Contents

1 Introduction .................................................................................................................. 3
  1.1 Requirements ............................................................................................................. 3
  1.2 Writing conventions .................................................................................................... 3

2 Kit content ................................................................................................................... 4
  2.1 Hardware content ....................................................................................................... 4
  2.2 Downloadable content .............................................................................................. 5

3 Quick start ................................................................................................................... 6
  3.1 Prepare the hardware .................................................................................................. 6
  3.2 Installing the required packages on Ubuntu ............................................................... 7
  3.3 Updating to the correct firmware for the nRF device ................................................ 7
  3.4 Voice over HID over GATT demo using Ubuntu ......................................................... 7

4 System overview .......................................................................................................... 11
  4.1 Input sources ............................................................................................................ 11
  4.2 Audio processing ....................................................................................................... 12
  4.3 Firmware upgrading ................................................................................................ 14
  4.4 Supported operating systems .................................................................................... 14

5 PCA63506 Voice Input Module hardware description .................................................... 16
  5.1 Key features ............................................................................................................ 16
  5.2 Hardware figures ...................................................................................................... 17
  5.3 Block diagram ......................................................................................................... 18
  5.4 Design description .................................................................................................... 19
  5.5 Schematics, Bill of Materials, PCB layout files, production files ................................ 29

6 Firmware update of the nRFready Voice Input Module .................................................. 30
  6.1 J-Link Lite CortexM JTAG/SWD-emulator ............................................................... 30
  6.2 Firmware update ....................................................................................................... 32

7 Troubleshooting ........................................................................................................... 33
1 Introduction

This reference design is the ideal platform for developing voice enabled remote controls for internet-enabled TVs and set-top boxes, or other voice enabled applications. The Voice Input Module is provides a low power and cost efficient voice input design. By sampling audio from a digital microphone or a line input, the Voice Input Module can send compressed audio stream to an nRFready Smart Remote or an Evaluation Kit, which then transmits it through a Bluetooth® low energy link.

The Voice Input Module has hardware support for two different audio input paths: digital microphone and analog line input. The on-board Cortex-M0 MCU handles digital sampling and compression of the audio with ADPCM codec, while the 64 kbps audio bitstream on UART can be received by the Smart Remote or Evaluation Kit.

1.1 Requirements

Demo requirements

To demonstrate the voice streaming capabilities of the nRFready Voice Input Module, the following is needed:

- One of the following Smart Remotes:
  - nRF6924 nRFready Smart Remote 2
  - nRF6920 nRFready 2.4 GHz RF Smart Remote with nRF6925 Smart Remote 2 Upgrade Kit
  - nRF6921 nRFready μBlue™ Smart Remote with nRF6925 Smart Remote 2 Upgrade Kit
- Bluetooth® low energy:
  - Bluetooth 4.0 enabled hardware
  - Linux with Bluetooth 4.0 and HID over GATT support (for example BlueZ 5.2 or later)
- Google Chrome

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 nRFready Voice Input Module Reference Design consists of hardware and access to software components, reference design files, and documentation from www.nordicsemi.com.

2.1 Hardware content

![Figure 1 Hardware content](image-url)
2.2 Downloadable content

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

2.2.1 nRFready Voice Input Module Reference Design firmware installer file

- Firmware for nRFready Voice Input Module
  - Precompiled HEX files
  - Source code
  - Keil ARM® project files

2.2.2 nRFready Voice Input Module Reference Design documentation

- This User Guide
- Firmware documentation (will be available in a folder on your 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 Voice Input Module Reference Design.

- 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 nRFready Voice Input Module Reference Design Ubuntu platform packages

The ZIP file contains .deb packages required to run the Voice Input Module demo on Ubuntu. Most of these packages are unmodified versions of standard Linux components. Source code is provided for the non-standard packages.

- Package binaries (.deb)
- Source code

2.2.5 Other relevant files

- nRFReady Smart Remote 2 Firmware and documentation
- nRFReady Smart Remote 2 User Guide
- nRF51822 Evaluation Kit User Guide
3 Quick start

This section shows you how to set up the nRFready Voice Input Module Reference Design and run a voice recognition example on an Ubuntu Linux PC using Google Chrome and voice search.

3.1 Prepare the hardware

1. Unpack your nRFready Voice Input Module kit.
2. Depending on your available hardware, plug the Voice Input Module onto the Smart Remote base board or the Evaluation Kit.

