

Nordic Thingy:52

User Guide

v1.1

Contents

Revision history	iv
1 Introduction	5
2 Minimum requirements	7
3 Kit content	8
3.1 Hardware content	8
3.2 Downloadable content	8
3.3 Related documentation	9
4 Getting started	10
4.1 Connecting Thingy	11
4.1.1 Connecting Thingy using Bluetooth	11
4.1.2 Connecting Thingy using NFC	12
4.1.3 Troubleshooting	13
4.2 Configuring Thingy	14
4.2.1 Configurable settings	15
5 Services	17
5.1 Environment	18
5.1.1 Live graphs	19
5.1.2 Configurable settings (iOS and Android)	19
5.2 User interface	20
5.2.1 LED	20
5.2.2 Button	21
5.3 Motion	21
5.3.1 3D Motion	22
5.3.2 Gravity vector (iOS and Android)	22
5.3.3 Configurable settings (iOS and Android)	23
5.4 Sound	24
5.4.1 Streaming: Microphone (iOS/Android)	25
5.4.2 Streaming: Speaker	26
5.5 Cloud (IFTTT)	27
6 Managing Thingy devices	29
6.1 iOS	29
6.1.1 Adding a Thingy	29
6.1.2 Removing a Thingy	29
6.1.3 Connecting and disconnecting a Thingy	30
6.2 Android	31
6.2.1 Adding a Thingy	31
6.2.2 Removing a Thingy	32
6.2.3 Connecting and disconnecting a Thingy	32
6.3 Over-the-air device firmware update	32
6.3.1 Updating firmware on a connected Thingy	33
6.3.2 Updating firmware on a Thingy in bootloader mode	33

7	Hardware description	35
7.1	Block diagram	35
7.2	Hardware figures	35
7.3	MCU	36
7.4	I/O expander	37
7.5	Pin maps	38
7.6	Motion sensors	40
7.6.1	Low-power accelerometer	40
7.6.2	Nine-axis motion sensor	41
7.7	Environment sensors	42
7.7.1	Pressure sensor	42
7.7.2	Humidity sensor	42
7.7.3	Color sensor	43
7.7.4	Gas sensor	43
7.8	Sound	44
7.8.1	Digital microphone	44
7.8.2	Amplifier and speaker	44
7.9	LEDs and button	45
7.9.1	RGB LED	45
7.9.2	Button	45
7.10	Power supply	45
7.10.1	Current measurement	47
7.11	Interface	48
7.11.1	N-MOS transistors	48
7.11.2	Connectors	49
	Legal notices	53

Revision history

Date	Version	Description
September 2017	1.1	Updated for Nordic Thingy:52 software version 2.0 <ul style="list-style-type: none">• New feature: Near-field communication (NFC) capability• Updated:<ul style="list-style-type: none">• Connecting Thingy on page 11• Managing Thingy devices on page 29• The LED color indicating connection changed from cyan to green• Restructured the document
June 2017	1.0	First release

1 Introduction

The Nordic Thingy:52™ (nRF6936) is a compact, power-optimized, multi-sensor development kit device designed for collecting environmental data of various types. It is also an easy-to-use development platform, designed to help you build IoT prototypes and demos, without the need to build hardware or write firmware from scratch.

Thingy is built around the nRF52832 *Bluetooth* 5 SoC from Nordic Semiconductor. It connects to Bluetooth-enabled mobile phones, laptops, tablets, Raspberry Pis and similar devices, and sends data to/from its sensors and actuators to an app and to the cloud.

Thingy can sense movement, orientation, temperature, humidity, air pressure, light, color, and air quality. It can also play sound via its speaker and stream sound to the host from its microphone.

The functionality of Thingy can be configured over-the-air via a Bluetooth API. That makes it possible to create demos and prototypes without actually programming the Nordic Thingy:52 itself, and do all development on the app or cloud solution. As new versions of the firmware are released, Thingy can be updated over-the-air using the provided apps. Advanced users can use Thingy as a development kit by building their own firmware and uploading it onto the board.

Key features of Nordic Thingy:52

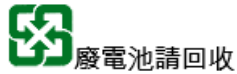
- Highly configurable IoT sensor kit
 - All sensors and *Bluetooth* low energy (BLE) parameters are configurable through a BLE interface.
- nRF52832 System on Chip (SoC) solution
- Configurable RGB LED and button
- Cloud connectivity
- Example apps for Android and iOS, as well as a web application that uses Web Bluetooth API.
- Environmental sensors
 - Temperature
 - Humidity
 - Air pressure
 - Air quality (CO₂ and TVOC)
 - Color and light intensity
- nine-axis motion sensing
 - Tap detection
 - Orientation
 - Step counter
 - Quaternions
 - Euler angles
 - Rotation matrix
 - Gravity vector
 - Compass heading
 - Raw accelerometer, gyroscope, and compass data
- Sound
 - Speaker for playing prestored samples, tones, or sound streamed over BLE (8 bit 8 kHz LoFi)
 - Microphone streaming (ADPCM compressed 16 bit 16 kHz)
- Secure over-the-air device firmware upgrade (OTA DFU)
- Low power consumption

- Near field communication (NFC) support
- All software source code available and documented

**Environmental Protection**

Waste electrical products should not be disposed of with household waste.

Please recycle where facilities exist. Check with your local authority or retailer for recycling advice.



The battery in this product cannot be easily replaced by users themselves. Batteries should only be removed by qualified professionals as appropriate due to safety concerns.

2 Minimum requirements

Before starting to use Thingy, check the minimum hardware and software requirements for the web app.

Supported operating systems

- Android
 - Android 6.0 Marshmallow or later
- Mac OS
 - OS X Yosemite or later
 - BLE-supported MacBook
- Linux
 - Kernel 3.19+
 - BlueZ 5.41+
- Chrome OS
 - BLE-supported hardware

Supported browsers

- Chrome
- Opera

3 Kit content

The Nordic Thingy:52 IoT Sensor Kit consists of hardware and access to software components, reference design files, mobile applications, and documentation.

3.1 Hardware content

Nordic Thingy:52 reference design hardware comprises one item, the IoT sensor kit.



Figure 1: Nordic Thingy:52 hardware content

3.2 Downloadable content

The Nordic Thingy:52 reference design includes firmware source code, documentation, hardware schematics, and layout files.

To obtain the firmware package, go to the [Thingy product page](#).

Firmware package

- Application firmware for Nordic Thingy:52
 - Precompiled HEX files
 - Source code
- nRF5 SDK v12.1.0
- S132 SoftDevice
- Firmware documentation

Mobile apps

Search for **Nordic Thingy** in App Store or Google Play, depending on the mobile platform that you want to use.

The web app is available through the Eddystone beacon that is advertised by the Thingy, unless you have changed the default Eddystone beacon configuration.

For more information, see [Getting started](#) on page 10.

Hardware files

The zip file and its subdirectories contain the hardware design files for the Nordic Thingy:52 reference design. The hardware files for the circuit board are located in the following folder in the hardware files zip package:

```
\Thingy52 - Hardware files x_x_x\PCA20020-Thingy52 Board x_x_x.
```

In this folder, you can find the following hardware design files:

- Altium Designer files
- Schematics and PCB layout files in PDF format
- Bill of materials
- Production files:
 - Drill files
 - Assembly drawings
 - Gerber files
 - Pick-and-place files

3.3 Related documentation

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

- [nRF52832 Product Specification](#)
- [S132 SoftDevice Specification](#)
- [nRF52832 Errata](#)
- [nRF5 SDK](#)

4 Getting started

Setting up Nordic Thingy:52 and controlling it through the provided mobile applications requires a few steps. The procedure for the iOS, Android, and web applications have only slight differences.

Before you start:

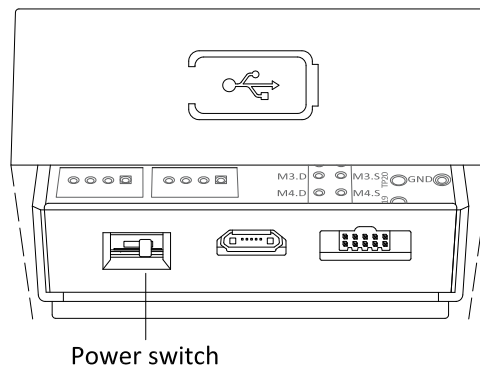
- Unpack Thingy.
- Check [Minimum requirements](#) on page 7.

Compared to the mobile apps, the web app provides somewhat restricted functionality in various areas.

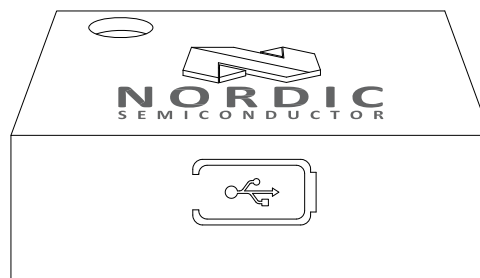
1. Download and install the Nordic Thingy app to control Thingy using your device.
 - iOS device: Nordic Thingy iOS App in Apple App Store
 - Android: Nordic Thingy Android App in Google Play
 - Web app: Access the web app through the Eddystone beacon that is advertised by Thingy. Use an app that allows for detecting nearby beacons, such as the Physical Web app, and go to the URL that is advertised by Thingy. On Android devices, you can also use the built-in Nearby functionality. Make sure that the link is opened by a supported browser, see [Minimum requirements](#) on page 7. You can also access the app by opening the URL `https://developer.nordicsemi.com/thingy/52/`.

After installing the app, you will find its icon on the home screen of your device.

2. Gently remove the rubber overlay to reveal the power switch. You will find it right next to the micro-USB port.



3. Power on Thingy and make sure that it is not connected to any other mobile device.
If Thingy is already connected to another device, its LED will breathe in green color.
4. To charge the Thingy, connect it to a computer or mobile charger via the micro-USB port located on the front side of the device, under a rubber cover.



Note:

The power switch must be on for the device to charge.

Thingy offers no USB connectivity. The micro-USB port is only used for charging the device.

4.1 Connecting Thingy

After installing and launching the mobile app or opening the web app, you must connect Thingy to your mobile device. You can either connect using *Bluetooth* or the NFC tap-to-connect function enabled in firmware 2.0.0 and later.

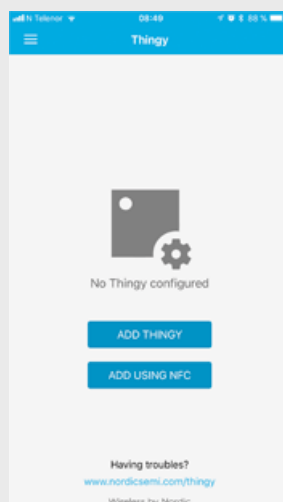
After starting the Thingy app, a welcome screen is displayed.

4.1.1 Connecting Thingy using Bluetooth

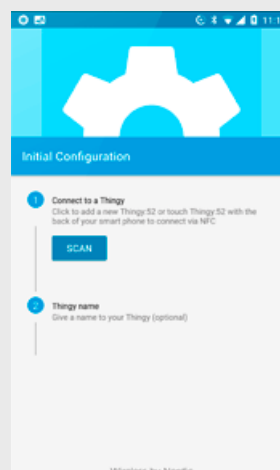
Follow the steps presented here to connect using *Bluetooth* scanning.

1. Add Thingy to your app.

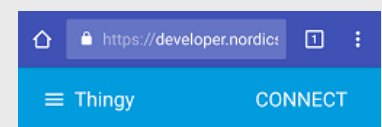
iOS: Tap **Add Thingy**



Android: Tap **Scan**



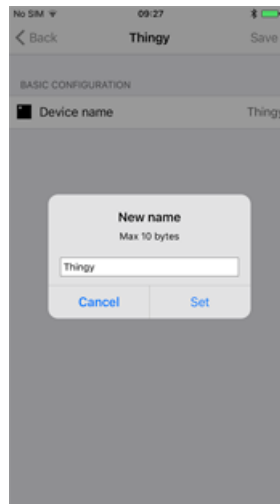
Web: Tap **Connect**, and **Pair**.



Tap **Proceed** to grant the Location Permission which is required for Android 6.0 Marshmallow and later, and tap **Scan** again.

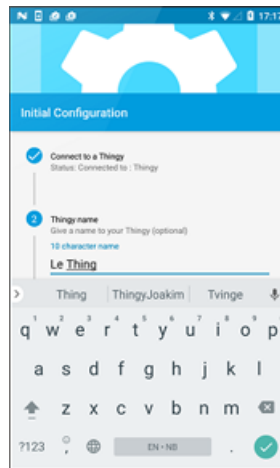
The Scanner starts the discovery process and lists any Thingy that is nearby. If Thingy does not appear in the list within a couple of seconds, check [Troubleshooting](#) on page 13.

2. Add a name for your device to replace the default (optional).
 - iOS: Add a new name and tap **Save**.



If no Thingy has not been configured in the iOS app, it will fall back to the **Add Thingy** view.

- Android: Add a new name.



A **Get Started** button is displayed. Tap it to proceed to the main application. You can also go back to any of the previous steps and make changes.

If there are no devices configured or all of them have been deleted, the app falls back to the **No Thingy configured** view.

- Web: Not configurable

iOS and Android: The app opens the **Environment** service, which is the default view for the Thingy app as long as one Thingy has been configured, regardless of its connection state.

For information on the configurable parameters of Thingy, see [Configurable settings](#) on page 15.

4.1.2 Connecting Thingy using NFC

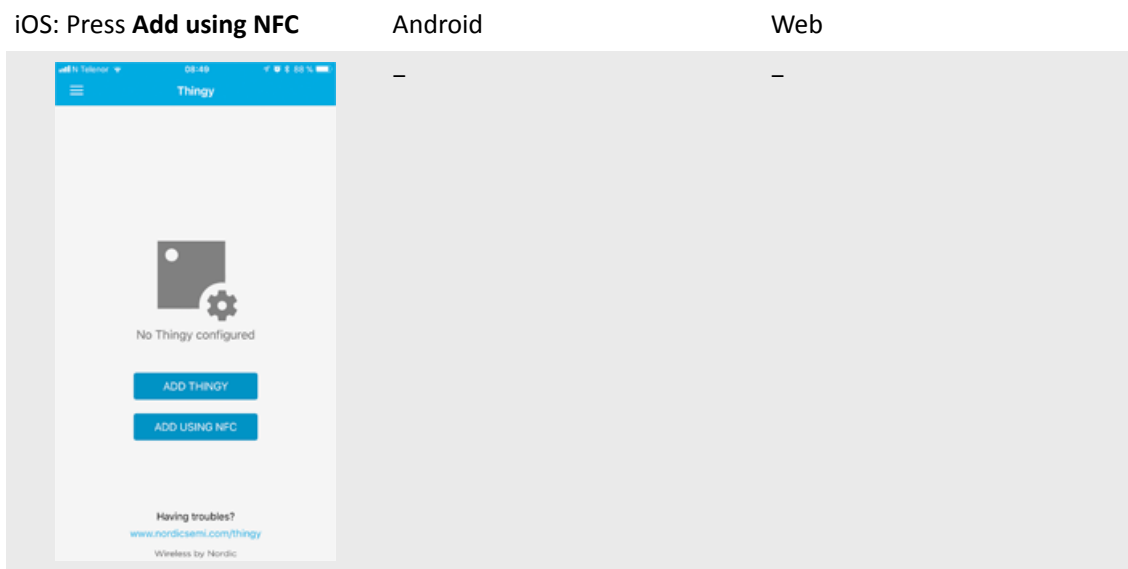
NFC tap-to-connect is available starting from Thingy firmware 2.0.0. The NFC scanning view will appear asking to touch a Thingy with NFC capability to pair.

