>P0.15</td>
<td>GP_IRQ_0</td>
<td>General purpose interrupt 0</td>
</tr>
<tr>
<td>P0.16</td>
<td>WU_IRQ_2</td>
<td>Wake-up interrupt 2</td>
</tr>
<tr>
<td>P0.17</td>
<td>WU_IRQ_1</td>
<td>Wake-up interrupt 1</td>
</tr>
<tr>
<td>P0.18</td>
<td>WU_IRQ_0</td>
<td>Wake-up interrupt 0</td>
</tr>
<tr>
<td>P0.19</td>
<td>IR_LED/RXD</td>
<td>PWM output/UART Receive</td>
</tr>
<tr>
<td>P0.20</td>
<td>TXD</td>
<td>UART transmit</td>
</tr>
<tr>
<td>P0.21</td>
<td>KEYIN_0</td>
<td>Input from keyboard row 0</td>
</tr>
<tr>
<td>P0.22</td>
<td>KEYIN_1</td>
<td>Input from keyboard row 1</td>
</tr>
</tbody>
</table>
6.4.3 I/O extension

The nRF51822 module is equipped with an two-wire I/O extender that is used for driving the LEDs on the base board. The I/O extender is chosen to get a cleaner design and to demonstrate multiple features without having to combine I/Os for different purposes. The I/O extender is a PCA9557 from NXP Semiconductors and the two-wire address of the I/O extender is 0x18. See Figure 30 and Table 8 for reference on the connections.

**Table 7 I/O usage**

<table>
<thead>
<tr>
<th>I/O</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0.23</td>
<td>KEYIN_2</td>
<td>Input from keyboard row 2</td>
</tr>
<tr>
<td>P0.24</td>
<td>KEYIN_3</td>
<td>Input from keyboard row 3</td>
</tr>
<tr>
<td>P0.25</td>
<td>KEYIN_4</td>
<td>Input from keyboard row 4</td>
</tr>
<tr>
<td>P0.26</td>
<td>KEYIN_5</td>
<td>Input from keyboard row 5</td>
</tr>
<tr>
<td>P0.27</td>
<td>KEYIN_6</td>
<td>Input from keyboard row 6</td>
</tr>
<tr>
<td>P0.28</td>
<td>KEYIN_7</td>
<td>Input from keyboard row 7</td>
</tr>
<tr>
<td>P0.29</td>
<td>KEYOUT_1</td>
<td>Output to keyboard column 1</td>
</tr>
<tr>
<td>P0.30</td>
<td>KEYOUT_2</td>
<td>Output to keyboard column 2</td>
</tr>
</tbody>
</table>

**Table 8 I/O configuration of the I/O extender**

<table>
<thead>
<tr>
<th>I/O</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO0</td>
<td>NC</td>
<td>No connection</td>
</tr>
<tr>
<td>IO1</td>
<td>NC</td>
<td>No connection</td>
</tr>
<tr>
<td>IO2</td>
<td>NC</td>
<td>No connection</td>
</tr>
<tr>
<td>IO3</td>
<td>NC</td>
<td>No connection</td>
</tr>
<tr>
<td>IO4</td>
<td>LED_G2</td>
<td>LD2, green LED driver signal output</td>
</tr>
<tr>
<td>IO5</td>
<td>LED_R2</td>
<td>LD2, red LED driver signal output</td>
</tr>
<tr>
<td>IO6</td>
<td>LED_G1</td>
<td>LD1, green LED driver signal output</td>
</tr>
<tr>
<td>IO7</td>
<td>LED_R1</td>
<td>LD1, red LED driver signal output</td>
</tr>
</tbody>
</table>
Note: The unused I/Os of the I/O extender should be set as output at a fixed level to avoid floating pins.

6.4.4 Battery monitoring

To measure and monitor the remote control battery voltage level, a resistor network is applied as shown in Figure 31.

![Figure 31 Resistor network for battery monitoring](image)

This network gives a scale factor of: 

$$ R_{\text{div}} = \frac{R_3}{R_2 + R_3} = \frac{560}{1560} = 0.359 $$

The internal band gap voltage reference (nominally 1.2 V) is selected during battery measurement. This means that the maximum input voltage from the resistor network above should be equal or less than 1.2 V for full range. The maximum battery voltage that could be measured is calculated according to:

$$ V_{\text{bat(max)}} \leq \frac{V_{\text{ref}}}{R_{\text{div}}} \Rightarrow V_{\text{ref}} \times \frac{R_2 + R_3}{R_3} = 3.34 \text{V} $$

6.4.5 Current measurement

A circuit for measuring the current consumption of nRF51822 has been added to the design. By default, a 0 Ω resistor is mounted, shorting the VCC and VCC_nRF nets. By removing R4 from the PCB, an ampere meter can be connected to the pins of JP1 for current measurement of the nRF51822, see Figure 32.

