



**Nordic Thingy:52**  
**User Guide**  
**v1.0**

# Contents

- Revision history..... 4**
- Chapter 1: Introduction..... 5**
- Chapter 2: Key features..... 6**
  - 2.1 Power switch..... 6
  - 2.2 Button..... 7
  - 2.3 USB port..... 7
- Chapter 3: Kit content..... 9**
  - 3.1 Hardware content..... 9
  - 3.2 Downloadable content..... 9
- Chapter 4: iOS app..... 11**
  - 4.1 Connecting Thingy..... 11
  - 4.2 Services..... 12
    - 4.2.1 Environment..... 13
    - 4.2.2 User interface..... 16
    - 4.2.3 Motion..... 17
    - 4.2.4 Sound..... 19
    - 4.2.5 Cloud..... 22
  - 4.3 Thingy Configuration..... 23
  - 4.4 Managing Thingy devices..... 24
    - 4.4.1 Add or remove a Thingy..... 24
    - 4.4.2 Connect or disconnect a Thingy..... 25
  - 4.5 Over-the-air device firmware update (OTA-DFU)..... 26
- Chapter 5: Android app..... 29**
  - 5.1 Connecting Thingy..... 29
  - 5.2 Services..... 31
    - 5.2.1 Environment..... 31
    - 5.2.2 User interface..... 33
    - 5.2.3 Motion..... 35
    - 5.2.4 Sound..... 37
    - 5.2.5 Cloud..... 38
  - 5.3 Thingy Configuration..... 39
  - 5.4 Managing Thingy devices..... 40
    - 5.4.1 Add or remove a Thingy..... 40
    - 5.4.2 Connect or disconnect a Thingy..... 42
  - 5.5 Over-the-air device firmware update (OTA-DFU)..... 42
- Chapter 6: Web app..... 44**
  - 6.1 Connecting Thingy..... 44
  - 6.2 Services..... 45

6.2.1 Environment.....	45
6.2.2 Motion.....	46
6.2.3 User interface.....	47
6.2.4 Sound.....	48
6.3 Thingy Configuration.....	49
6.4 IFTTT.....	50

**Chapter 7: Hardware description..... 51**

7.1 Hardware figures.....	51
7.2 Block diagram.....	52
7.3 MCU.....	53
7.4 I/O expander.....	53
7.5 Pin maps.....	54
7.6 Motion sensors.....	56
7.7 Environment sensors.....	58
7.8 Sound.....	60
7.9 LEDs and button.....	61
7.10 Power supply.....	62
7.11 Interface.....	66

**Legal notices..... 70**

# Revision history

Date	Version	Description
June 2017	1.0	First release

---

# Chapter 1

## Introduction

---

The Nordic Thingy:52™ (nRF6936) is a compact, power-optimized, multi-sensor 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.



### 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.



廢電池請回收

This device features a Lithium-Ion battery with a capacity of 1440 mAh.

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.

---

# Chapter 2

## Key features

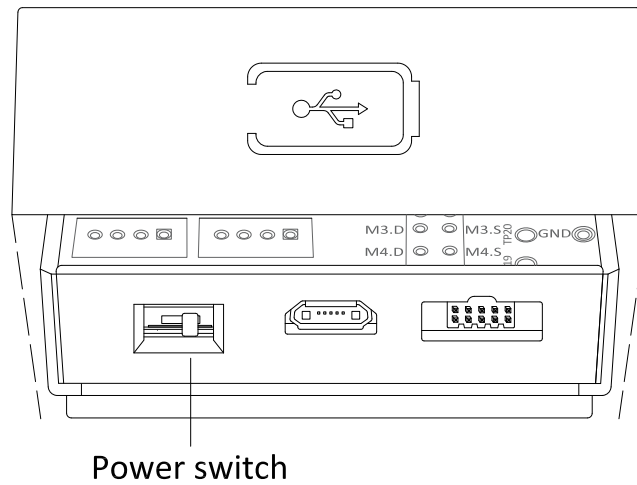
---

The following are the key features of Nordic Thingy:52.

- Highly configurable IoT sensor kit
  - All sensors and *Bluetooth*<sup>®</sup> 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 (CO<sub>2</sub> and TVOC)
  - Color and light intensity
- 9-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 (DFU)
- Low power consumption
- All software source code available and documented

### 2.1 Power switch

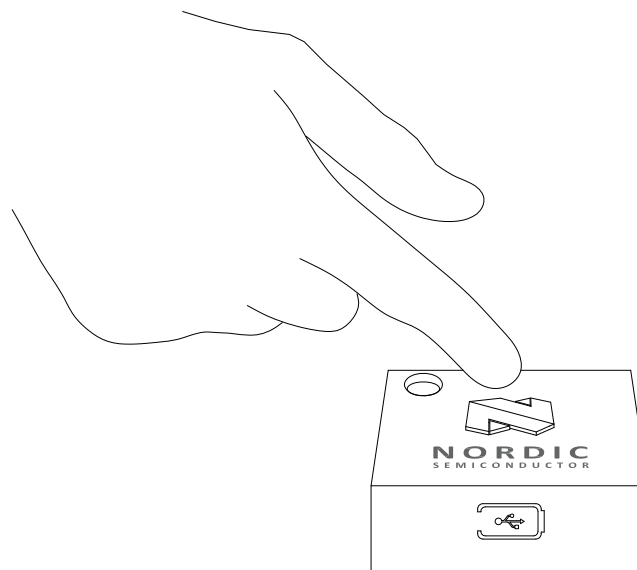
To locate the power switch, you must lift the top rubber case of Thingy.