*Figure 2 Voice Input Module plugged into the Smart Remote, bottom view*

*Figure 3 Voice Input Module plugged into the Evaluation Kit*
3.2 Installing the required packages on Ubuntu

Along with this User Guide and Firmware package, a set of Ubuntu Linux packages are available for download on the Nordic website. Both the compiled packages and the source code is available to download.

In order to run the voice recognition example, these packages must be installed on a Ubuntu 14.04 LTS Desktop Edition (http://releases.ubuntu.com/trusty/).

The packages are found in the `binaries` folder of the Ubuntu package download. `HOWTO-intall.txt` found in this folder details the installation process.

3.3 Updating to the correct firmware for the nRF device

Refer to the relevant User Guide for firmware upgrade procedure. In the case the Smart Remote 2 is used, download the latest firmware from the Nordic website and follow the firmware upgrade instructions found in the Smart Remote 2 User Guide.

3.4 Voice over HID over GATT demo using Ubuntu

Pairing the Bluetooth low energy device

1. Start a new Bluetooth pairing procedure with the Smart Remote or the Evaluation Kit.
2. On your computer, click the Bluetooth icon on the toolbar and then click Bluetooth Settings.
3. To search for a new device, click + in the Bluetooth window.
4. When discovered, you will see Smart Remote 2 or your device name in the Device list. Select your device and click **Continue** to begin pairing.

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

**Configure audio input**

1. Under All Settings, select **Sound** and then select the **Input** tab.
2. Select the NVS device from the list of input sources.
3. The **Input level** should now indicate that it is receiving input. The LED on the Voice Input Module indicates it is streaming audio by lighting up.

**Note:** Press button SW2 on the Voice Input Module to toggle the LED streaming indication.

4. To stop streaming, select **All Settings** or close the window.
Test voice recognition

1. Open up Chrome. Select **Dash Home** and type "Google Chrome". Click the Google Chrome icon that is displayed.

![Dash Home](image)

2. Go to www.google.com. If you are redirected to a local Google version, click **Google.com** in the lower-right corner of the webpage.

3. Click the microphone icon. The recording LED **LD1** on the Voice Input Module should light up immediately (press button **SW2** to turn it off), and you can speak into the Smart Remote. Chrome will stop recording automatically when you stop talking.

**Note:** If you do not see the microphone symbol, the GVoice application might not be installed. Go to the Google Chrome Web Store to download and install GVoice.
Listening to audio quality

Audio quality can be verified by looping the sound directly from the Ubuntu input source (Voice Input Module) to the output (speakers).

Enable loopback

1. Open a terminal (CTRL + ALT + T).
2. In the terminal window, type the following command:
   `pactl load-module module-loopback latency_msec=20`.

   **Note:** The command `latency_msec=20` is optional. It helps to minimize the acoustic feedback. Some machines do not accept the command and may return error codes. In that case, just invoke module-loopback without specifying latency.

3. If successful, this command returns a handle number, invoking this command multiple times generates multiple loopback instances with independent handles.

Disable loopback

In the terminal window, type the following command `pactl unload-module x`, where x is the handle module number returned when enabling the loopback.

   **Note:** If the Smart Remote is not connected and selected as the audio input source, sound will be streamed from the PC microphone to the PC speakers causing acoustic feedback.
4 System overview

This chapter describes the functionality of the Voice Input Module including how it can be used for development purposes. The nRFready Voice Input Module is designed to be plugged into a nRFready Smart Remote or an Evaluation Kit and cannot be used when not plugged into one of these devices. The module includes a digital microphone, 3.5 mm input jack, analog filtering circuitry, and a Cortex M0-based MCU.

![Figure 4 System overview](image)

4.1 Input sources

Two different audio input sources are available:

- Digital microphone
- Line input 3.5 mm jack

The main input source is the digital microphone. Placed on the Voice Input Module itself, it allows audio to be recorded without connecting an audio source. In the event that an analog audio input signal is desired, an optional 3.5 mm jack is included.

**Note:** The initial firmware version only supports digital microphone sampling.
4.2 Audio processing

Depending on the audio input source, different means of sampling and filtering are used. The digital audio signal from the microphone is sampled and filtered digitally. The analog signal on the other hand, passes through an analog filter prior to being sampled.