![Figure 32 Current measurement circuit](image)
6.4.6 Base board interface

To connect to the base board, the module has a 40 pin PCB edge connector. All power and I/O lines needed between the base board and the module go through this connector. The connector has VCC line for power supply and VBAT (battery voltage) for battery measurement. It has two-wire and SPI lines, interrupt lines, PWM for IR, outputs for status LEDs, and 16 I/Os for the keyboard matrix. See Table 9 for reference and pin description.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KEYIN_0</td>
<td>Input to the MCU from keyboard matrix row 0</td>
</tr>
<tr>
<td>2</td>
<td>KEYIN_1</td>
<td>Input to the MCU from keyboard matrix row 1</td>
</tr>
<tr>
<td>3</td>
<td>KEYIN_2</td>
<td>Input to the MCU from keyboard matrix row 2</td>
</tr>
<tr>
<td>4</td>
<td>KEYIN_3</td>
<td>Input to the MCU from keyboard matrix row 3</td>
</tr>
<tr>
<td>5</td>
<td>KEYIN_4</td>
<td>Input to the MCU from keyboard matrix row 4</td>
</tr>
<tr>
<td>6</td>
<td>KEYIN_5</td>
<td>Input to the MCU from keyboard matrix row 5</td>
</tr>
<tr>
<td>7</td>
<td>KEYIN_6</td>
<td>Input to the MCU from keyboard matrix row 6</td>
</tr>
<tr>
<td>8</td>
<td>KEYIN_7</td>
<td>Input to the MCU from keyboard matrix row 7</td>
</tr>
<tr>
<td>9</td>
<td>KEYOUT_1</td>
<td>Output from the MCU to keyboard matrix column 1</td>
</tr>
<tr>
<td>10</td>
<td>KEYOUT_2</td>
<td>Output from the MCU to keyboard matrix column 2</td>
</tr>
<tr>
<td>11</td>
<td>KEYOUT_3</td>
<td>Output from the MCU to keyboard matrix column 3</td>
</tr>
<tr>
<td>12</td>
<td>KEYOUT_4</td>
<td>Output from the MCU to keyboard matrix column 4</td>
</tr>
<tr>
<td>13</td>
<td>KEYOUT_5</td>
<td>Output from the MCU to keyboard matrix column 5</td>
</tr>
<tr>
<td>14</td>
<td>KEYOUT_6</td>
<td>Output from the MCU to keyboard matrix column 6</td>
</tr>
<tr>
<td>15</td>
<td>KEYOUT_7</td>
<td>Output from the MCU to keyboard matrix column 7</td>
</tr>
<tr>
<td>16</td>
<td>KEYOUT_8</td>
<td>Output from the MCU to keyboard matrix column 8</td>
</tr>
<tr>
<td>17</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>18</td>
<td>SDA</td>
<td>Two-wire serial data</td>
</tr>
<tr>
<td>19</td>
<td>SCL</td>
<td>Two-wire serial clock</td>
</tr>
<tr>
<td>20</td>
<td>SPI_SCK</td>
<td>SPI serial clock</td>
</tr>
<tr>
<td>21</td>
<td>SPI_MOSI</td>
<td>SPI serial master output data</td>
</tr>
<tr>
<td>22</td>
<td>SPI_MISO</td>
<td>SPI serial master input data</td>
</tr>
<tr>
<td>23</td>
<td>CSN_1</td>
<td>SPI serial slave select 1</td>
</tr>
<tr>
<td>24</td>
<td>CSN_0</td>
<td>SPI serial slave select 0</td>
</tr>
<tr>
<td>25</td>
<td>GP_IRQ_1</td>
<td>General purpose interrupt 1</td>
</tr>
<tr>
<td>26</td>
<td>GP_IRQ_0</td>
<td>General purpose interrupt 0</td>
</tr>
<tr>
<td>27</td>
<td>WU_IRQ_2</td>
<td>Wake-up interrupt 2</td>
</tr>
<tr>
<td>28</td>
<td>WU_IRQ_1</td>
<td>Wake-up interrupt 1</td>
</tr>
<tr>
<td>29</td>
<td>WU_IRQ_0</td>
<td>Wake-up interrupt 0</td>
</tr>
<tr>
<td>30</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>31</td>
<td>NC</td>
<td>No connection</td>
</tr>
<tr>
<td>32</td>
<td>IR_LED</td>
<td>Infrared LED control signal input</td>
</tr>
<tr>
<td>33</td>
<td>LED_G2</td>
<td>Battery LED green color control signal input</td>
</tr>
<tr>
<td>34</td>
<td>LED_R2</td>
<td>Battery LED red color control signal input</td>
</tr>
<tr>
<td>Pin</td>
<td>Label</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>35</td>
<td>LED_G1</td>
<td>Connection LED green color control signal input</td>
</tr>
<tr>
<td>36</td>
<td>LED_R1</td>
<td>Connection LED red color control signal input</td>
</tr>
<tr>
<td>37</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>38</td>
<td>VBAT</td>
<td>Battery voltage</td>
</tr>
<tr>
<td>39</td>
<td>VCC_MOD</td>
<td>Power supply for module</td>
</tr>
<tr>
<td>40</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>