Before you start, make sure your device has NFC capability and uses the following application versions:

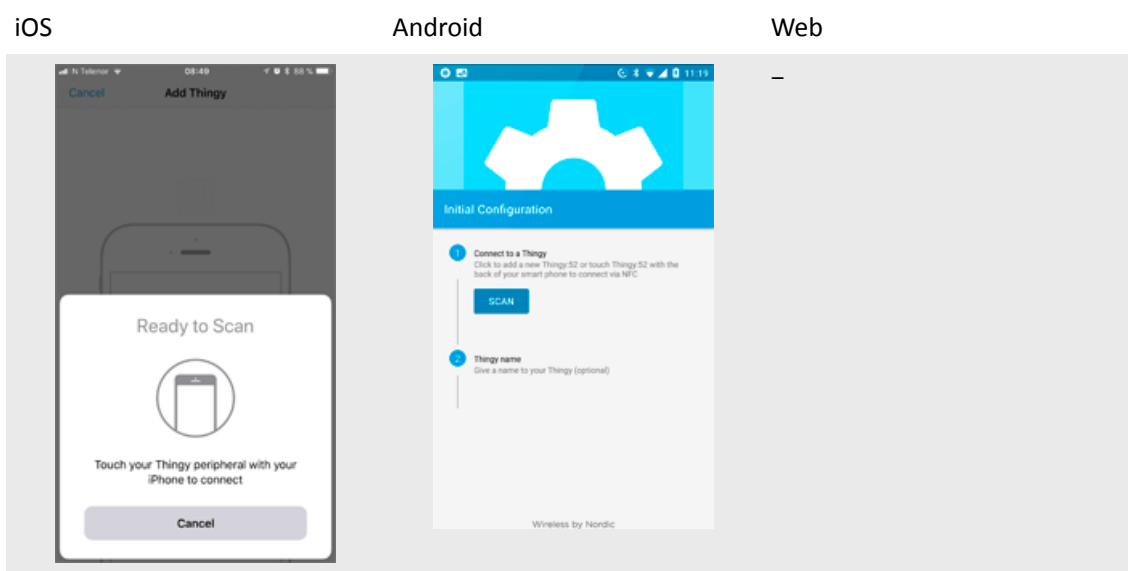
- iOS: Application version 1.2.0 or later.
- Android: Application version 1.3.0 or later.

Follow the steps below to connecting Thingy using NFC.

1. Add Thingy.



2. Tap the top of the Thingy with the NFC sensor of the mobile device.



After tapping, you will see a browser view in which you can add a new Thingy.

You will be notified that a Thingy with an NFC Tag has been found.

Continue the initial configuration as instructed in [Connecting Thingy using Bluetooth](#) on page 11.

4.1.3 Troubleshooting

If you have difficulties in connecting Thingy with your mobile device, there are a few possible causes to this.

If Thingy does not appear in the list within a couple of seconds, check the following:

No power

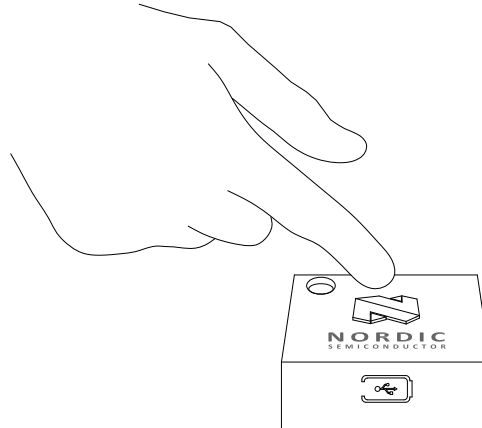
Use the power switch located under the black rubber cover, to the left of the USB port, to switch power on.

Battery empty

Connect Thingy to a PC or charger using a micro-USB to recharge the battery. Make sure that the power switch is on when charging.

Thingy is in sleep mode

Wake Thingy up either by pressing the Thingy button located right under the Nordic Semiconductor logo or by shaking the device. See [Low-power accelerometer](#) on page 40.





Thingy connected to another mobile device

By default, the LED will breathe in green color to indicate that it is already connected to another device.

4.2 Configuring Thingy

The Configuration service allows you to customize several settings on Thingy.

iOS:

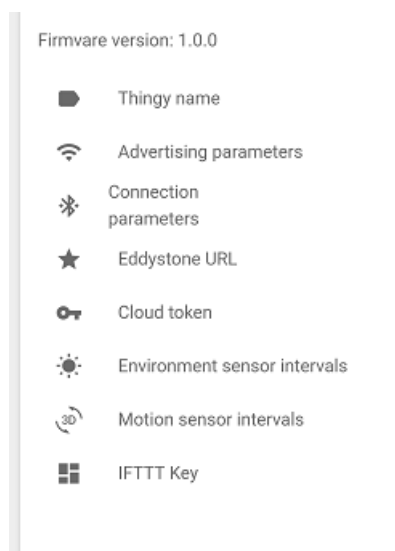
-  iOS: The Configuration button  is located to the right of the application's navigation bar.
- To modify a specific configuration, tap the cell you would like to change and modify the contents. You will see appropriate configuration types and value ranges, as well as have your input validated when you change a value. The new configuration is automatically saved.

Android:

- Basic tab: Configure the basic Thingy settings.
- Advanced tab: Configure the interval settings of the Thingy notification updates for Environment and Motion sensors. To modify a specific configuration, tap the cell you would like to change and modify the contents. You will see appropriate configuration types and value ranges, as well as have your input validated when you change a value. The new configuration is automatically saved.

Web

Configuration view:



4.2.1 Configurable settings

Thingy has a number of settings that can be customized to suit your needs.

Parameter	Description
Name	The name used when advertising the Thingy device. IOS and Android: The name is limited to 10 bytes of alphanumeric values and emojis. Alphanumeric values correspond to 1 byte, while most emojis account for 2 bytes.
Advertising parameters	
<ul style="list-style-type: none"> Advertising interval 	The frequency of Thingy advertising.
<ul style="list-style-type: none"> Advertising time-out 	The duration of Thingy advertising for a connection request before timing out.
Connection parameters	
<ul style="list-style-type: none"> Minimum connection interval 	The minimum interval in milliseconds at which the mobile device asks for data from Thingy.
<ul style="list-style-type: none"> Maximum connection interval 	The maximum interval in milliseconds at which the mobile device asks for data from Thingy.
<ul style="list-style-type: none"> Slave latency 	The number of events between the mobile device asking Thingy for data and Thingy actually sending the data. This allows Thingy to stay in low-power mode for a longer time when it has no new data for the mobile device.
<ul style="list-style-type: none"> Supervision time-out 	A time-out in seconds from the last data exchange before considering the Thingy link lost. If you expect that Thingy will often go in and out of range, it is better to have a short time-out to detect it quickly.

Parameter	Description
Eddystone Beacon URL	<p>URL in the Eddystone URI format.</p> <p>Thingy is equipped to act as an Eddystone URL beacon. During a nearby scan, the Thingy may pop up as a notification with the default nordicsemi URL. However, the URL can be changed in this configuration.</p> <p>iOS: Requires and Eddystone beacon app.</p> <p>Android: According to the Eddystone URL beacon specification, the URL cannot be longer than 20 bytes. To configure a URL longer than 20 bytes, tap Shorten URL and you will be provided with a shortened version of it.</p>
Cloud Token (IFTTT key)	<p>iOS and Android: A token for any cloud service.</p> <p>Web: Paste the IFTTT key in this field to integrate your Thingy with the IFTTT web service.</p>
Sensor intervals	<p>The interval of Thingy notification updates for sensors.</p> <p>Advanced configuration: In order to modify a specific configuration, tap the cell you would like to change and modify the contents. Here you will see appropriate configuration types and value ranges, as well as have your input validated when you change a value. The new configuration is automatically saved.</p>

Table 1: Configurable settings

Viewable parameter (iOS, Android and Web)

You can also view the current firmware version that is running on Thingy.

5 Services

Once you sync Thingy to the app, a set of services become available. You can access all of them from the main menu of the app.

The services listed here and marked with an "X" are available in the Thingy app. They do not reflect the exact service names or number on the device.

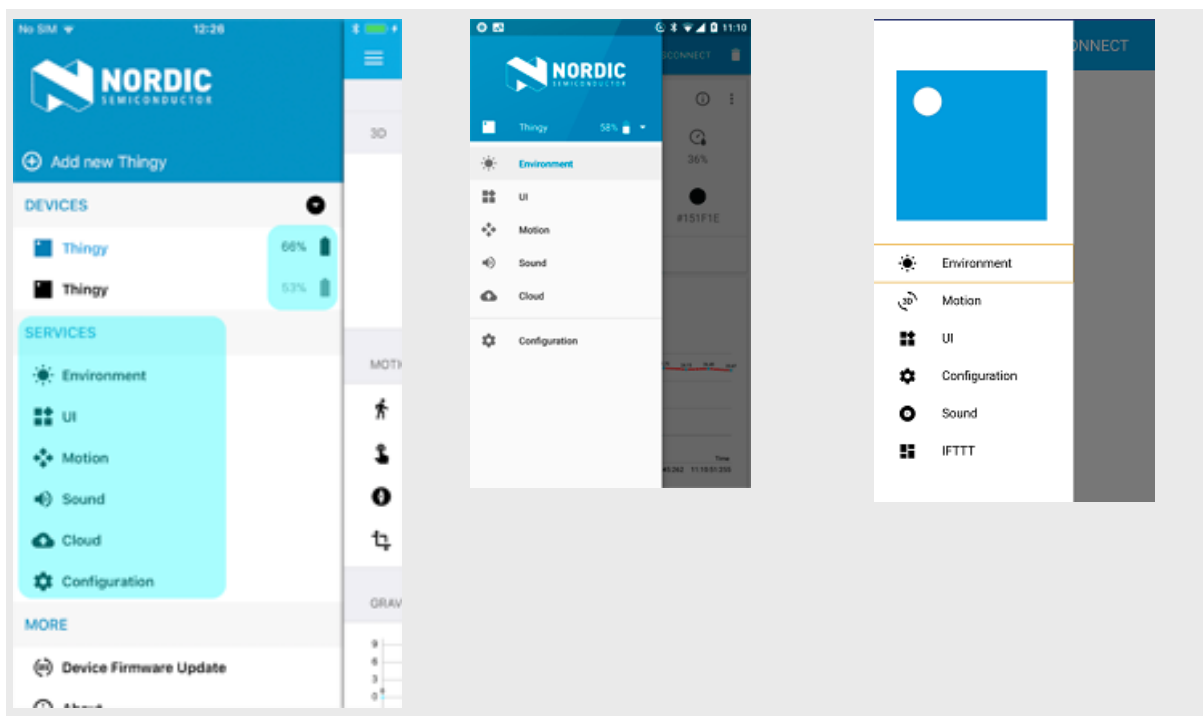
Service	iOS	Android	Web
Environment	X	X	X
User interface (UI)	X	X	X
Motion	X	X	X
Sound	X	X	X
Cloud (IFTTT)	X	X	X
Configuration	X	X	X
Battery	X	X	–

Table 2: Thingy services

iOS

Android

Web



A battery indicator will appear beside connected peripherals in the main menu with Thingy firmware that supports the battery service. The following application versions are required:

- iOS: Application v1.2.0 and later
- Android: Application v1.3.0 and later

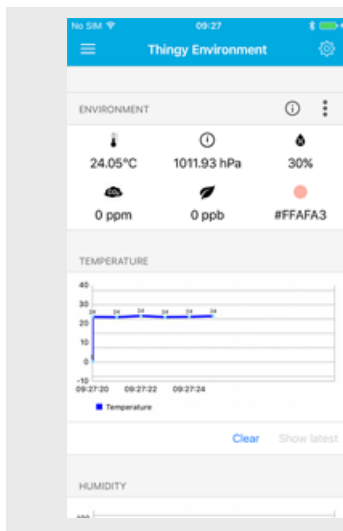
- Web: Not available

5.1 Environment

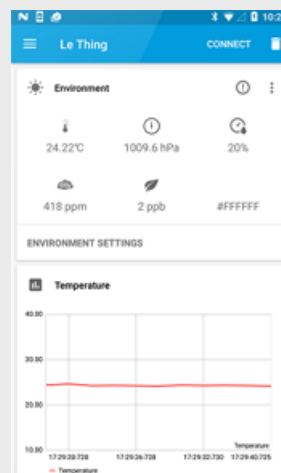
The Environment service provides you with a set of environmental data, such as temperature, air humidity, air pressure, and light intensity.

iOS and Android: When the app is first launched, it displays the **Thingy Environment** with all features enabled and notifying.

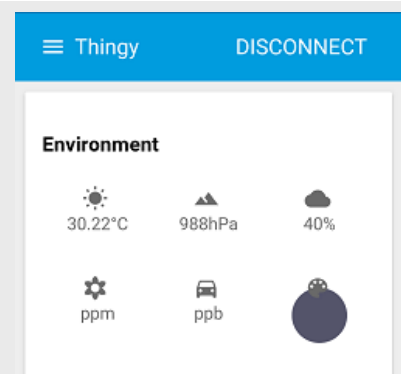
iOS



Android



Web



The Environment features available are the following:

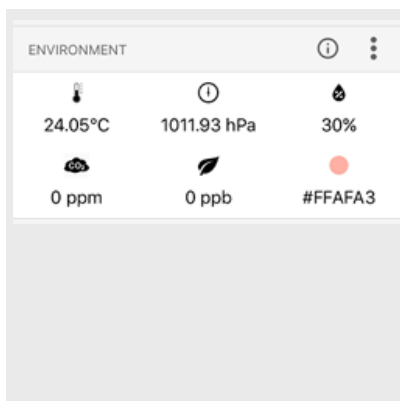
- Temperature
- Pressure
- Humidity
- Air quality:
 - CO₂
 - Total volatile organic compounds (TVOC)
- Light/color intensity

iOS and Android:

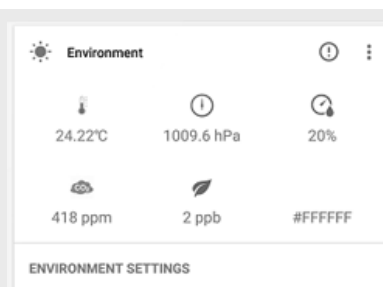
- Tap the Info button ⓘ for clarification on what each symbol stands for.
- Tap the More options button ⋮ to enable or disable the available features.

After you have customized a feature's setting, it is saved and will display the next time you launch the app. Then you can see the most current data for all enabled features in the Environment service.