4.2.1 Digital sampling

The digital microphone is a 1 bit sigma-delta modulator, which outputs a pulse-density modulated (PDM) signal. The signal is synchronized with the microphone clock input, and can be sampled using SPI.

The audio MCU acts as an SPI master, and provides the clock for the digital microphone. The microphone data output is sampled on the SPI input line of the audio MCU. This type of microphone requires a clock frequency between 1 and 3 MHz. A typical PDM signal is shown in the figure below.

![Figure 5 Typical PDM signal](image)

Compared to the desired 16 kHz sampling frequency ($f_s$), the PDM output signal is oversampled by a factor of 60 or more. The oversampling shifts noise components into the upper frequency band of the PDM signal. The signal must pass through a digital low-pass filter in order to remove the high-frequency noise components.

Note that the SPI clock defines the sampling rates of the entire digital audio signal chain. If the SPI clock speed is altered, adjustments must be made to filter coefficients, buffer sizes, and so on. While the default SPI clock is provided by the audio MCU, the hardware can be configured so that the SPI clock is provided externally (by the nRF51) instead. See Section 5.4.2 “Digital microphone” on page 20 for more information. This makes it possible to remove the 18.432 MHz crystal, at the cost of not being able to achieve $f_s$ of 16 kHz.
4.2.2 Analog sampling

The analog line in signal is passed through an analog filter and connected to one of the ADC inputs on the audio MCU. Because of the analog filter, the ADC can be run at 16 kHz without any oversampling. See Section 5.4.3.2 “Anti-aliasing filter” on page 24 for details regarding the analog filtering.

4.2.3 Digital filtering

In order to fit the encoder input format and the bandwidth requirements, the digital signal needs to be filtered and compressed. The filtering process reduces $f_s$ from 1-3 MHz to 16 kHz and converts the signal from PDM to PCM (pulse-code modulated) format. Such a process of low-pass filtering and downsampling a signal is called ‘decimation’.

Figure 6 illustrates the decimation filter. However, as this filter is purely digital, the implementation might differ between firmware versions. Details regarding the decimation filtering are found in the firmware documentation.

![Decimation filter diagram]

Figure 6 Decimation filter

4.2.4 Encoding

Adaptive differential pulse-code modulation (ADPCM) encoding is used to compress the audio signal to a data rate suitable for Bluetooth low energy. In order to keep the decoder synchronized with the encoder, the audio stream is divided into 32 ms long frames. Each frame is prefixed with the encoder state. This makes each frame an individual set of data, which will handle frame loss in the event of range or RF noise issues. When a frame is ready, it is transmitted using the UART interface.

<table>
<thead>
<tr>
<th>Byte no.</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>MSByte of first frame sample</td>
</tr>
<tr>
<td>1</td>
<td>LSBByte of first frame sample</td>
</tr>
<tr>
<td>2</td>
<td>ADPCM step table index</td>
</tr>
<tr>
<td>3 - 258</td>
<td>Encoded samples</td>
</tr>
</tbody>
</table>

Table 1 Frame format
4.3 Firmware upgrading

The Voice Input Module firmware can be modified and upgraded. See Chapter 6 “Firmware update of the nRFready Voice Input Module” on page 30 for how to upgrade the firmware.

4.4 Supported operating systems

The nRFready Voice Input Module requires the host to be able to decode the ADPCM encoded audio the module delivers. The demo described in this user guide complies with the Bluetooth 4.0 HID over GATT profile and is compatible with all host systems incorporating this profile. The voice recognition example and ADPCM decoding have been implemented on a version of BlueZ supporting HID over GATT.
4.4.1 Ubuntu

The nRFready Voice Input Module is delivered with a set of compiled software packages which enables the voice recording feature on Ubuntu 14.04 LTS Desktop Edition.

Figure 7 illustrates the relevant parts of Ubuntu software architecture and specifically highlights what has been modified.

The main components seen in the figure are:

- BlueZ (Linux Bluetooth subsystem)
- PulseAudio (audio subsystem used by Ubuntu)

The BlueZ version needs to support the HID over GATT Profile (version 5.2 or newer).

Note: This implementation uses an earlier version of BlueZ (4.101) which has been modified to include HID over GATT.