*Table 9 Base board connector pin information*
6.4.7  Matching network

The layout of the matching network, component size, and component values are exactly as given in the application example found in the *nRF51822 Product Specification*. This reference matching network layout is also available for download from the Nordic Semiconductor website.

6.4.8  Antenna

The antenna is a quarter wavelength monopole antenna printed on the PCB. The antenna is tuned to be resonant at 2.44 GHz, and the impedance is matched to the 50 Ω output of the matching network with the use of a shunt capacitor to ground, C13 = 1.0 pF. The antenna in this design is tuned for this layout only, and with the module connected to the base board.

If the layout is changed and/or the base board is changed, it is likely the antenna must be re-tuned. The antenna can be re-tuned by adjusting its length and/or tuning the value of C13.

The exact length of the antenna and the value of C13 must be determined by measurements with a vector network analyzer.

6.5  Schematics, Bill of Materials, PCB layout files, production files

The hardware files for the radio module will be located in the following folder in the hardware files zip package: \nRFready Smart Remote 2 x_x_x\Hardware files\PCA20002-nRF51822 Smart Remote Module x_x_x. In this folder you can find Bill of Materials, schematics, and pcb-layout files in pdf-format, Altium Designer files, and production files (assembly drawings, gerber files, drill files, pick & place files).
7 nRF2728 USB dongle hardware description

The nRF2728 USB dongle is a receiver with a USB interface. This single-chip miniature USB dongle uses the nRF24LU1+ SoC solution from Nordic Semiconductor.

7.1 Key features

The nRF2728 USB dongle has the following key features:

- nRF24LU1+ 2.4 GHz RF flash based SoC with USB
- Compact form factor
  - 12 x 23 mm without USB connector
  - 12 x 32 mm with USB connector
- Meander type antenna
- USB v2.0 compatible
- ETSI 300-440, FCC CFR 47 paragraph 15.247 and Industry Canada RSS-210 Issue 7 precompliant
- Supports firmware development directly on the dongle

7.2 Hardware figures

![Figure 33 nRF2728 top view](image)

![Figure 34 nRF2728 bottom view](image)
7.3 Block diagram

![Block Diagram](image_url)

Figure 35 nRF2728 nRF24LU1+ USB dongle functional block diagram

7.4 Design description

The nRF24LU1+ USB dongle consists of one printed circuit board, the nRF2728.

7.4.1 nRF ISP interface - firmware upgrade

A connector for the nRF24LU1+ ISP interface is included on the USB dongle for easy firmware upgrade purposes. The interface is found on connector J2. The USB dongle can be programmed using the nRFgo nRF24LU1+ Development Kit together with an nRFgo Starter Kit. For further information, please see the nRFgo nRF24LU1+ Development Kit User Guide.

7.4.2 Power supply

The nRF24LU1+ is connected to the USB 5 V supply through a 10 Ω resistor (R6). An internal voltage regulator enables the chip to be powered directly from the USB bus; it does not require an external voltage regulator. The VDD pins are only for internal decoupling. They are connected together and decoupling capacitors are placed close to the VDD pins, as recommended in the nRF24LU1+ Product Specification.

7.4.3 Matching network

To make the USB dongle as small as possible, the layout of the matching network is changed from the reference layout in the nRF24LU1+ Product Specification. The component values are different in order to compensate for the changed layout. As the component values in this design are very sensitive to layout changes, it must be stressed that if the layout is changed, the matching network and the antenna must be re-tuned for optimum performance by using a vector network analyzer.