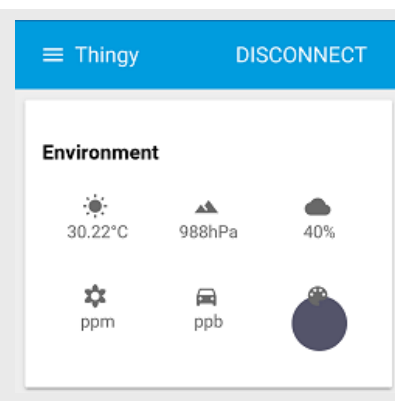
iOS



Android



Web



5.1.1 Live graphs

Live graphs are available for the temperature and pressure features for iOS, Android and Web, and they represent the current and historical temperature measurements. For iOS and Android, live graphs are also available for humidity features. Once enabled, data will start populating the graphs in realtime.

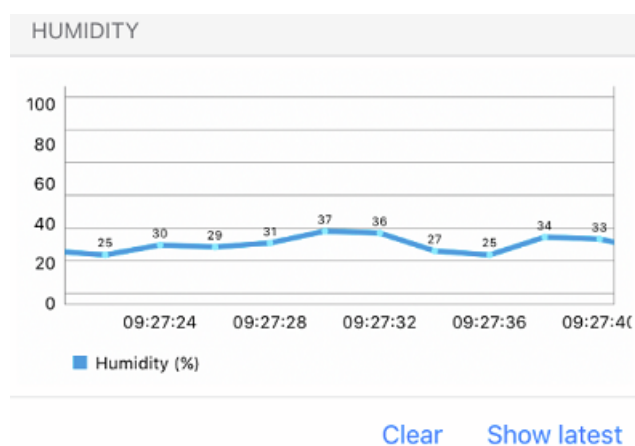
Here are some tips when interacting with a live graph.

iOS and Android:

- Graphs automatically scroll to the latest value. (iOS: With **Show latest** button disabled)
- Tap any part of the graph to see the timestamp and exact value of the entry.
- You can scroll through the graph manually, as with any other scrollable view.


iOS only:

- Tap and hold the graph to stop scrolling. The **Show latest** button will now be enabled.
- To resume autoscrolling, tap **Show latest**.
- To clear a graph, tap **Clear**. This will empty all data and will reenable autoscrolling.



5.1.2 Configurable settings (iOS and Android)

In the configuration menu, you can customize several aspects of the Environment service.

Tap the Configuration button  to the right of the application's navigation bar to open configuration settings.

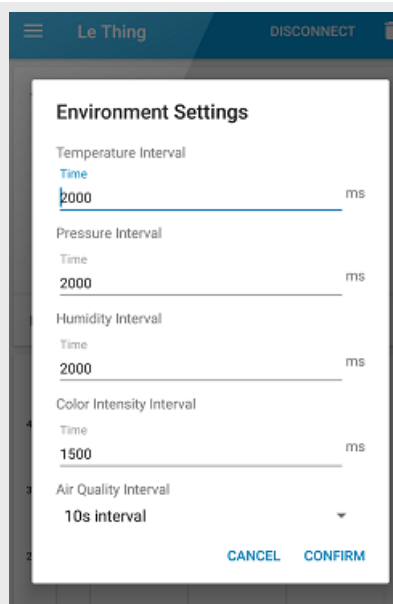
- Temperature sampling interval (ms)
- Pressure sampling interval (ms)

- Humidity sampling interval (ms)
- Light/Color intensity sampling interval (ms)
- Air quality sampling interval - This setting uses predefined values of 1, 10, or 60 seconds only.
- iOS only: Temperature Unit - Celsius or Fahrenheit.

iOS



Android



To change a setting, tap the feature that you would like to change and type in the new value.

Tap **Set** (iOS)/**Confirm** (Android) to save the changes.

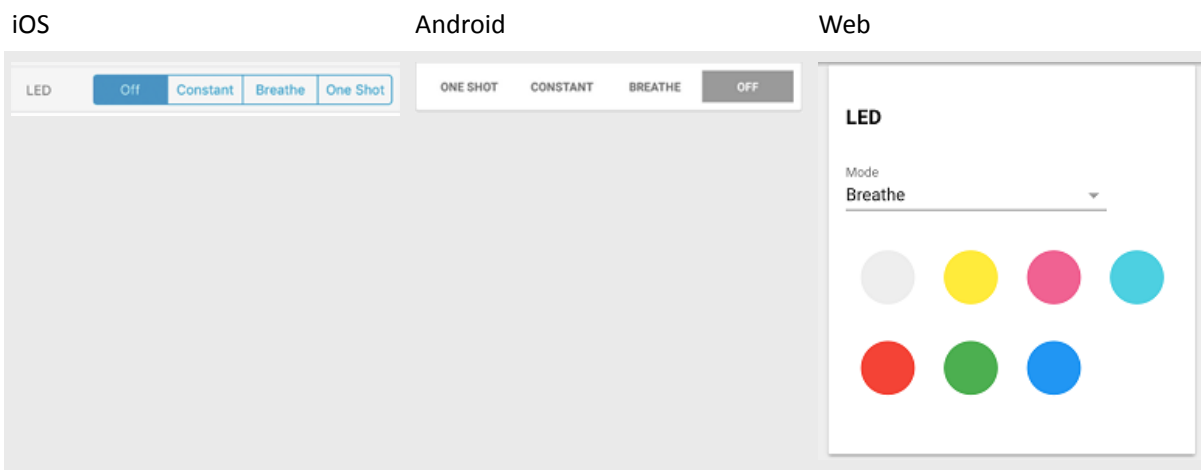
Tap **Cancel** to ignore your changes.

5.2 User interface

The User interface (UI) service allows you to interact with the LED and button on Thingy.

5.2.1 LED

Here you can change the properties of the Thingy LED. Use the selections at the top to control the mode of the LED. Only one mode can be active at a time.



- Off mode: The LED is turned off completely and Thingy will remain in Off mode while disconnected. There is no configuration for this mode.
- Constant mode: The LED stays on, but once disconnected the mode turns off. For iOS and Android, there are red, green, and blue value sliders that allow you to change the color of the LED.
- Breathe mode: The LED fades in and out at a given interval, but turns off after Thingy is disconnected. For iOS and Android, you can choose from one of the preset colors, adjust the light intensity, and increase or decrease the breathe delay.
- One-shot mode: The LED fades in then out once, then remains off indefinitely. For iOS and Android, you can select the color of the LED and adjust the light intensity in the settings. This mode does not persist while disconnected.

5.2.2 Button

In the Button section, you can see the current state of the Thingy button.

- Unknown: Displayed when the view is initially loaded and the app has no information about the button's state. (iOS and Android)
- Pressed: Displayed when you press and hold the button.
- Released: Displayed when you release the button.

5.3 Motion

The Motion service contains the pedometer, tap sensor, accelerometer, and other motion-related sensor data received from Thingy. Here you can see information related to 3D motion, motion characteristics, and the gravity vector.

The Motion features come from all motion-related sensors.

- Step counter/Pedometer: A standard pedometer that shows step count and duration.
- Tap sensor: A sensor that counts the taps and their direction axis on Thingy.
- Heading: A compass showing the heading angles.
- Orientation: Displays the orientation of Thingy (portrait, landscape, etc.)

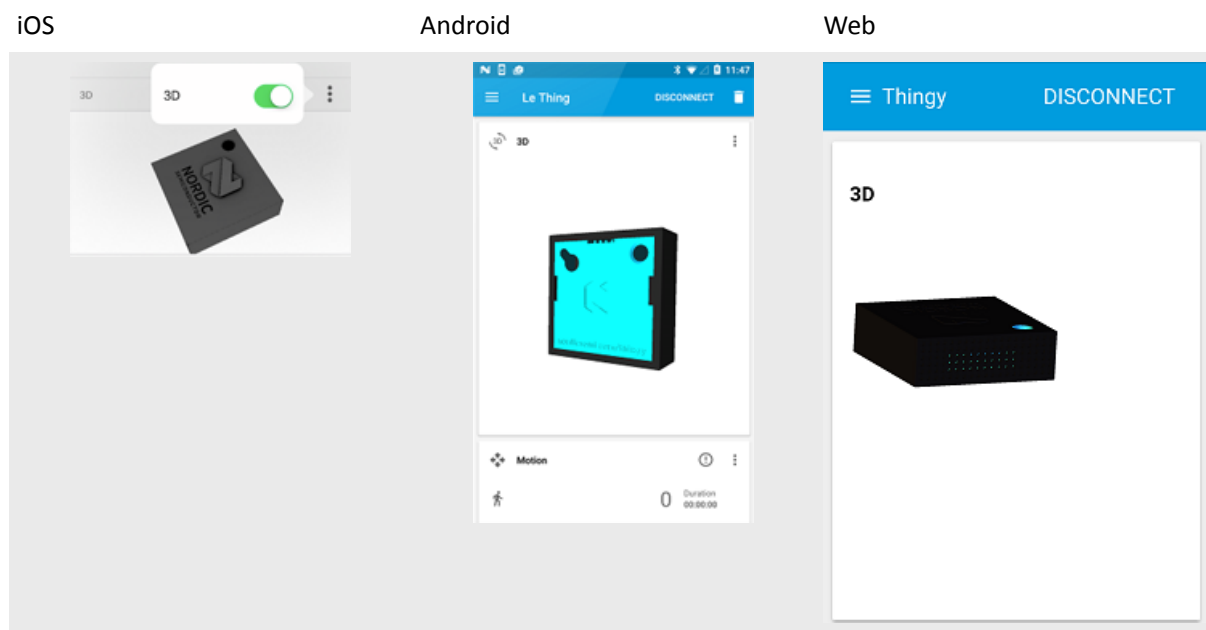
iOS and Android:

Tap the Info button ⓘ to display what the icons represent.


Tap the Action button ⋮ on the Motion section to enable or disable different characteristics by selecting or deselecting a feature.

5.3.1 3D Motion

3D Motion simulates the current rotation attitude of Thingy on a 3D model in the app. Rotating Thingy on any of the axes will immediately be reflected onto the simulation.




For iOS and Android, follow these steps to interact with this feature:

1. Tap the More options (iOS)/Action (Android) icon  in the 3D section.
2. Toggle the 3D switch. The model immediately starts simulating Thingy.

The iOS and Android simulations use quaternions, and the Web simulation uses Euler angles to rotate the model at a given configured interval. To make the animation smooth, the simulation interpolates between the current and target positions with a duration of the update interval. This makes the animation as smooth as possible.

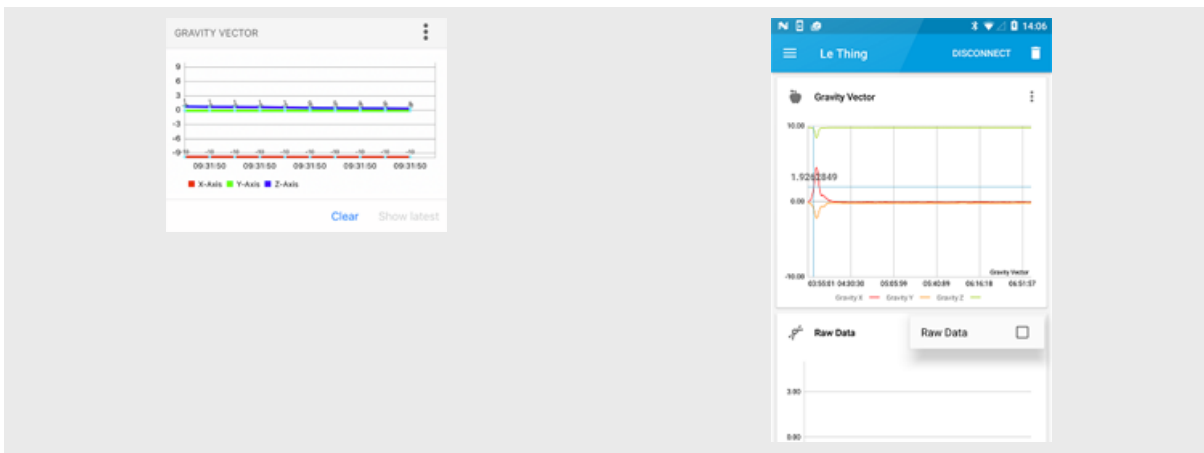
5.3.2 Gravity vector (iOS and Android)

The Gravity vector section displays the effect of gravity on each axis in units of m/s^2 .

To use this feature, tap the Action/More options icon  and use the toggle to enable it. The live graph will start displaying data. You can move Thingy around to see how gravity affects it.

iOS

Android




You can interact with the graph in the following way:

- The graph will autoscroll when you first open it.
- Tap a point on the graph to view its value.
- Tap and hold to stop the graph from autoscrolling. You can now manually scroll the graph back and forth.

For iOS, the following also applies:

- When there are new values available, the **Show latest** button will be enabled.
 - Tap the button to scroll the graph to the last entry. This will also reenable autoscrolling.
- Tap **Clear** to clear all entry points and reenable autoscrolling.

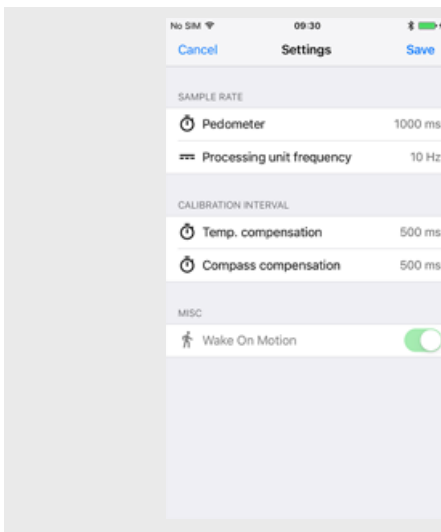
5.3.3 Configurable settings (iOS and Android)

The Configuration button  is located to the right of the application's navigation bar. In the Configuration view, you can customize several aspects of the Motion service.

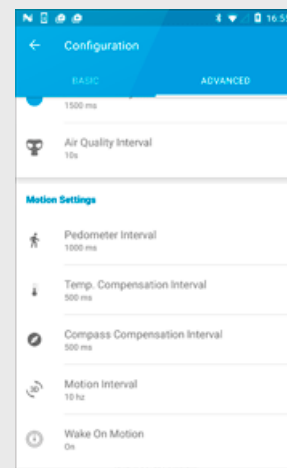
The configurable settings are:

- Pedometer interval (ms)
- Processing unit frequency/Motion interval (Hz)
- Temperature compensation interval (ms): Used for internal calibration of the sensor.
- Compass compensation interval (ms): Used for internal calibration of the sensor.
- Wake On Motion: When enabled, Thingy will wake up from low-power mode when moved.

iOS: Settings



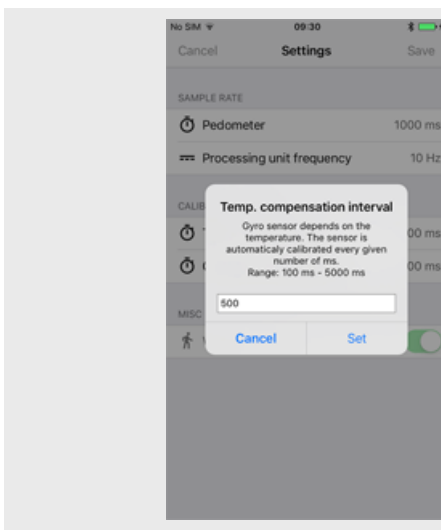
Android: Configuration



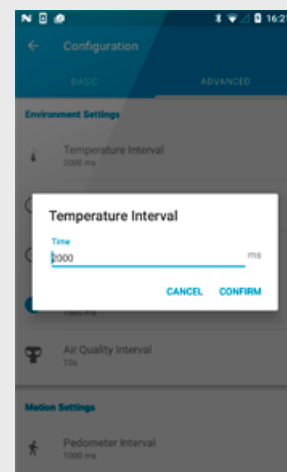
To change a setting:

1. Tap the value you would like to change.
2. Type the new value into the field.
3. Tap **Set/Confirm** to save the new value.
4. Tap **Cancel** to ignore your changes.

iOS



Android



5.4 Sound

The Sound service lets you control Thingy's microphone (iOS and Android) and speaker.