A set of PulseAudio plugins is used to detect the presence of a voice enabled HID device and to decode and insert the decoded audio into the audio subsystem. module-nvs-audio makes the voice enabled Bluetooth device appear as a normal microphone and translates "start/stop recording" signals into Bluetooth packet transmissions. module-nvs-detect looks for voice enabled HID devices and initiates the necessary components when one is found.
5 PCA63506 Voice Input Module hardware description

The PCA63506 Voice Input Module is designed to plug onto the nRFready Smart Remote or the Evaluation Kit. The module contains a Cortex M0 MCU from ST Microelectronics for audio sampling and compression, and has two audio input options: a digital MEMS (Micro Electrical-Mechanical System) microphone and an analog line input with a filter and preamplifier. A programming interface for firmware upgrade and development for the MCU is also included. To simplify current measurements, the module has a set of jumpers which can be replaced with ampere meters for current measurements of the individual blocks on the board.

![High level block diagram](image)

**Figure 8** High level block diagram

5.1 Key features

The Voice Input Module has the following key features:

- STM32 Cortex M0 MCU for audio compression
- Digital MEMS microphone
- Line input with active anti-aliasing filter and preamplifier
- Current measurement headers
- Interface compatible with Smart Remote and Evaluation Kit
- Configurable design and test points for development purposes
- ARM SWD programming/debugging interface
5.2 Hardware figures

Figure 9 Voice Input Module top view
5.3 Block diagram

![Voice Input Module block diagram](image)

**Figure 11** Voice Input Module block diagram
5.4 Design description
This chapter contains details about the hardware blocks on the module.

5.4.1 Power supply scheme and current measurement
A set of jumpers are available for simple measuring of the current consumption of the different blocks of the hardware. The jumpers are in series with the power supply for:

- MCU
- Digital microphone
- Analog filter and preamplifier

The power supply for the MCU is fed directly from the Smart Remote or the Evaluation Kit, but the power supply for the digital microphone and analog filter circuitry passes through a power switch circuit that is controlled by the MCU. See Figure 12 for more information.

![Figure 12 Power supply scheme and current measurement headers](image)

![Figure 13 Current measurement on the MCU](image)
5.4.2 Digital microphone

The PCA63506 module is equipped with a digital MEMS microphone. The microphone is located on the bottom side of the PCB and has a bottom port, meaning the sound is recorded through a hole in the PCB, see *Figure 14*.

![Figure 14 Location of microphone port](image)

The data and clock lines to the microphone are by default connected to the on-board audio compression MCU. Optionally, the lines can also be connected to the nRF chip on the Smart Remote or the Evaluation Kit. By soldering **SB5** and **SB6**, the clock and data signals for the digital microphone is routed to the Smart Remote and Evaluation Kit interface connectors, see *Figure 15*. See *Section 5.4.5 “Smart Remote interface”* on page 26 and *Section 5.4.6 “Evaluation Kit interface”* on page 27 for more information on the connection to the nRF51 chip.

**Note:** Requires FW update of the MCU.

![Figure 15 Connecting digital microphone lines to the interface connectors](image)
5.4.2.1 Using the nRF chip as digital microphone clock source

Even though the on-board MCU is used for sampling data from the digital microphone by default, there is also an option to supply the clock signal from the nRF51 chip. To do this, make sure SB5 is soldered and flip SW3 to the nRF position, see *Figure 16*.

**Note:** Requires FW update of the MCU.

![Supplying clock from the nRF chip](image)

*Figure 16* Supplying clock from the nRF chip

When driving the clock from an external device, we change the data input line for the audio sampling MCU from MOSI to MISO. This must be done because the SPI peripheral of the audio sampling MCU uses a different data input pin when configured as a SPI Master compared to when it is configured as a SPI slave (the SPI master provides the clock signal). *Figure 17* shows how the signals are swapped.

![Digital microphone configuration](image)

*Figure 17* Digital microphone configuration
5.4.3 Analog line input

The module has a 3.5 mm jack input for connecting analog audio signals. By default the signal is routed through an active anti-aliasing filter and a preamplifier before it is routed into the ADC of the MCU. The analog signal chain is designed to be configurable for development purposes. The filter and preamplifier can be bypassed by cutting SB10 and SB15, and soldering SB17. See Figure 18.

![Figure 18 Bypassing the filter and preamplifier](image_url)

As an option, the analog signal can be routed to the Evaluation Kit to be sampled on the nRF device. By soldering SB8 (see Figure 19) the analog signal is routed to the P0.02 pin on the Evaluation Kit.