7.4.4 Antenna

The antenna is a quarter wavelength meander type antenna printed on the PCB. The antenna is tuned to be resonant at 2.44 GHz, and the impedance is matched to the 50 Ω output of the matching network with the use of L4 and C13. When the USB dongle is connected to a computer, the computer adds a significant ground plane to the USB dongle. This means that any antenna measurements must be done with the dongle connected to the computer. The antenna is tuned for this layout only. If the layout is changed and/or a plastic housing is added, the antenna should be tested and possibly re-tuned. The antenna can be re-tuned by changing its length and/or changing the values of L4 and C13. The exact length of the antenna and the values of L4 and C13 must be determined by doing measurements with a vector network analyzer.
7.5 Regulatory pre-compliance testing results

The nRF2728 nRF24LU1+ USB dongle has been tested in a certified lab on key RF parameters for the purpose of demonstrating compliance with ETSI 300-440, FCC CFR 47 paragraph 15.247 and Industry Canada RSS-210 Issue 7.

The reports are located in the hardware files zip package at the following path:

\nRFready Smart Remote 2 x_x_x\Docs\Test reports

The file names are:

- nRF2728 - ETSI 300 440 Test report.pdf
- nRF2728 - FCC part 15 Test report.pdf

7.6 Schematics, Bill of Materials, PCB layout files, production files

The hardware files for the USB dongle will be located in the following folder in the hardware files zip package: \nRFready Smart Remote 2 x_x_x\Hardware files\nRF2728-USB dongle x_x_x. In this folder you can find Bill of Materials, schematics and pcb-layout files in pdf-format, Altium Designer files, and production files (assembly drawings, gerber files, drill files, pick & place files).
8  Firmware update of the nRFready Smart Remote 2

A step-by-step guide for connecting your remote control to the supplied debugger for firmware upgrades on the nRF51822 device is included in this section.

8.1  J-Link Lite CortexM JTAG/SWD-emulator

The programming and debugging interface of the nRF51822 is accessed through a 10 pin connector (P1). To program and debug the device, the reference design kit includes a J-Link Lite CortexM JTAG/SWD emulator from SEGGER. Make sure that pin 1 on the 10 pin connector on the SEGGER J-Link Lite CortexM is connected to pin 1 on the P1 connector on the PCA20002 radio module. Figure 36 shows the position of pin 1 on the SEGGER J-Link Lite Cortex M.

![Figure 36 Pin 1 position on SEGGER J-Link Lite](image)

To connect the nRF51822 radio module to the JTAG/SWD emulator, use the 10 pin flat cable supplied with the J-Link. Connect the cable to the nRF51822 radio module so there will be a 1-1 mapping of the pins. Figure 37 shows what the connection should look like.

![Figure 37 SEGGER J-Link Lite connected to the nRF51822 radio module](image)
8.2 Flash programming

The programming interface on the PCA20002 allows you to program your application to flash memory. Either as a stand-alone application or on top of the S110 nRF51822 SoftDevice (protocol stack). Note that the S110 is not a part of your application, but a completely separate binary. For more information, see the S110 nRF51822 SoftDevice Specification.

In this chapter we describe how to program and erase the S110 nRF51822 SoftDevice or another application HEX file on the nRF51822 chip.

If you want to start developing on the nRF51822 chip without using the S110 nRF51822 SoftDevice see section 8.2.5 on page 47.

8.2.1 Programming and erasing flash using nRFgo Studio

Use nRFgo Studio to program or erase a SoftDevice or application HEX file onto the nRF51822 chip.

**Note:** For details on memory organization and protection see the nRF51 Series Reference Manual.

![nRFgo Studio dashboard](figure38.png)
8.2.2 Selecting a board to program

1. Open nRFgo Studio.
2. In the Device Manager pane select nRF51 Programming.
3. The nRF51822 Smart Remote module cannot be selected directly. The module must be selected through the J-Link debugger connected to it. To do this, select the debugger you want to use from the **Segger to use** drop-down list with available J-Link debuggers.

8.2.3 Identifying the nRF51 chip and chip content

When you select a board, nRFgo Studio identifies the nRF51822 chip and how its memory is organized. The following chip and memory information is displayed:

- **nRF51 chip identification** - Identifies the chip by name and code variant (for example, nRF51822 QFAACA). If the debugger is not connected to the chip, or the debugger has a problem communicating with the chip, it will show the following message “No device detected. Ensure that you have the SEGGER connected correctly to the board and that the board is powered and configured for debugging.”