The Thingy Sound service is based on two building blocks, the speaker and the microphone, which can work in specific modes.

Speaker modes:

- 8-bit PCM (streaming)
- Frequency and duration (piano keyboard)
- Sample (sample sound effects)

If a sound sample or keyboard input is detected while any of these modes are running, the current mode will quit and the new mode will start. The speaker can only function in one mode at a time.

Microphone mode (iOS and Android):

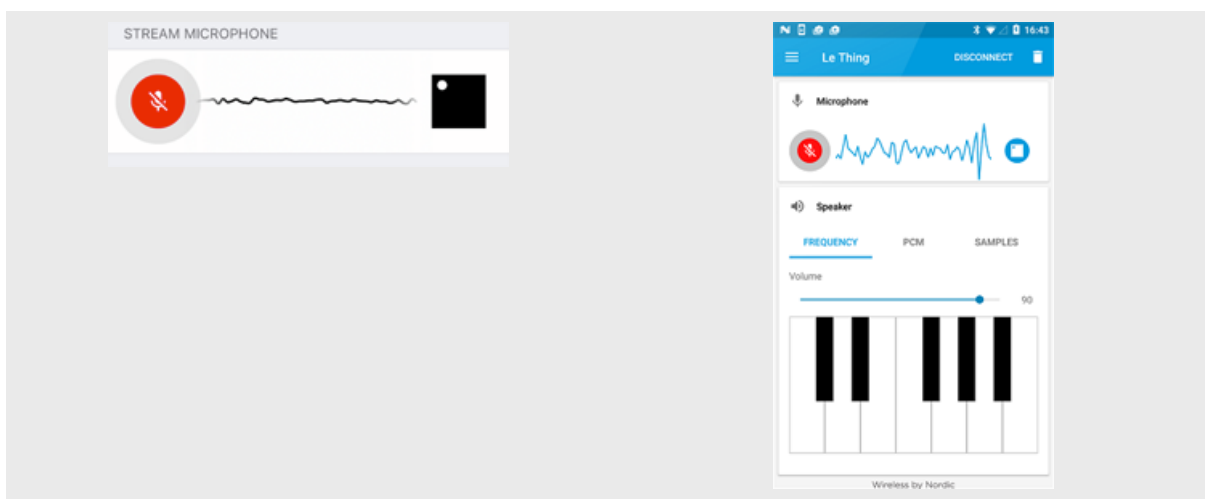
- ADPCM mode (only supported mode within the app)

5.4.1 Streaming: Microphone (iOS/Android)

Stream microphone (iOS)/Microphone (Android) allows controlling the streaming process from Thingy to the mobile device and vice versa.

iOS

Android



5.4.1.1 Streaming from mobile to Thingy

Follow these steps to stream sound from the microphone on a mobile device to the Thingy speaker. This method applies to both iOS and Android.

1. Tap the microphone icon on the left.

If the microphone permission is not granted yet, you will receive a permission request.

If the permission was declined, the mobile device might not ask again, leaving this feature useless.

- a) Optional: To manually grant the permission, go to the app settings, and toggle the permissions for the Thingy app in the privacy menu.

The microphone icon will turn red and the visualization will display the current audio being streamed.

2. To stop streaming, tap the microphone button again.
The button will turn black and the graph will stop updating.

5.4.1.2 Streaming from Thingy to mobile

Follow these steps to stream sound from the Thingy's microphone to a mobile device's speaker. This method applies to both iOS and Android.

Streaming from the Thingy to the mobile device requires no special permissions.

1. Tap the Thingy icon on the right.
The Thingy icon will turn red, and the visualizer will display the sound wave coming from Thingy. You should hear audio playing through the mobile device's speakers.
2. To stop streaming, tap the Thingy icon.
The icon will turn black and the graph will stop updating.

5.4.1.3 Streaming from Thingy to mobile (iOS alternative)

For iOS, this is an alternative method for streaming sound from the Thingy's microphone to the mobile device's speaker.

Streaming from the Thingy to the mobile device requires no special permissions.

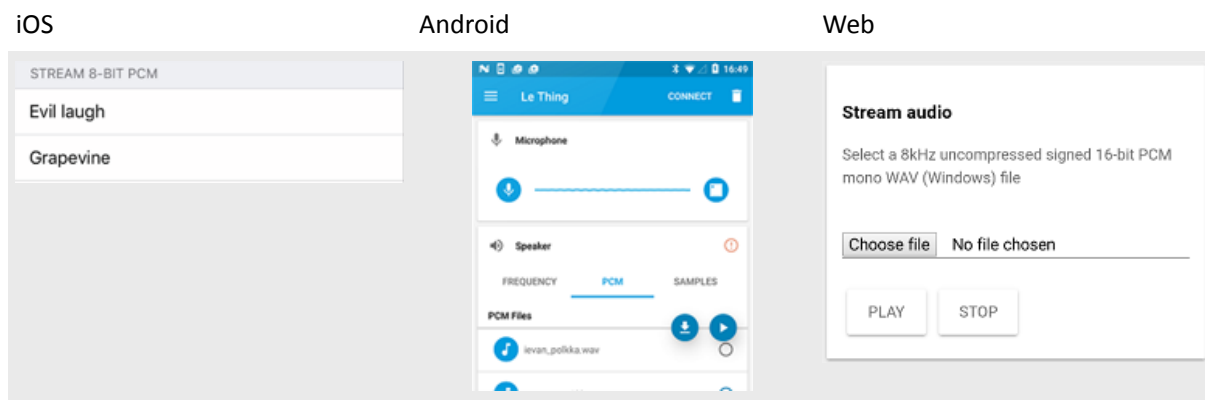
1. Navigate to the Sound view and open it.
This type of streaming will only work when the Sound view is open.
2. Press the Thingy button located on the side of the casing (not the one in the app).
3. To stop streaming, press the Thingy button again.

5.4.2 Streaming: Speaker

There are three audio streaming modes.

5.4.2.1 8-bit PCM streaming

Thingy is capable of streaming 8-bit PCM audio from the mobile device to its speaker. Select one of the example audio files to stream it to Thingy.



iOS:

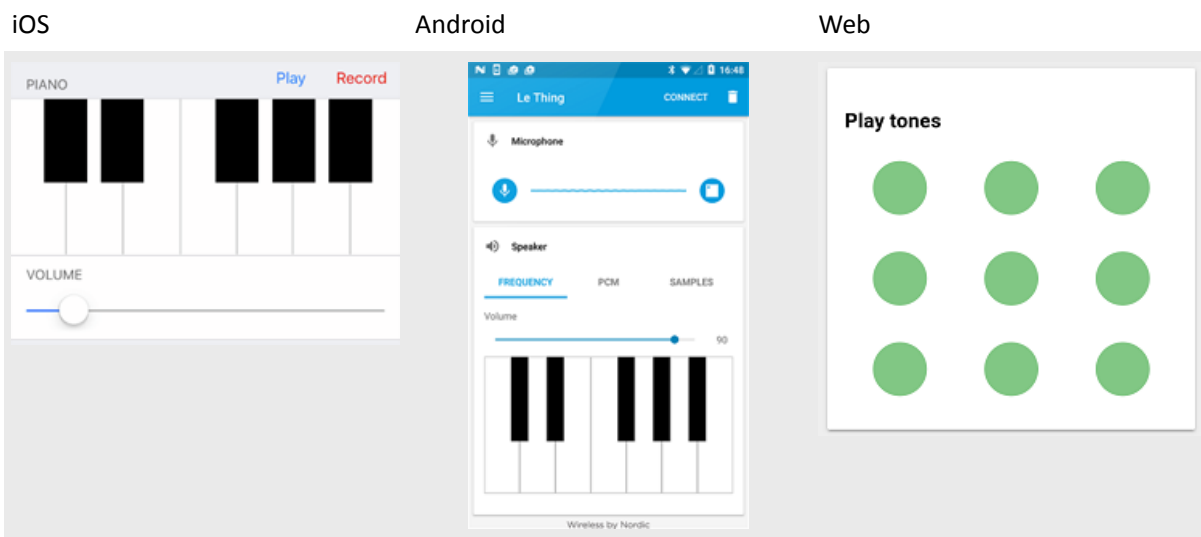
This feature only works properly on iPhone 7 and iPhone 7+ devices with iOS 10 and later, because they support the Data Length Extension (DLE) feature. Older devices do not have this feature and audio streaming will not work properly (lots of chipping and skipping), because they are not fast enough to keep Thingy streaming fully buffered.

Android:

Older or low-end devices running Android version earlier than Lollipop may encounter interruptions with audio streaming because of lower MTU sizes. If you notice interruptions during audio streaming, try lowering the maximum connection interval in the Thingy app configuration. However, this might not solve the problem, due to hardware limitations. A mobile device can only stream to one Thingy at a given time due to bandwidth limitations.

5.4.2.2 Frequency mode

This mode simulates a piano keyboard that plays at different frequencies. Only one note can be played at a time.



To play a note, tap the respective piano key or tone button.

iOS: Record and Play

Record and **Play** buttons are available to store user input and replay these commands on Thingy. No actual audio is recorded in that process, just key strokes and durations.

To record input for later playback, press the **Record** button, play some notes, and press **Stop** to store the keystrokes.

Whenever there is recorded data, the **Play** button will appear next to the **Record** button.


To start playing back the recorded keystrokes, press **Play**. Playback will stop automatically, or when you press **Stop**.

iOS: Volume control

Use the volume slider to control the loudness of the sound on Thingy. This slider only affects the notes played on the piano keyboard and not any other sound feature on the device.

5.4.2.3 Sound samples

Thingy comes bundled with short sample sound effects that are stored in its firmware. Note that this is not actually streaming, as the audio files are on the device already.


- iOS: In the **Sample sound** view, tap any of the samples. The sound will be immediately played.
- Android: Tap buttons 1–9 to directly play the sound samples stored in Thingy.
- Web: In the **Play samples** view, tap the  icons to directly play the sound samples stored in Thingy.

Only one sample can be played at a time. Playing any other sample will stop the current sample.


5.5 Cloud (IFTTT)

The Cloud service lets you control the integration of the Thingy app with the IFTTT (If This Then That) web service.

Before you can start using this service, you must register with IFTTT:

- iOS and Android: tap the Info icon  in the upper right corner of the Cloud service view.
- Go to the IFTTT website.

You must obtain the Maker Webhooks Token (iOS/web)/Cloud Token (Android) that is necessary for IFTTT integration. Paste this token in the following location:

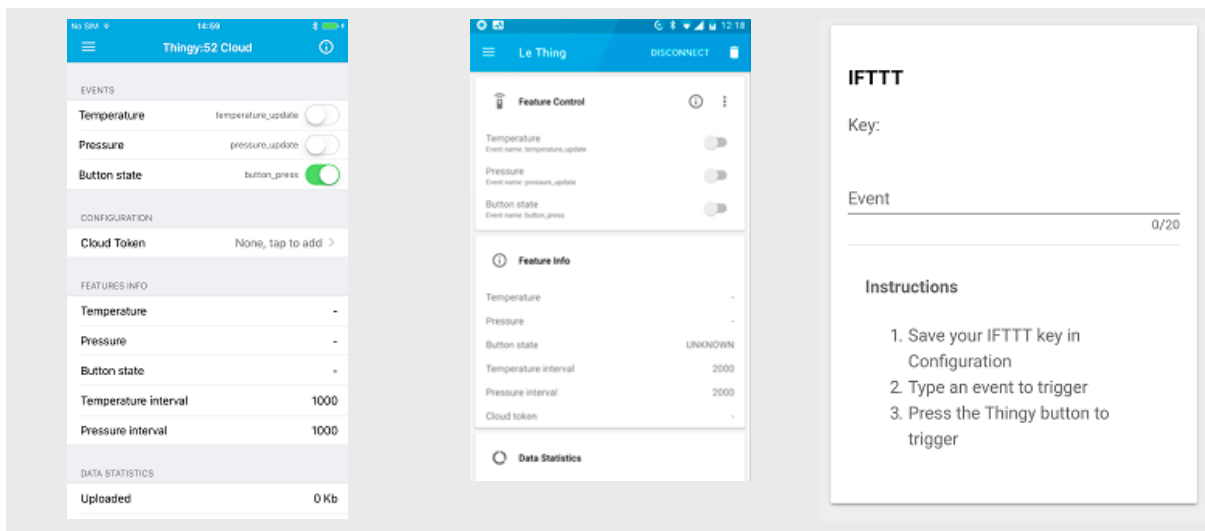
- iOS: **Cloud Token**
- Android: **Add/Edit IFTTT token** available under the Action  icon)
- Web: [Configuration view](#)

Once you have the key, you can use the Thingy button as an event that will be used for triggering IFTTT conditional statements.

iOS

Android

Web



Events (iOS)/Feature Control (Android)

Use the sliders to enable or disable the events that will be used for triggering IFTTT conditional statements. The available events are **Temperature**, **Pressure**, and **Button state**.

Configuration (iOS)

Paste the Cloud Token in this field.

Feature(s) Info

If a particular type of event from the **Events/Feature Control** section is enabled, you can see its status here. You can also check the interval at which temperature and pressure are measured.

Data Statistics

The two fields in this section show the amount of data uploaded to and downloaded from the cloud, in Kb (iOS)/B (Android).

6 Managing Thingy devices

You can add, remove, connect, or disconnect Thingy devices, and update their firmware using the mobile app.

The iOS and Android mobile applications support connections to multiple Thingy devices, meaning you can have more than one Thingy configured in the app.

6.1 iOS

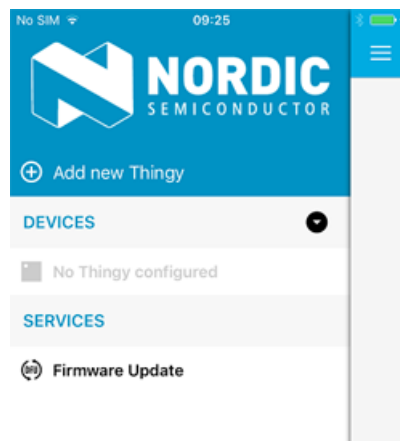
You can add, remove, connect and disconnect a Thingy.

6.1.1 Adding a Thingy

You can add a Thingy in the main application menu.

You can either add a Thingy using *Bluetooth* scanning or the NFC tap-to-connect function enabled in firmware 2.0.0 and later. The steps below show the *Bluetooth* scanning method. For the NFC method, see [Connecting Thingy using NFC](#) on page 12.

1. Open the main application menu from the top left side of the screen.



2. Tap **Add new Thingy**

The Scanner view appears and automatically starts scanning.