Test point TP6 is located at the output of the analog chain for monitoring of the signal entering the ADC.

![Figure 19 Routing the analog signal to the Evaluation Kit](image_url)
5.4.3.1 Channel selection and DC offset

There are two solder bridges located next to the 3.5 mm jack connector, SB2 and SB3, which make it possible to choose the left or right channel of the input signal, see **Table 2**.

<table>
<thead>
<tr>
<th>Solder bridge</th>
<th>Channel</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SB2</td>
<td>Left</td>
<td>Default shorted</td>
</tr>
<tr>
<td>SB3</td>
<td>Right</td>
<td>Open</td>
</tr>
</tbody>
</table>

**Table 2 Line input channel selection**

To switch from left to right channel input, cut the shorting of SB2 and solder SB3, see **Figure 20**.

After the channel selection the signal passes through a DC block capacitor, C18, and is then DC shifted with the resistor network R7 and R8 which will make the signal swing around a voltage greater than zero. The voltage divider network has been tuned to match the filter and preamplifier circuit, giving the most optimal voltage swing on the input of the ADC of the MCU.

Test point TP1 is available for monitoring the signal after the DC block capacitor.
5.4.3.2 Anti-aliasing filter

The anti-aliasing filter on the Voice Input Module is designed as a 6th order elliptic filter with a frequency response of -3 dB at 5 kHz. The filter is built up of three 2nd order low pass notch stages, which has a Buctor topology.

The overall filter frequency response is shown in Figure 21. In this figure the response is normalized to 0 dB, but the filter has an overall voltage gain of 4.9 at 1 kHz.

![Figure 21: Anti-aliasing filter frequency response](image1)

In the filter chain, test points are available for monitoring the signal.

- TP2 is on the output of stage 1
- TP3 is on the output of stage 2
- TP4 is on the output of stage 3

The filter can be bypassed by cutting the short on SB10 and SB13, and solder SB16, see Figure 22.

![Figure 22: Bypassing anti-aliasing filter](image2)
5.4.3.3 Preamplifier

The preamplifier is a simple op-amp amplifier circuit with adjustable gain. The voltage gain can be adjusted from 1 to 4.6 by using the *Rp1* potentiometer. See *Figure 23*.

Test point *TP5* is available on the output of the preamplifier for monitoring of the output signal.

*Figure 23 Preamplifier gain and test point*
5.4.4 Button and LED

The Voice Input Module features a tactile button and a LED. The button is connected to pin PB1 of the MCU and has active low functionality. The LED is connected to pin PA3 and is sunked through a current limiting resistor, See Figure 24.

![Figure 24 Button and LED]

5.4.5 Smart Remote interface

The interface to the Smart Remote consists of three connectors, P1, P2, and P3. All connectors are placed on the bottom side of the board and are positioned so the module will fit on the available headers on the Smart Remote Base Board, see Figure 10 on page 18. P1 is only used for mechanical stability. For pin descriptions on connector P2 and P3, see Table 3 and Table 4.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Smart Remote pin name</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCC</td>
<td>VCC’</td>
<td>Power supply</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>3</td>
<td>SDA</td>
<td>I2C_SDA</td>
<td>Two-wire slave data</td>
</tr>
<tr>
<td>4</td>
<td>SCL</td>
<td>I2C_SCL</td>
<td>Two-wire slave clock</td>
</tr>
</tbody>
</table>

Table 3 P2 connector pin information

<table>
<thead>
<tr>
<th>Pin</th>
<th>Smart Remote pin name</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCC</td>
<td>VCC’</td>
<td>Power supply</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>3</td>
<td>SPI_SCK</td>
<td>UART_TXD</td>
<td>UART data output for audio</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CLOCK_SYNC</td>
<td>Signal for clock synchronization</td>
</tr>
<tr>
<td>4</td>
<td>SPI_MOSI</td>
<td>MIC_CLK_EXT</td>
<td>Digital microphone clock line</td>
</tr>
<tr>
<td>5</td>
<td>SPI_MISO</td>
<td>MIC_DOUT_EXT</td>
<td>Digital microphone data line</td>
</tr>
<tr>
<td>6</td>
<td>CSN_1</td>
<td>UART_CTS/WAKEUP</td>
<td>UART handshake/wakeup on pin signal</td>
</tr>
</tbody>
</table>