- **Code memory** - Shows how the code memory is organized in one or two regions (Region 0 and 1) and the size of each region. For devices containing a SoftDevice, the code memory is divided in two regions, with the SoftDevice in Region 0. The tool shows you how much memory is used by the SoftDevice and how much is left for the application.
  - Memory readback protection - Shows how the readback protection is set. The two possible options are readback protection on Region 0 or readback protection of the whole code memory. If there is only one region the option is readback protection on (All) or off.
  - SoftDevice identification - nRFgo Studio tries to identify the firmware located in the chip at Region 0. For the firmware that it recognizes it prints the ID (in clear text) for the unrecognized firmware it prints the FWID number.

8.2.4 Erase all

Use Erase all in the following situations:

- You have a chip that is programmed with a SoftDevice but you want to remove it and have a blank chip.
- You have programmed an application on a clean chip using nRFgo Studio with the option “Lock entire chip from readback”.

To use the Erase All function, follow the steps in section 8.2.2. Then click **Erase all**.
8.2.5 Programming a SoftDevice

This function lets you program the SoftDevice onto the chip.

1. Follow the steps in section 8.2.2 on page 46 and then select the Program SoftDevice tab.
2. Click Browse and select the HEX file to program.
3. Select whether to enable or disable readback protection of Region 0.
4. Set the SoftDevice size. This sets the size of the code memory region 0 and will not be available if the size is defined by the HEX file.

**Note:** The S110 nRF51822 SoftDevice is included in the installer file of the Smart Remote 2.
8.2.6 Programming an application

This function lets you program an application onto the chip.

Before nRFgo Studio starts programming it verifies that the HEX file matches the actual memory configuration. If it matches, nRFgo Studio continues with the programming, if not it stops the programming and returns an error message. For example, if an application requires the SoftDevice on the chip, it will check the memory configuration for the SoftDevice before programming the chip.

**Note:** This programming will not set up any memory Regions.

![Programming of application on nRF51 device](image)

1. Follow the steps in section 8.2.2 on page 46 and then select the **Program Application** tab.
2. Click **Browse** and select the HEX file to program.
3. Select whether to enable or disable readback protection of the entire chip. If you enable readback protection, you will have to do an Erase All to reprogram the chip again.

A chip that is programmed with **Lock entire chip from read back** enabled will not work with a development toolchain. To make it work you must perform Erase all.

**Lock entire chip from read back** can be used to prevent an accidental overwrite of the chip content.
9 Troubleshooting

The Synaptics Pointing Device Driver does not recognize the dongle.

1. Make sure the drivers on the computer are installed correctly.
2. Restart the computer after installation.
3. If the dongle has been put into a new USB port, the Synaptics driver might need to be reinstalled before it can detect the dongle.

The trackpad is not working.
The remote control is using an accelerometer to wake it up from sleep. If the remote is lying completely still, you might need to tap the trackpad a bit to wake up the system.

Getting bad RF performance while using USB 3.0
Some USB 3.0 controllers send out noise that can disturb RF performance of the USB dongle resulting in a shorter operating range for the remote control. Try positioning the USB dongle as far away from other connected USB 3.0 peripherals as possible if you are experiencing problems.

The Smart Remote does not bond to the master.
The Smart Remote advertises for a certain amount of time before going to sleep. The advertising state is indicated by the **Connection** LED blinking orange. All steps in the bonding procedure must be completed while the Smart Remote is in this state.

If the **Connection** LED stops blinking, try moving the Smart Remote to wake up the system from sleep or push a button on the keyboard.

**nRFgo Studio says that the supply voltage is too low when trying to program the remote control.**
Make sure the nRF51822 Smart Remote module is connected to the base board and that the power switch is set to **ON**.

The SEGGER J-Link Lite CortexM should be connected to the nRF51822 Smart Remote module as shown in **Figure 37 on page 44**.

**nRFgo Studio says that the chip cannot be detected when trying to program the USB dongle.**
Make sure the dongle is plugged into the USB port on the nRF2727 Development Kit module and that both the nRFgo Starter Kit Motherboard and nRF2727 module are connected to the USB port on the computer. Furthermore, ensure that the power switch on the nRF2727 module is set to **VBUS**.

Make sure the **Program dongle** check box in nRFgo Studio is enabled. Please see the **nRFLU1+ Development Kit User Guide** for details.
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 dealer, please see http://www.nordicsemi.com. Information regarding product updates, downloads, and technical support can be accessed through your My Page account on our homepage.

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

Revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 2013</td>
<td>1.0</td>
<td>First release</td>
</tr>
</tbody>
</table>

ARM statement
Keil, μVision, and Cortex are trademarks of ARM Limited. All other brands or product names are the property of their respective holders.