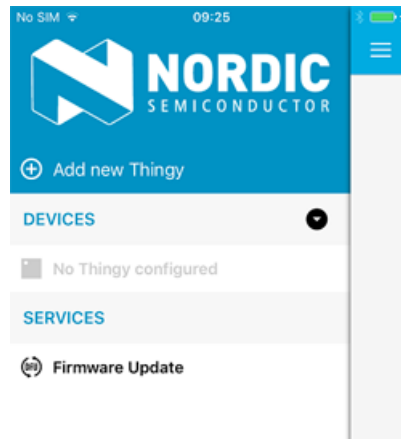
3. Once the Thingy device is discovered, select it from the menu and follow the wizard.

The procedure is the same as during the first launch. See [Connecting Thingy](#) on page 11.

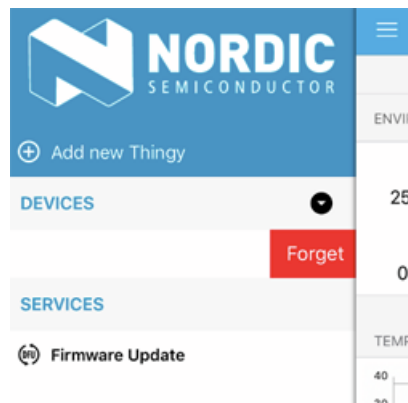
6.1.2 Removing a Thingy

You can remove a Thingy in the main application menu.

1. Open the main application menu from the top left side of the screen.



2. Swipe left on the Thingy that you want to remove.
A **Forget** button appears.
3. Tap the **Forget** button.

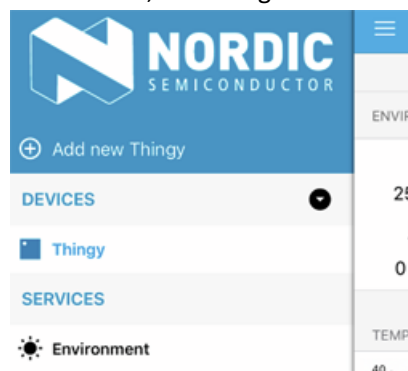


Thingy will be removed. If the device was connected, it will be disconnected first.

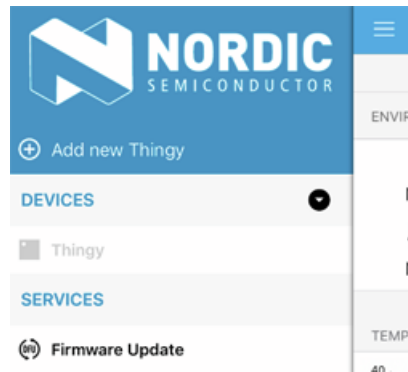
6.1.3 Connecting and disconnecting a Thingy

The Thingy app supports configuring and storing more than one Thingy. Only one Thingy can be active at a time.

- To connect a disconnected Thingy, tap it in the main menu.
The device will connect and its icon turns blue, indicating it is connected.



- To disconnect a Thingy, tap a connected Thingy in the main menu.
The device will disconnect and its icon turns grey, indicating that it is not connected.



- If one Thingy is connected, and the user selects another peripheral as the active one, the currently connected peripheral will be marked as inactive, but it will remain connected.


6.2 Android

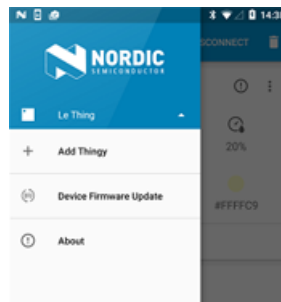
You can use add, remove, connect and disconnect a Thingy.

6.2.1 Adding a Thingy

You can either add a Thingy using *Bluetooth* or the NFC tap-to-connect function enabled in firmware 2.0.0 and later.

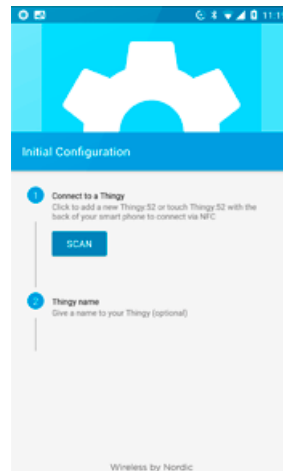
The steps below show the *Bluetooth* method. For the NFC method, see [Connecting Thingy using NFC](#) on page 12.

1. Open the main application menu from the top left of the screen.
2. To switch to the menu section that allows for device management, tap the arrow  next to the Thingy name.
3. Tap **Add Thingy**.



4. To scan for a nearby Thingy, tap **Scan** in the Initial Configuration.


Once you have discovered the one you want, you can connect to it, add a name, and share the location. The procedure is the same as during the first launch. See [Connecting Thingy](#) on page 11. You can tap X any time to cancel the process.

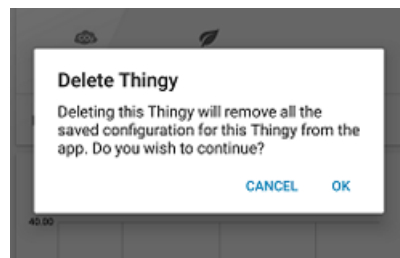


Once you have added a Thingy to your app, the main menu will contain a new device. The currently selected Thingy is always shown at the top of the list.

6.2.2 Removing a Thingy

You can remove a Thingy in the main application menu.

1. In any of the service views, tap the trash icon  located on the toolbar.
2. Tap **OK** when prompted to confirm your selection.



6.2.3 Connecting and disconnecting a Thingy

Only one Thingy can be active at a time. The connectivity state of the currently selected Thingy can be determined by the **Connect/Disconnect** button in the top toolbar. A connected device will feature a **Disconnect** button, while a disconnected Thingy will have a **Connect** option.

- To connect a Thingy device, select it from the main menu and tap **Connect** in the top toolbar. The device will connect and its icon turns blue, indicating it is connected.
- To disconnect a Thingy device, select it from the main menu and tap **Disconnect** in the top toolbar. The device will disconnect and its icon turns grey, indicating that it is not connected.
- If one Thingy is connected, and the user selects another Thingy as the active one, the currently connected device will be marked as inactive, but it will remain connected.

6.3 Over-the-air device firmware update

The mobile apps provide support for over-the-air device firmware updates (OTA-DFU). These allow for seamless updates to the Thingy firmware application through the Thingy mobile application.

When there is a new feature or a bug fix in the Thingy firmware, you can use the mobile application to connect to Thingy and replace the current firmware with the latest firmware application.

In this view, you can see the name of the firmware package file, as well as its type, size, and version number. In addition, the currently connected device name is also displayed in the view.

Important: Currently, the iOS app contains a bundled firmware package with every release that is hardcoded within the app.

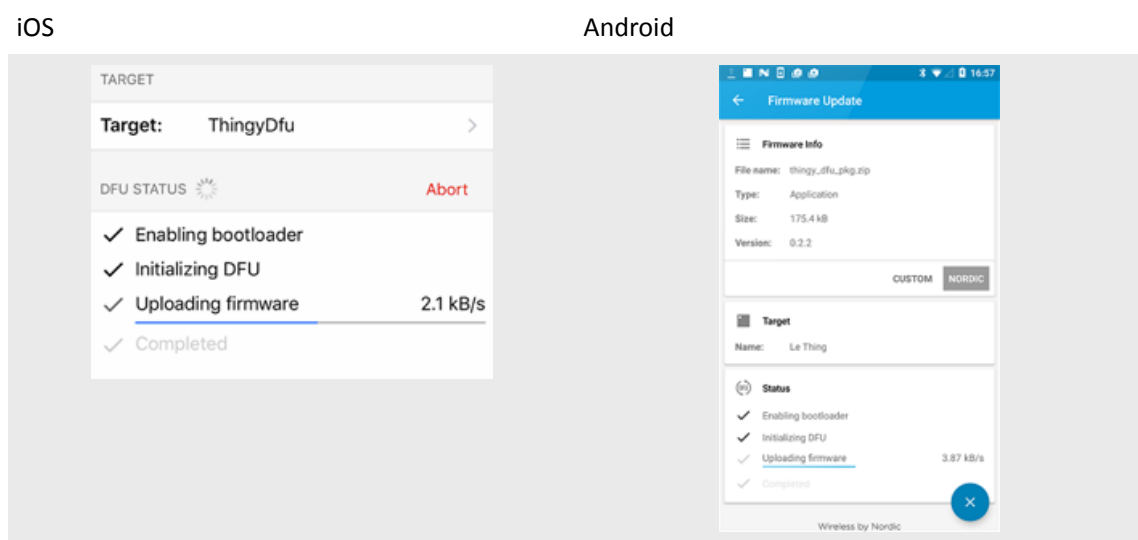
6.3.1 Updating firmware on a connected Thingy

This scenario is a default use case.

To update firmware on a connected Thingy, follow the steps below.

1. Connect to Thingy.
2. Navigate to the **Firmware Update** menu.
3. To start the DFU process:
 - iOS: Tap **Start** in the top right corner of the navigation bar.
 - Android: Tap the blue DFU icon in the bottom right corner of the screen to start the DFU process

The status of the DFU is highlighted on the tick marks at the bottom of the screen, and the progress bar is updated with the average DFU speed.



In this mode, the application tries to reconnect to Thingy once the DFU process is completed. However, if this fails, you can always go back to any of the main views and from there you can connect to Thingy as usual. For iOS, see [Connecting and disconnecting a Thingy](#) on page 30 and for Android, see [Connecting and disconnecting a Thingy](#) on page 32.

Important: (Android) Currently, this DFU procedure uploads a firmware package that is bundled and hardcoded within the mobile application. For future firmware updates, you will be notified of new available firmware through the app. This will allow you to download the latest firmware and flash Thingy with it. Selecting custom firmware is not supported but is planned as a future update.

During the DFU update, you can pause and resume the DFU update from the point when it was stopped.

6.3.2 Updating firmware on a Thingy in bootloader mode

This scenario is not a default use case.

To update firmware on a Thingy that is in bootloader mode, follow the steps below.

1. To put Thingy into bootloader mode, power it off and then on while holding the button on top of Thingy

The device starts advertising with the new name "ThingyDFU". The MAC address is always incremented by 1 when a Thingy device is in bootloader mode.

2. To enter the Scanner view if Thingy is in bootloader mode and not connected yet:

- iOS: Tap **Target**.
- Android: Tap the blue DFU icon.

All Thingy devices that are in bootloader mode are displayed in the list. Their default name is "ThingyDfu".

3. Select the device on which you want to run DFU. The process is initiated in the same manner as in the default method.

Important: When Thingy is in bootloader mode, it is impossible to connect to it using the **Connect** button in the main menu of the app.

During the DFU update, you can pause and resume the DFU update from the point when it was stopped.

7 Hardware description

This chapter focuses on the hardware components of Nordic Thingy:52 with detailed descriptions of the various hardware blocks that are present on the device.

The sensors available in Thingy are not calibrated in production. Nordic Semiconductor does not specify the accuracy of measurements. Users who want to reuse parts of this design to create measurement devices should conform to documentation of the specific sensors.

7.1 Block diagram

The following block diagram represents interactions between hardware components on Thingy.

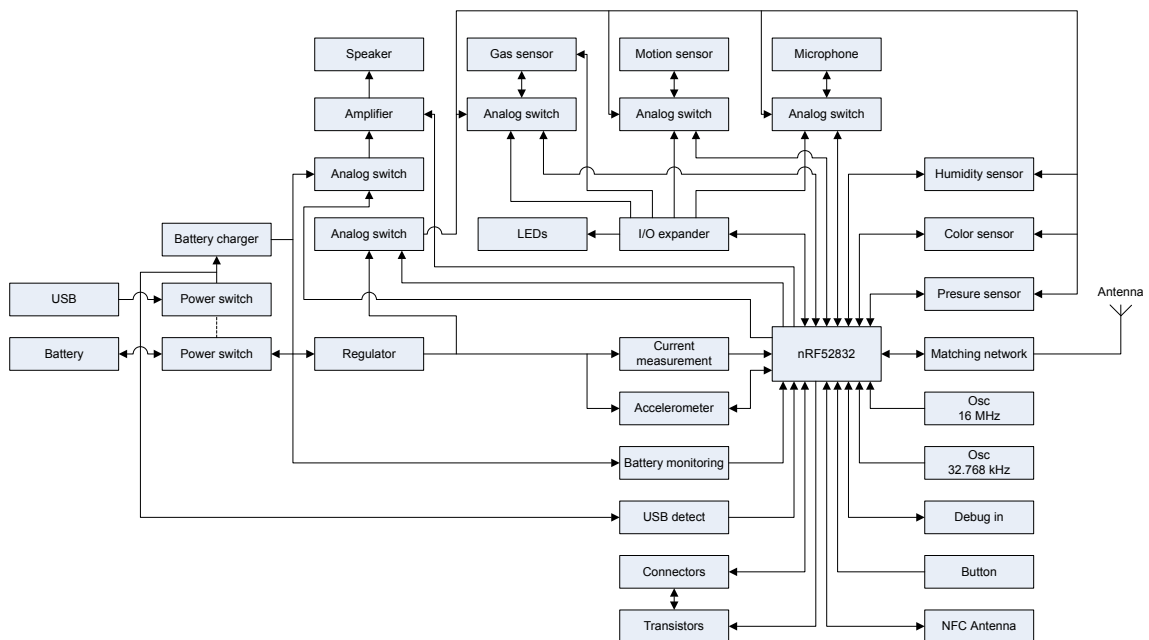


Figure 2: Thingy hardware block diagram

7.2 Hardware figures

The hardware figures show various elements on both sides of the Thingy PCB.

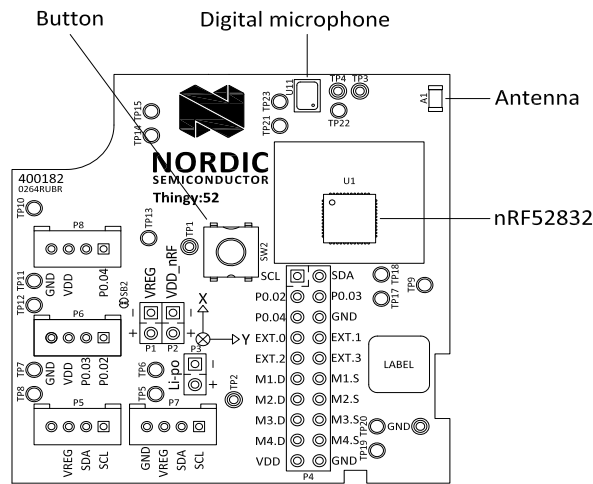


Figure 3: Thingy PCB, top

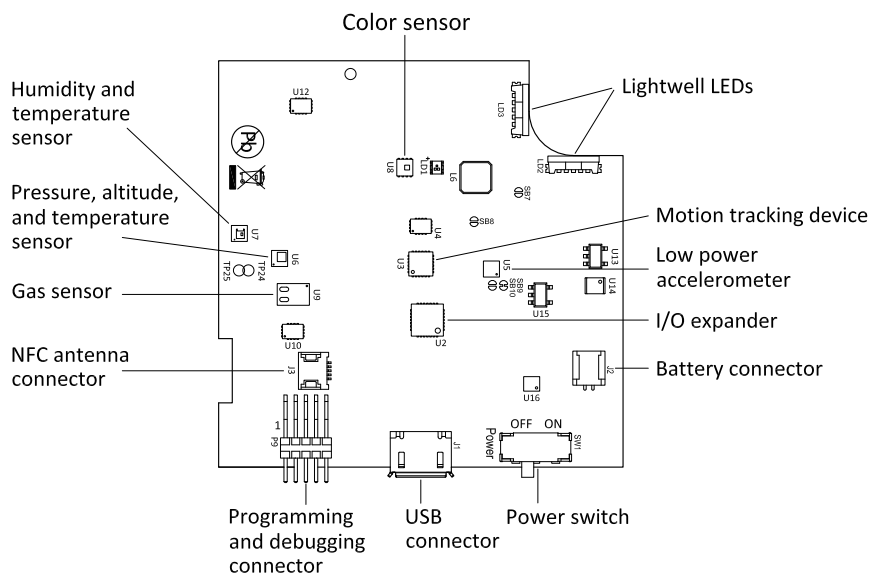


Figure 4: Thingy PCB, bottom

7.3 MCU

The nRF52832 SoC functions as the brain of Thingy. It is a powerful, highly flexible, ultra-low power SoC that incorporates a *Bluetooth* low energy radio and a 32-bit ARM[®] Cortex[®]-M4F CPU.