Table 4 P3 connector pin information
5.4.6 Evaluation Kit interface

To connect to the Evaluation kit, the Voice Input Module has two 10 pin connectors placed on the bottom side of the Module. They are positioned so the module will snap in easily on the Evaluation Kit board, see Figure 10 on page 18. See Table 5 and Table 6 for reference and pin descriptions on connector P4 and P5.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Evaluation Kit pin name</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P0.24</td>
<td>BOOT0</td>
<td>MCU boot memory selection</td>
</tr>
<tr>
<td>2</td>
<td>P0.25</td>
<td>NC</td>
<td>Not connected</td>
</tr>
<tr>
<td>3</td>
<td>P0.26</td>
<td>NC</td>
<td>Not connected</td>
</tr>
<tr>
<td>4</td>
<td>P0.27</td>
<td>NC</td>
<td>Not connected</td>
</tr>
<tr>
<td>5</td>
<td>P0.28</td>
<td>MIC_CLK_EXT</td>
<td>Digital microphone clock line</td>
</tr>
<tr>
<td>6</td>
<td>P0.29</td>
<td>MIC_DOUT_EXT</td>
<td>Digital microphone data line</td>
</tr>
<tr>
<td>7</td>
<td>P0.30</td>
<td>NC</td>
<td>Not connected</td>
</tr>
<tr>
<td>8</td>
<td>P0.31</td>
<td>NC</td>
<td>Not connected</td>
</tr>
<tr>
<td>9</td>
<td>GND</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>10</td>
<td>VCC_IO</td>
<td>VCC'</td>
<td>Power supply</td>
</tr>
</tbody>
</table>

Table 5 P4 connector pin information

<table>
<thead>
<tr>
<th>Pin</th>
<th>Evaluation Kit pin name</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P0.00</td>
<td>NC</td>
<td>Not connected</td>
</tr>
<tr>
<td>2</td>
<td>P0.01</td>
<td>UART_CTS/WAKEUP</td>
<td>UART flow control/wakeup on pin signal</td>
</tr>
<tr>
<td>3</td>
<td>P0.02</td>
<td>EK_ANA</td>
<td>Signal from analog input after preamplifier</td>
</tr>
<tr>
<td>4</td>
<td>P0.03</td>
<td>UART_TXD</td>
<td>UART data output for audio</td>
</tr>
<tr>
<td>5</td>
<td>P0.04</td>
<td>I2C_SCL</td>
<td>Two-wire slave clock</td>
</tr>
<tr>
<td>6</td>
<td>P0.05</td>
<td>I2C_SDA</td>
<td>Two-wire slave data</td>
</tr>
<tr>
<td>7</td>
<td>P0.06</td>
<td>CLOCK_SYNC</td>
<td>Signal for clock synchronization</td>
</tr>
<tr>
<td>8</td>
<td>P0.07</td>
<td>nRST</td>
<td>MCU reset</td>
</tr>
<tr>
<td>9</td>
<td>GND</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>10</td>
<td>VCC_IO</td>
<td>VCC'</td>
<td>Power supply</td>
</tr>
</tbody>
</table>

Table 6 P5 connector pin information
5.4.7 I/O usage

The STM32F050F4P6 has 15 generic I/Os available. All I/Os are used in this design and are organized as shown in Table 7.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA0</td>
<td>UART_CTS/WAKEUP</td>
<td>UART flow control / wakeup on pin signal</td>
</tr>
<tr>
<td>PA1</td>
<td>PWR_CTRL</td>
<td>Power control signal for digital microphone and analog filter</td>
</tr>
<tr>
<td>PA2</td>
<td>UART_TXD</td>
<td>UART data output</td>
</tr>
<tr>
<td>PA3</td>
<td>LD1</td>
<td>LED output</td>
</tr>
<tr>
<td>PA4</td>
<td>ADC_IN</td>
<td>Input from analog chain</td>
</tr>
<tr>
<td>PA5</td>
<td>MIC_CLK</td>
<td>Digital microphone clock line</td>
</tr>
<tr>
<td>PA6</td>
<td>PA6</td>
<td>Data from digital microphone/clock sync line (see Section 5.4.2.1 on page 21)</td>
</tr>
<tr>
<td>PA7</td>
<td>PA7</td>
<td>Data from digital microphone/clock sync line (see Section 5.4.2.1 on page 21)</td>
</tr>
<tr>
<td>PA9</td>
<td>I2C_SCL</td>
<td>Two-wire slave clock</td>
</tr>
<tr>
<td>PA10</td>
<td>I2C_SDA</td>
<td>Two-wire slave data</td>
</tr>
<tr>
<td>PA13</td>
<td>SWDIO</td>
<td>Programming and debugging data line</td>
</tr>
<tr>
<td>PA14</td>
<td>SWCLK</td>
<td>Programming and debugging clock line</td>
</tr>
<tr>
<td>PB1</td>
<td>SW2</td>
<td>Button input</td>
</tr>
<tr>
<td>PF0</td>
<td>OSC_IN</td>
<td>18.432 MHz crystal input</td>
</tr>
<tr>
<td>PF1</td>
<td>OSC_OUT</td>
<td>18.432 MHz crystal output</td>
</tr>
</tbody>
</table>