For more information on the SoC, see [nRF52832 Product Specification](#).

MCU and antenna

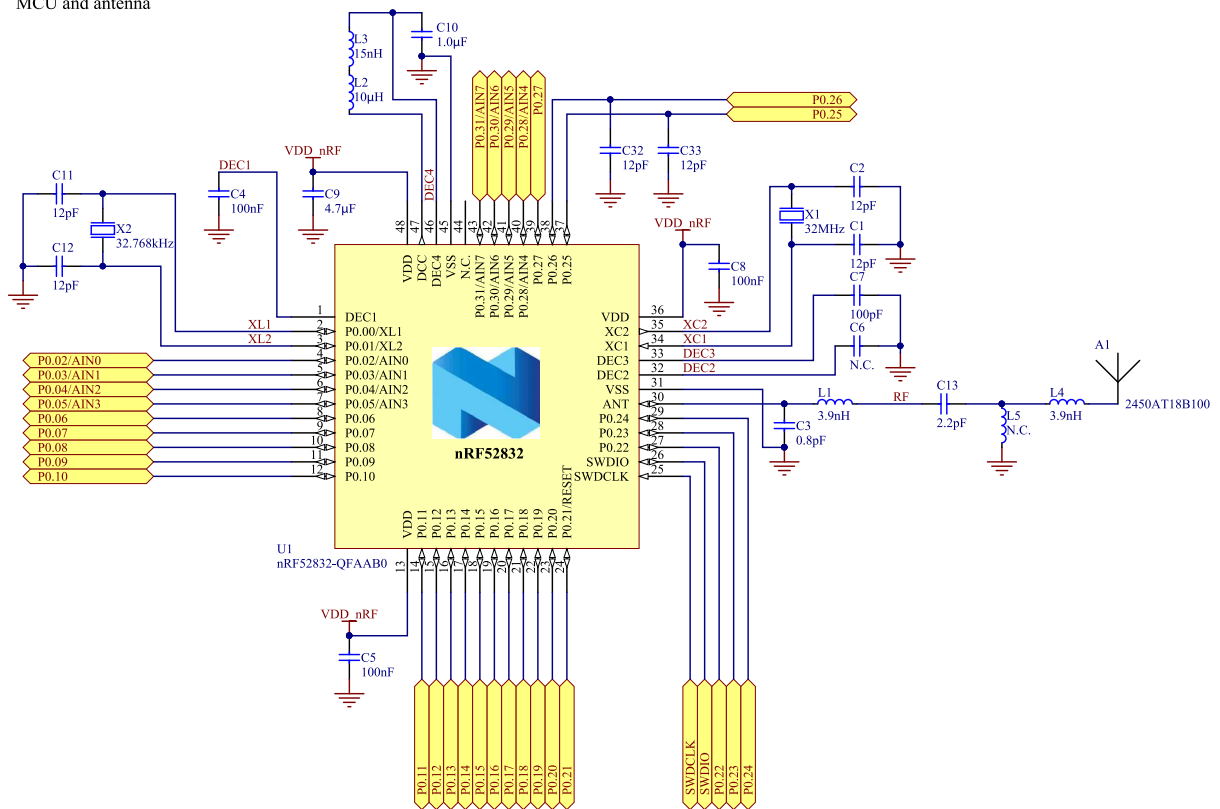


Figure 5: MCU schematic

7.4 I/O expander

Because of the high hardware complexity of Thingy, the design requires additional GPIOs beyond the 32 GPIOs of the nRF52832 SoC.

To increase the number of GPIOs, a 16-channel, ultra-low power I/O expander is connected to the I²C bus. The I²C slave address for the I/O expander is 0x3E. Most of the GPIOs on the expander are used internally on Thingy, but some of the GPIOs are made available on a connector for the user to connect additional hardware. The I/O expander has an integrated LED driver that supports intensity control, blink control, and breathing control.

I/O Expander I2C address: 0x3E

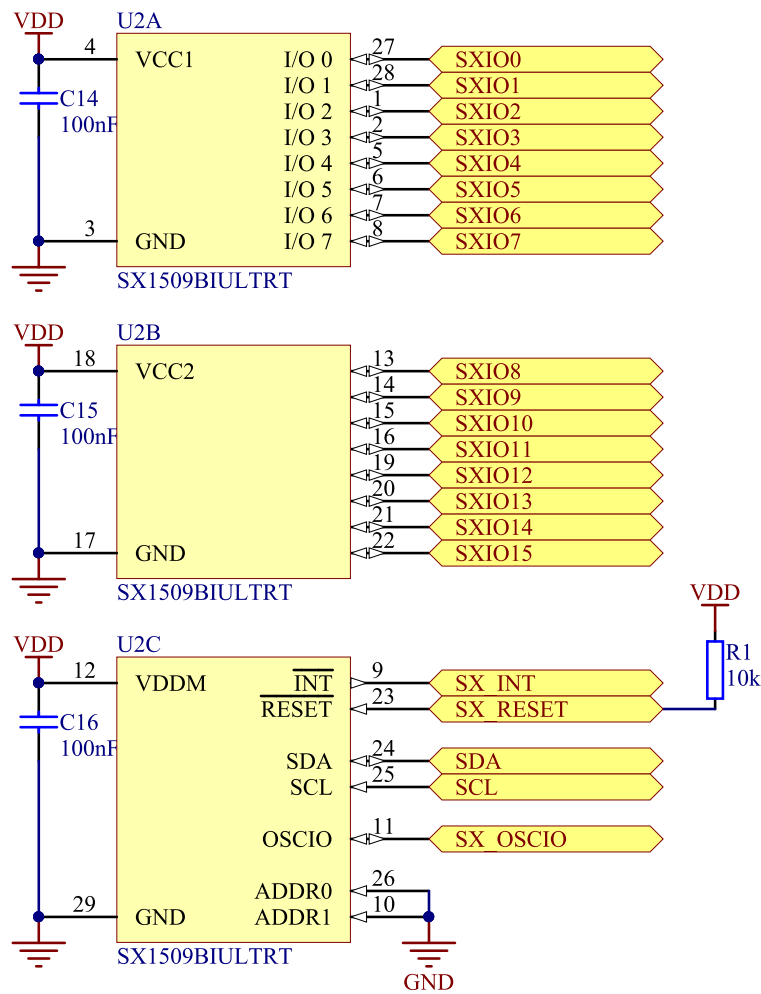


Figure 6: I/O expander schematics

7.5 Pin maps

The pin assignments for the nRF52832 SoC and for the I/O expander are listed in tables.

I/O	Label	Description
P0.00	XL1	Low frequency crystal
P0.01	XL2	Low frequency crystal
P0.02	ANA/DIG0	Analog/Digital GPIO externally available
P0.03	ANA/DIG1	Analog/Digital GPIO externally available
P0.04	ANA/DIG2	Analog/Digital GPIO externally available
P0.05	SX_OSCIO	I/O expander oscillator input line
P0.06	MPU_INT	Motion sensor interrupt line

I/O	Label	Description
P0.07	SDA	I ² C data line
P0.08	SCL	I ² C clock line
P0.09	NFC1	Near field communication antenna
P0.10	NFC2	Near field communication antenna
P0.11	BUTTON	Button input
P0.12	LIS_INT1	Low power accelerometer interrupt line
P0.13	USB_DETECT	USB detect signal
P0.14	SDA_EXT	External and low power accelerometer I ² C data line
P0.15	SCL_EXT	External and low power accelerometer I ² C clock line
P0.16	SX_RESET	I/O expander reset line
P0.17	BAT_CHG_STAT	Battery charge status
P0.18	MOS_1	Gate of N-MOS transistor externally available
P0.19	MOS_2	Gate of N-MOS transistor externally available
P0.20	MOS_3	Gate of N-MOS transistor externally available
P0.21	MOS_4	Gate of N-MOS transistor externally available
P0.22	CCS_INT	Gas sensor interrupt line
P0.23	LPS_INT	Pressure sensor interrupt line
P0.24	HTS_INT	Humidity sensor interrupt line
P0.25	MIC_DOUT	Microphone PDM data
P0.26	MIC_CLK	Microphone PDM clock
P0.27	SPEAKER	Speaker PWM signal
P0.28	BATTERY	Battery monitoring input
P0.29	SPK_PWR_CTRL	Speaker amplifier power control
P0.30	VDD_PWD_CTRL	Power control for sensors, I/O expander, and LEDs
P0.31	BH_INT	Color sensor interrupt line

Table 3: nRF52832 pin map

I/O	Label	Description
SXIO0	IOEXT0	Digital GPIO externally available
SXIO1	IOEXT1	Digital GPIO externally available
SXIO2	IOEXT2	Digital GPIO externally available
SXIO3	IOEXT3	Digital GPIO externally available
SXIO4	BAT_MON_EN	Battery monitoring enable
SXIO5	LIGHTWELL_G	Green color of the lightwell LEDs
SXIO6	LIGHTWELL_B	Blue color of the lightwell LEDs
SXIO7	LIGHTWELL_R	Red color of the lightwell LEDs
SXIO8	MPU_PWR_CTRL	Motion sensor power control
SXIO9	MIC_PWR_CTRL	Microphone power control
SXIO10	CCS_PWR_CTRL	Gas sensor power control
SXIO11	CCS_RESET	Gas sensor reset line
SXIO12	CCS_WAKE	Gas sensor wake line
SXIO13	SENSE_LED_R	Red color of the color sensor support LED
SXIO14	SENSE_LED_G	Green color of the color sensor support LED
SXIO15	SENSE_LED_B	Blue color of the color sensor support LED

Table 4: I/O expander pin map

7.6 Motion sensors

Thingy includes a low-power accelerometer and a nine-axis motion sensor.

7.6.1 Low-power accelerometer

When Thingy is in low-power sleep mode, any user interaction will be detected by the accelerometer, which will then wake up the device.

The accelerometer has an I²C interface and it can detect motion on three axes. It is connected directly to the voltage regulator output so it is powered regardless of the VDD power control, see [Power supply](#) on page 45 for more information. For this reason, the I²C interface is separated from the rest of the sensors of the board, and the accelerometer is connected to the same I²C bus that is available externally. If external I²C units are connected, they must not use the I²C address used for the low-power accelerometer (0x19).

Low power accelerometer I2C address: 0x19

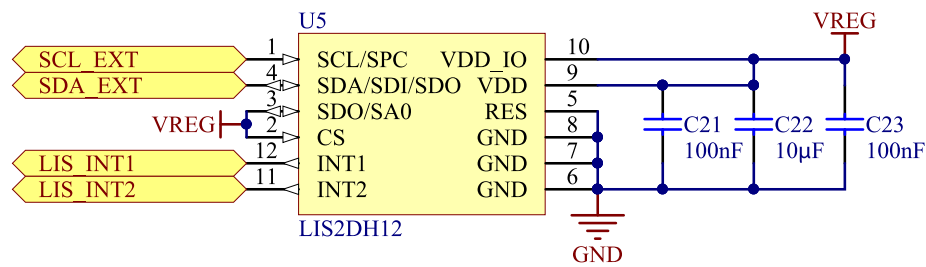


Figure 7: Low-power accelerometer schematic

By default, the INT1 line of the accelerometer is connected to nRF52832. If you want to use the INT2 line instead, cut the short on **SB9** and solder **SB10**.

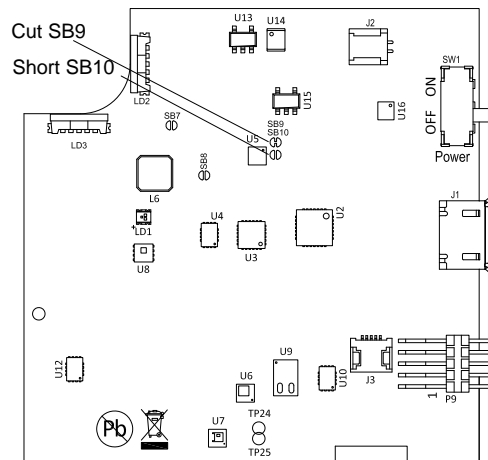


Figure 8: Low-power accelerometer interrupt line selection

7.6.2 Nine-axis motion sensor

For advanced motion tracking features, Thingy uses a three-axis gyro, three-axis accelerometer, and three-axis magnetometer integrated device (**U3**).

The motion sensor is interfaced through the I²C bus (0x68). To reduce power consumption when the motion sensor is not in use, the power supply and signal lines are connected through an analog switch (**U4**). This enables the device to be completely cut off from the rest of the circuit when not in use. When the motion sensor is not in use, the GPIO used for the interrupt signal can be accessed on test point **TP1**.

Motion tracking device I2C address: 0x68

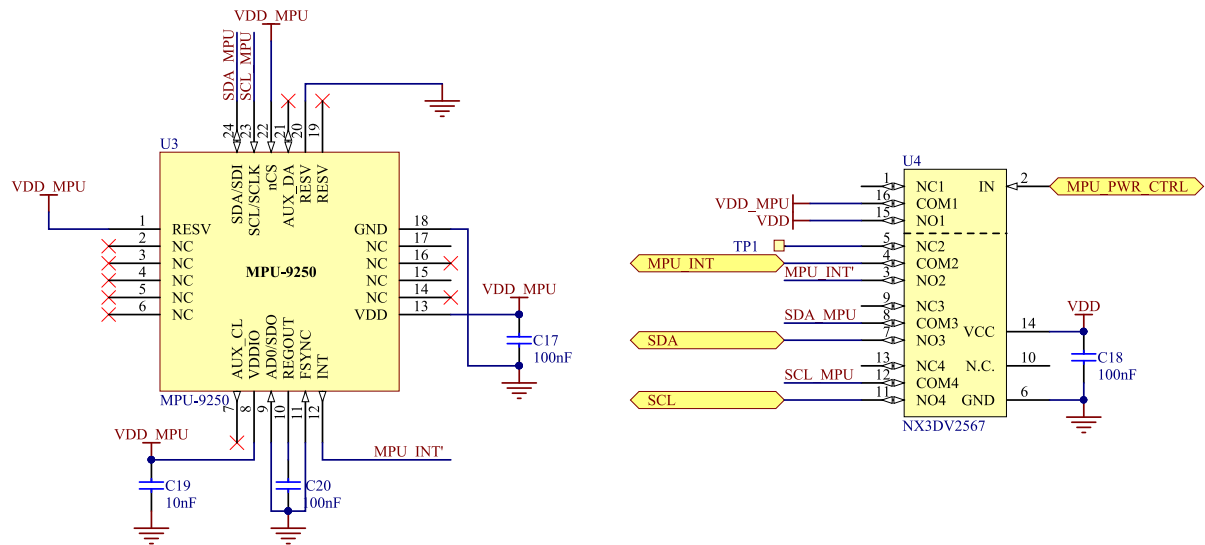


Figure 9: Motion sensor schematic

7.7 Environment sensors

To monitor its surroundings, Thingy contains several sensors for detecting different environmental properties.

7.7.1 Pressure sensor

The pressure sensor (**U6**) onboard Thingy is capable of measuring 260 to 1260 hPa absolute pressure with 24-bit data output.

The sensor has built-in temperature compensation, and as a secondary function, it provides 16-bit temperature data output. The pressure sensor is interfaced through I²C (slave address 0x5C).

Pressure/Altitude and Temperature sensor I2C address: 0x5C

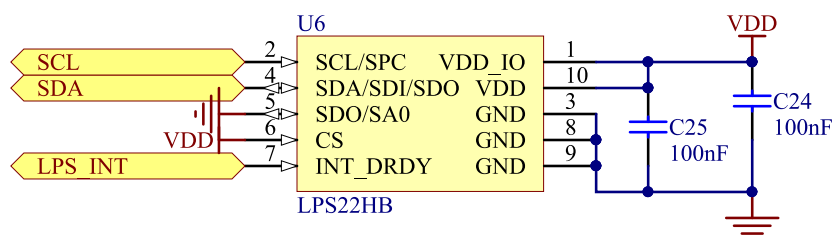


Figure 10: Pressure sensor schematic

7.7.2 Humidity sensor

For measuring humidity, Thingy has a combined humidity and temperature sensor onboard (**U7**).

It has a 0 to 100% relative humidity range and provides 16-bit humidity and temperature data output. The humidity sensor accesses the MCU through I²C (slave address 0x5F).

Humidity and Temperature sensor I2C address: 0x5F

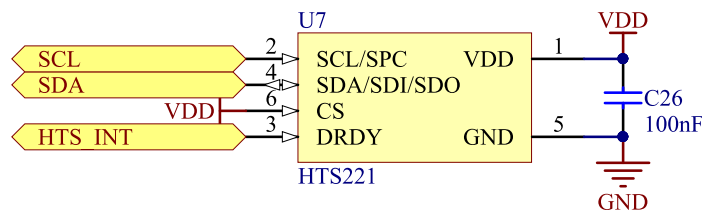


Figure 11: Humidity sensor schematic

7.7.3 Color sensor

The color sensor (**U8**) onboard Thingy senses red, green, and blue light with a 0.005 – 40k lx dynamic range.

The sensor faces towards the blue transparent bottom case with light pipes guiding the light towards the sensor. To measure the color on a surface, the color sensor is accompanied with an RGB LED that can illuminate the surface enabling the color sensor to read the color of the reflected light. The color sensor is accessed through I²C (slave address 0x38).

Color sensor I2C address: 0x38

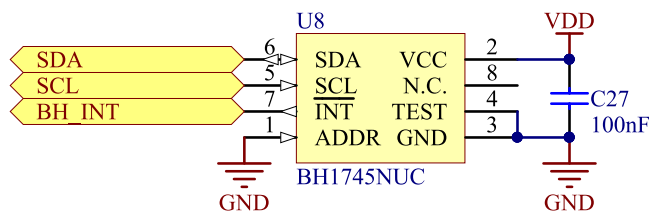


Figure 12: Color sensor schematic

7.7.4 Gas sensor

The gas sensor (**U9**) monitors indoor air quality and can detect a wide range of Volatile Organic Compounds (VOCs).

It provides a TVOC value or equivalent CO₂ (eCO₂) as output over the I²C bus (slave address 0x5A). To improve current consumption when the gas sensor is not used, the power, I²C lines, and the interrupt line are routed through an analog switch (**U10**). When the gas sensor is not in use, the GPIO used for the interrupt signal can be accessed on test point **TP2**.

Gas sensor I2C address: 0x5A

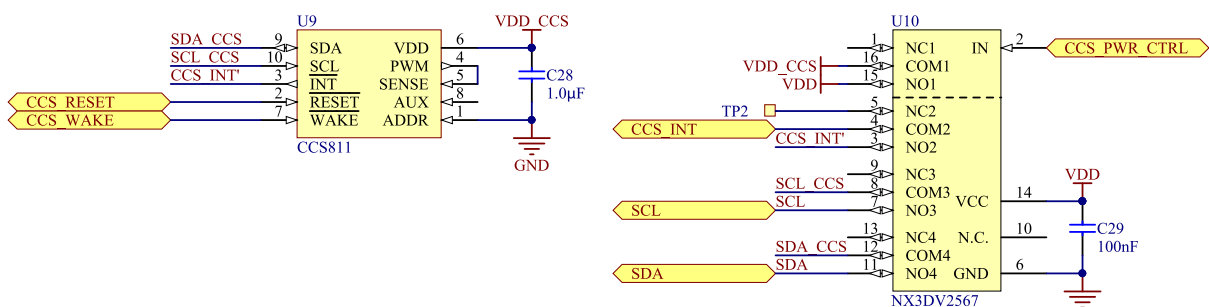


Figure 13: Gas sensor schematic

7.8 Sound

A digital microphone and a speaker make up the two hardware blocks that are responsible for sound on Thingy.

7.8.1 Digital microphone

For audio input, Thingy is equipped with a digital output PDM microphone (**U11**).

The microphone's power and signal lines are routed through an analog switch (**U12**) to save power when the microphone is not in use. When the microphone is switched off, the GPIOs used for the PDM signal are available on two test points (**TP3** and **TP4**). This allows the user to connect other hardware if the microphone is not in use.

Digital Microphone

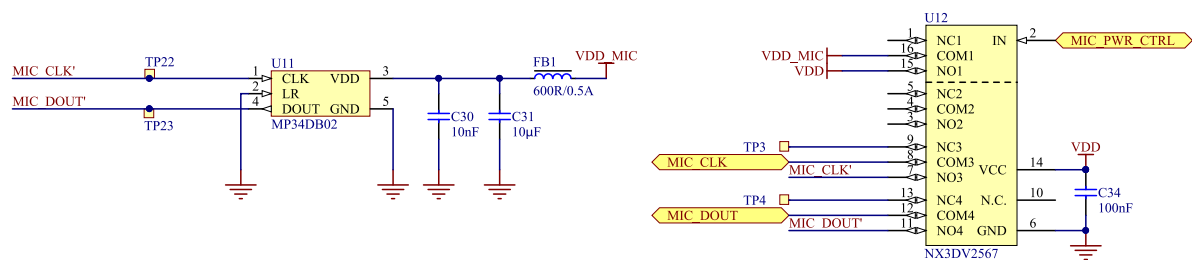


Figure 14: Digital microphone schematic

7.8.2 Amplifier and speaker

For audio output, Thingy has a miniature speaker that is driven from a simple H-bridge amplifier with a PWM input.

A couple of zener diodes minimize the time where both the P-channel and the N-channel MOSFET are turned on at the same time. Power supply for the amplifier comes directly from the battery to maximize the power to the speaker. The power supply is routed through a switch to avoid leakage currents when the speaker is not in use.

Amplifier and speaker

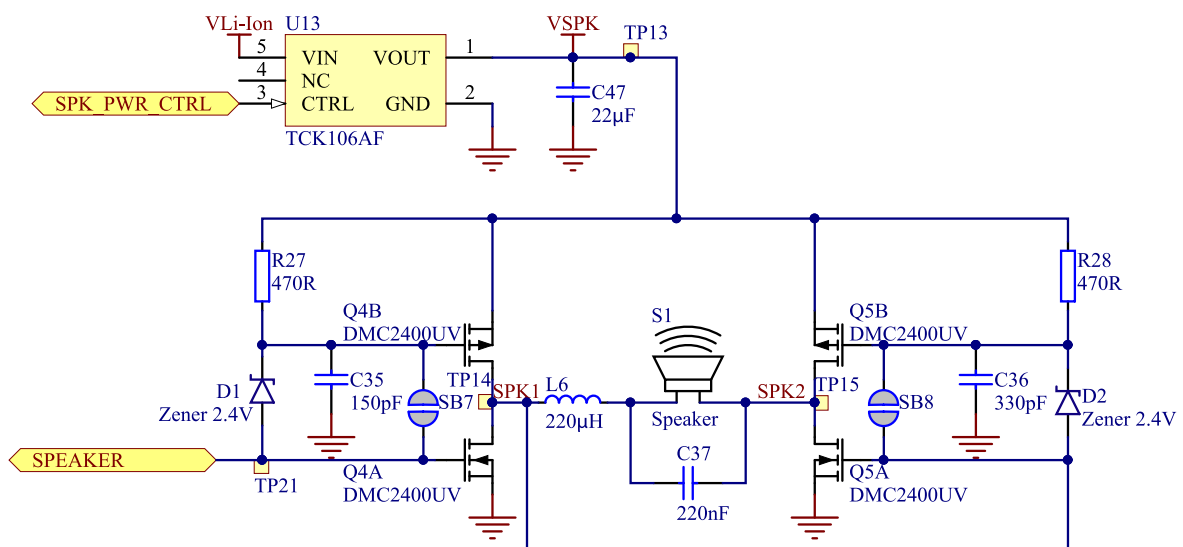


Figure 15: Amplifier and speaker schematic

7.9 LEDs and button

Thingy user interface consists of RGB LEDs and a button.

7.9.1 RGB LED

Thingy is equipped with three RGB LEDs.

Two of the LEDs are used to light up the light well and are controlled by the same signals using transistors as switches. The third LED is located near the color sensor and is used as auxiliary light for color measurements.

LEDs

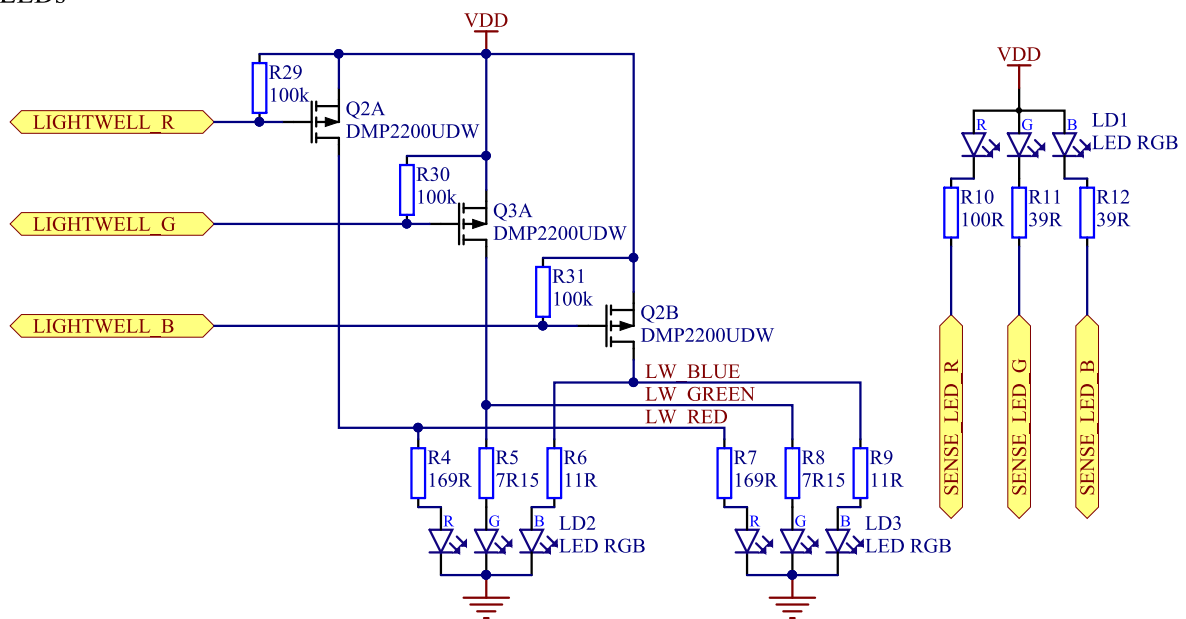


Figure 16: LED schematic

7.9.2 Button

The button located on the top of Thingy is for user input.

Button

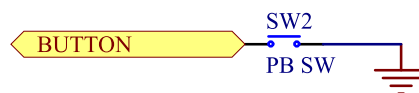


Figure 17: Button schematic

7.10 Power supply

The main power source is a rechargeable lithium polymer (Li-Po) battery. The battery has a nominal capacity of 1440 mAh and it can be recharged through USB.

Thingy has a power switch that physically disconnects the battery and the USB power from the rest of the circuits. This switch must be on in order for Thingy to work and to charge the battery. A voltage divider circuit is connected to the USB power so the MCU can detect when a USB cable is connected.

USB- and battery connector and power switch

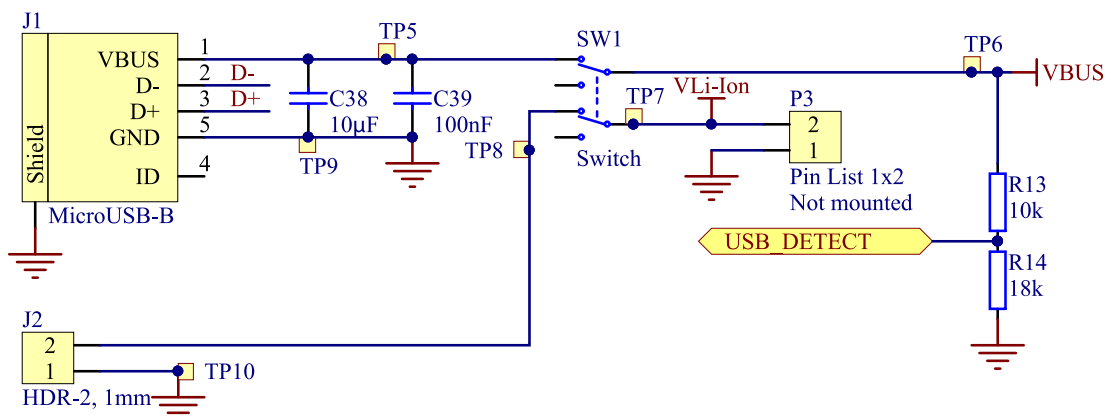


Figure 18: Schematics for USB and battery connectors, and the power switch

A battery charger circuit is present on Thingy in order to charge the battery. The max charge current is set to 0.5 C. Battery charge status output is connected to the MCU. While the charger has a thermistor input, the battery used in Thingy has no internal thermistor. Therefore, there is a thermistor on the main circuit board to monitor the temperature in the casing.