*Table 7 I/O usage*
5.4.8 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 P6, see Figure 9 on page 17.

Chapter 6 “Firmware update of the nRFready Voice Input Module” on page 30 describes firmware upgrade and debugging of the Voice Input Module.

**Note:** Make sure that pin 1 on the PCA63506 P6 connector is connected to pin 1 of the connector on the programmer unit.

<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 8 Voice Input Module SWD interface connector P6 pin description*

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

The hardware design files for the module are located in a downloadable zip package on the Nordic web page, and contains the following: \nRFready Voice Input Module vx.x.x\Hardware files\PCA63506-nRFready Voice Input 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).
6  Firmware update of the nRFready Voice Input Module

To be able to update the firmware on the MCU you will need a programmer/debugging tool that is compatible with the SWD protocol. Please refer to ST Microelectronics, www.st.com, for tools that are compatible with the STM32F050F4P6 MCU used on the Voice Input Module. We will use the SEGGER J-Link Lite debugger that is supplied in the Smart Remote 2 Kit as an example. In this section is a step-by-step guide for connecting your Voice Input Module to the SEGGER J-Link Lite debugger and to update the firmware.

6.1  J-Link Lite CortexM JTAG/SWD-emulator

The programming and debugging interface of the MCU is accessed through a 10 pin connector (P6). To program and debug the device, you may use 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 P6 connector on the PCA63506 Voice Input Module. *Figure 25* shows the position of pin 1 on the SEGGER J-Link Lite Cortex M.

![Figure 25 Pin 1 position on SEGGER J-Link Lite](image-url)

*Figure 25* Pin 1 position on SEGGER J-Link Lite
To connect the Voice Input Module to the JTAG/SWD emulator, use the 10 pin flat cable supplied with the J-Link. Connect the cable to the Voice Input Module so there will be a one-to-one mapping of the pins. Figure 26 shows what the connection should look like.

Figure 26 SEGGER J-Link Lite connected to the Voice Input Module
6.2 Firmware update

After running the firmware installer file, there will be a “Firmware update” project for Keil μVision in the folder structure \Precompiled hex\Firmware upgrade projects\. You will need to download and install Keil to be able to open the project. An evaluation version of Keil μVision can be downloaded from http://www.keil.com free of charge.

After opening the Firmware update project in Keil μVision, click the Download button to program the MCU, see Figure 27.

![Figure 27](image.png)  
*Figure 27* Downloading firmware to the STM32F050F4P6 MCU
7 Troubleshooting

No audio is received when I use the digital microphone and stream audio to the host.
• Make sure that the jumper is present in the **MIC** position on **P7**.
• Make sure the **SW3** switch is in the **STM** position.
• LED **LD1** will light up when the digital microphone is being sampled.

The analog input does not work.
• Make sure that the jumper is present in the **ANALOG** position on **P7**.

For accurate current measurement of the module.
• Make sure the LED **LD1** is not lit when measuring. It can be toggled on and off by pressing button **SW2** while streaming.
• Remove power jumpers for unused functionality (remove **ANALOG** jumper when using the digital microphone, and vice versa when using the line in).

The Voice Input Module does not work when plugged into the Evaluation Kit.
• Check the orientation of the board and that you are using the correct set of Evaluation Kit connectors.
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 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>November 2014</td>
<td>1.1</td>
<td>Changed the reference to Ubuntu version in Section 3.2 “Installing the required packages on Ubuntu” on page 7 and Section 4.4.1 “Ubuntu” on page 15.</td>
</tr>
<tr>
<td>November 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.