Battery charger

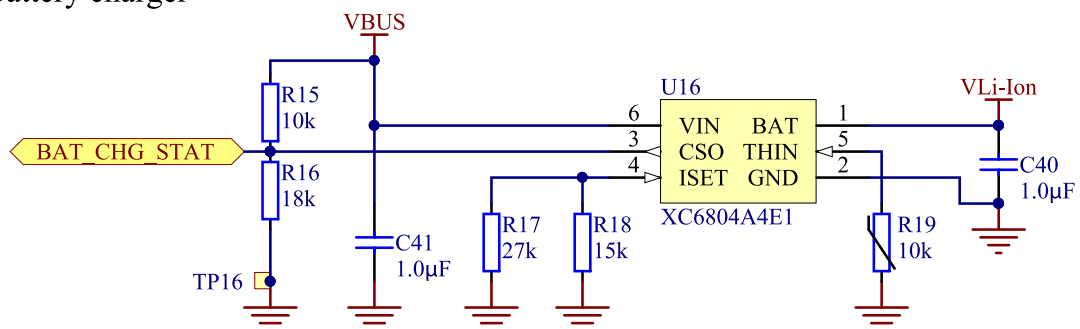


Figure 19: Battery charger schematic

To provide the circuits with a stable voltage, a DC/DC buck regulator with an output of 3.3 V is used.

Voltage regulator

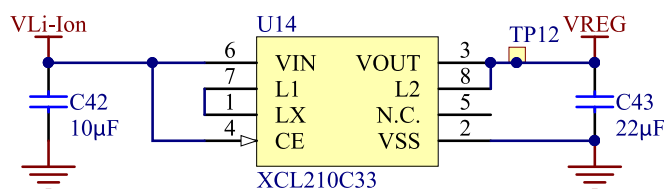


Figure 20: Voltage regulator schematic

To save power during sleep mode, an analog switch is used to turn off the VDD power net. The low-power accelerometer and nRF52832 are powered directly from VREG and will always be powered to wake up Thingy.

VDD power switch

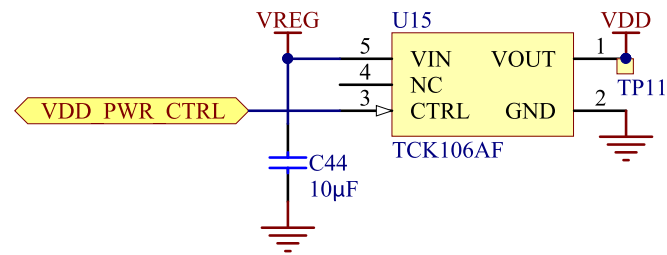


Figure 21: VDD power switch schematic

To monitor the battery voltage, a voltage divider circuit is connected to the battery. nRF52832 will use the ADC to read out the voltage level. To avoid unnecessary current drain, the voltage divider is turned off with a transistor when not in use.

Battery monitoring

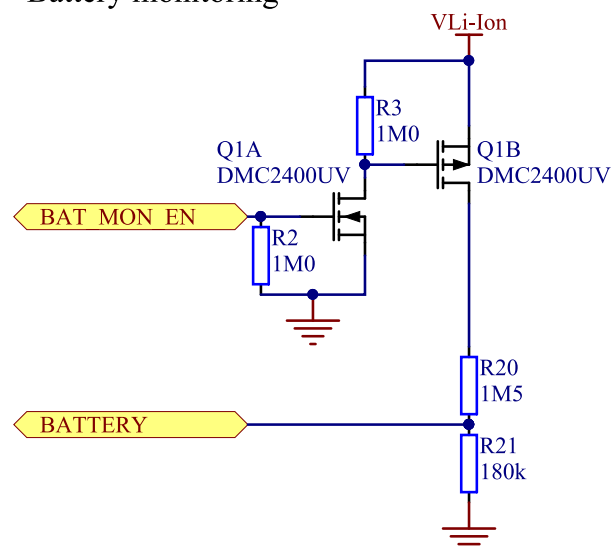


Figure 22: Battery monitoring schematic

7.10.1 Current measurement

It is possible to measure the current flowing to nRF52832 by cutting the short on SB2 and placing an ampere meter between the positive terminal and **P1** and positive terminal and **P2**.

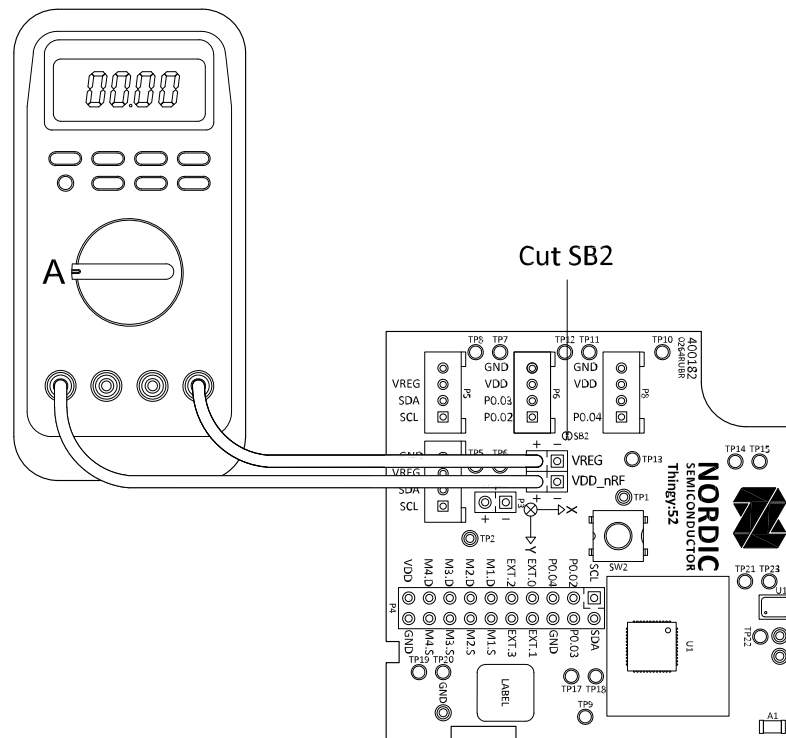


Figure 23: Measuring current to the nRF52832

nRF current measurement connectors

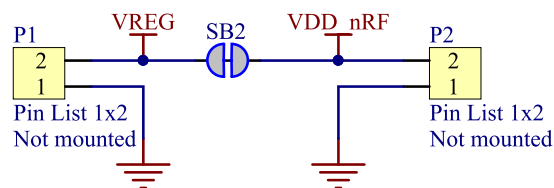


Figure 24: Current measurement connectors schematic

7.11 Interface

To enable the user to connect external hardware, Thingy is equipped with a set of connector footprints and transistors to drive higher currents.

7.11.1 N-MOS transistors

For more flexibility, Thingy is equipped with four N-MOS transistors that can be used to drive small DC motors or LEDs. The drain and source of the transistors are available on external connectors and the gate is connected directly to the nRF52832.

N-MOS transistors

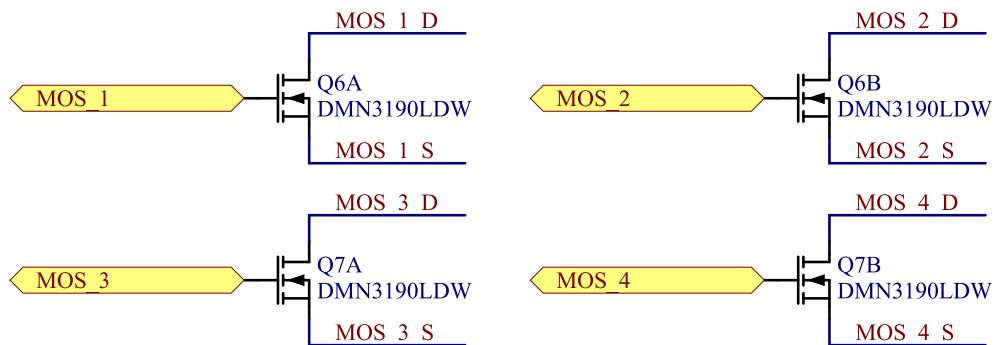


Figure 25: Schematics for the N-MOS transistors

7.11.2 Connectors

Extra GPIOs can be found on connector **P4**. In addition to **P4**, some of the GPIOs are available on connectors **P5-P8**.

For more information, see [Connector pinouts](#) on page 51.

These connectors are compatible with the Grove system from Seeedstudio. **P5** and **P7** are connected to the same I²C bus as the low-power accelerometer, and only I²C devices should be connected to these connectors. **P6** has two analog/digital I/Os, and **P8** has one.

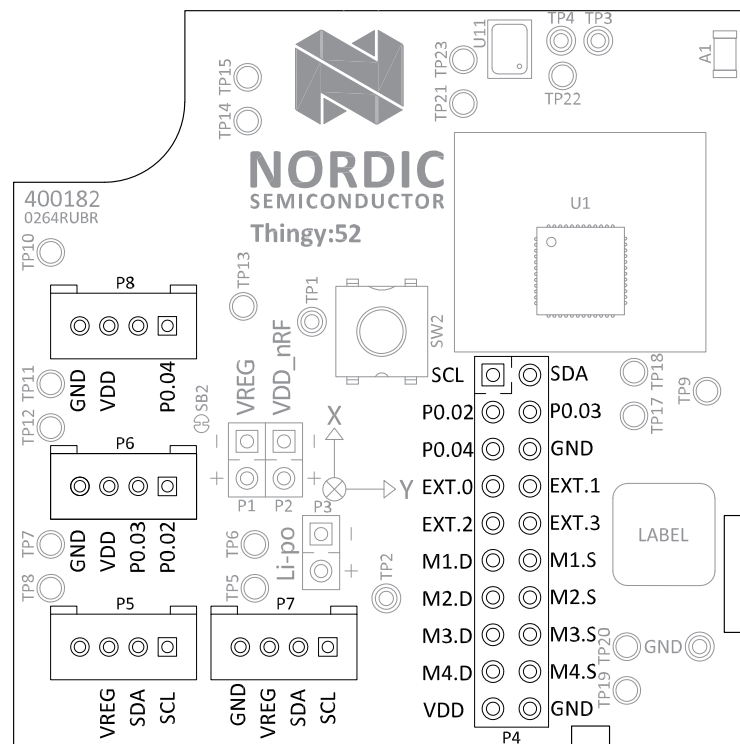


Figure 26: Thingy external connectors

Interface connectors

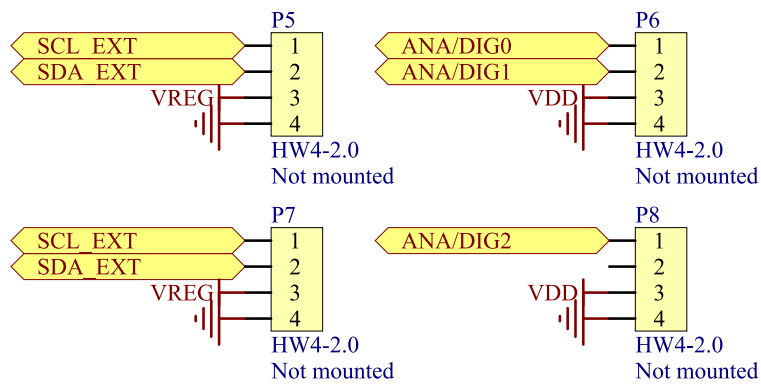
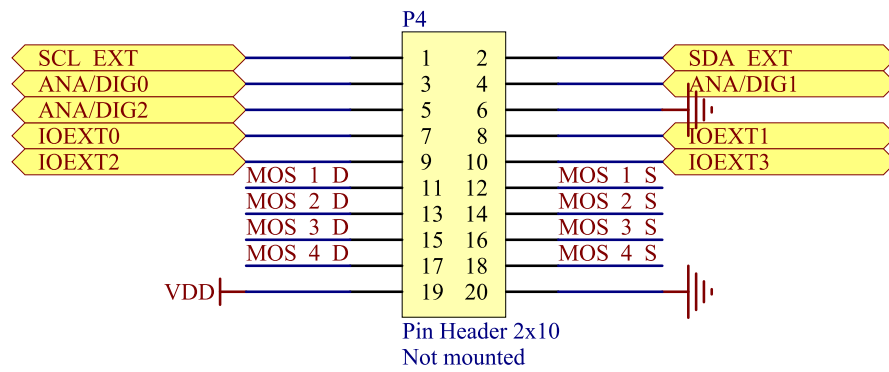


Figure 27: Interface connectors

7.11.2.1 Connector pinouts

Pin	Signal	Description
1	SCL_EXT	External I ² C clock
2	SDA_EXT	External I ² C data
3	ANA/DIG0	GPIO of the nRF52832
4	ANA/DIG1	GPIO of the nRF52832
5	ANA/DIG2	GPIO of the nRF52832
6	GND	Ground
7	IOEXT0	GPIO of the I/O expander
8	IOEXT1	GPIO of the I/O expander
9	IOEXT2	GPIO of the I/O expander
10	IOEXT3	GPIO of the I/O expander
11	MOS_1_D	Drain of n-channel MOSFET 1
12	MOS_1_S	Source of n-channel MOSFET 1
13	MOS_2_D	Drain of n-channel MOSFET 2
14	MOS_2_S	Source of n-channel MOSFET 2
15	MOS_3_D	Drain of n-channel MOSFET 3
16	MOS_3_S	Source of n-channel MOSFET 3
17	MOS_4_D	Drain of n-channel MOSFET 4
18	MOS_4_S	Source of n-channel MOSFET 4
19	VDD	Power supply, controlled by the VDD power switch
20	GND	Ground

Table 5: Pinout of connector P4

Pin	Signal	Description
1	SCL_EXT	External I ² C clock
2	SDA_EXT	External I ² C data
3	VREG	Power supply, no power control
4	GND	Ground

Table 6: Pinout of connectors P5 and P7

Pin	Signal	Description
1	ANA/DIG0	GPIO of the nRF52832
2	ANA/DIG1	GPIO of the nRF52832
3	VDD	Power supply, controlled by the VDD power switch
4	GND	Ground

Table 7: Pinout of connector P6

Pin	Signal	Description
1	ANA/DIG2	GPIO of the nRF52832
2	N/A	No connection
3	VDD	Power supply, controlled by the VDD power switch
4	GND	Ground

Table 8: Pinout of connector P8

Legal notices

By using this documentation you agree to our terms and conditions of use. Nordic Semiconductor may change these terms and conditions at any time without notice.

Liability disclaimer

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

All information contained in this document represents information on the product at the time of publication. Nordic Semiconductor ASA reserves the right to make corrections, enhancements, and other changes to this document without notice. While Nordic Semiconductor ASA has used reasonable care in preparing the information included in this document, it may contain technical or other inaccuracies, omissions and typographical errors. Nordic Semiconductor ASA assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

Life support applications

Nordic Semiconductor products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury.

Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

RoHS and REACH statement

Nordic Semiconductor products meet the requirements of *Directive 2011/65/EU of the European Parliament and of the Council on the Restriction of Hazardous Substances (RoHS 2)* and the requirements of the *REACH* regulation (EC 1907/2006) on Registration, Evaluation, Authorization and Restriction of Chemicals.

The SVHC (Substances of Very High Concern) candidate list is continually being updated. Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website www.nordicsemi.com.

Trademarks

All trademarks, service marks, trade names, product names and logos appearing in this documentation are the property of their respective owners.

Copyright notice

© 2017 Nordic Semiconductor ASA. All rights are reserved. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