**Figure 1: Power switch**

## 2.2 Button

The Thingy button is located right under the Nordic Semiconductor logo. Press it through the top rubber case.



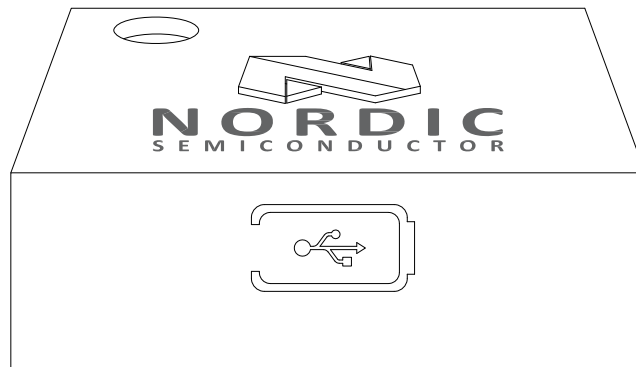
**Figure 2: Button**

## 2.3 USB port

The micro USB port is located on the front side of the device, under a rubber cover. Use it to charge your Thingy.

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

**Important:** The power switch must be on for the battery to charge.



**Figure 3: USB port**



---

## Chapter 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.



**Figure 4: 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.

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

### Other relevant nRF52832 documentation

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

---

# Chapter 4

## iOS app

---

Download and install the Nordic Thingy app from the Apple App Store to control Thingy using an iOS device.

To enjoy the functionalities of the iOS app that are described in this chapter, you need to complete the following steps.

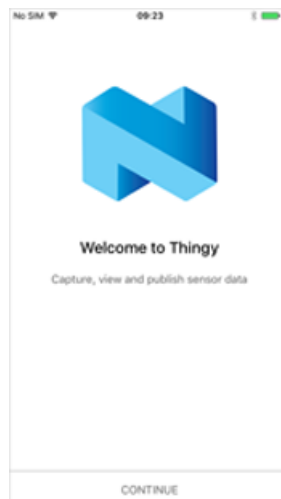
1. On your iOS device, download and install the Thingy app for iOS from the App Store.
2. Power on Thingy and make sure that it is not connected to any other mobile device. If Thingy is already connected to another device, its LED will breathe in cyan color.

**Note:** If Thingy goes into sleep mode, you can shake it or press the button to wake it up.

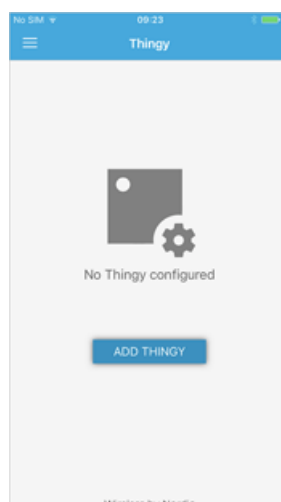
### 4.1 Connecting Thingy

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

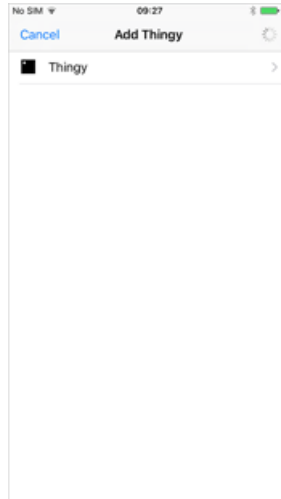
1. Tap the icon to launch the app. A welcome screen is displayed.
2. Tap **Continue**.



3. If you are running the app for the first time, a screen appears where you can add Thingy to your app. Tap **Add Thingy**. The Scanner starts the discovery process and lists any Thingy that is nearby.

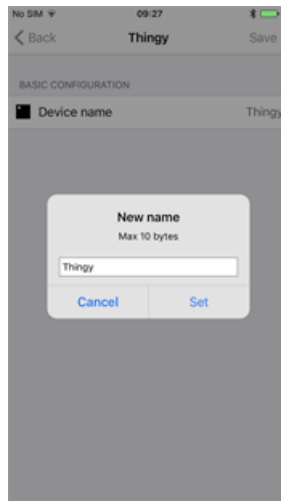


4. Once Thingy is discovered, tap its name in the list to start the configuration process (the default name is "Thingy"). The scanner is dismissed and the initialization view appears. If Thingy doesn't appear in the list within a couple of seconds, check the following:
  - a) Is Thingy powered on? The power switch is located under the rubber cover, to the left of the USB port.
  - b) Is Thingy connected to another mobile device? By default, the LED will breathe in cyan color to indicate that it is already connected to another device.
  - c) Is Thingy in sleep mode? Press the button or shake the device to make sure that it is not in sleep mode.
  - d) Has Thingy run out of battery? Connect Thingy to a charger. Make sure that the power switch is on when charging.



5. Add a name for your device in the **New name** field if you want to change the name from the default one.

6. Tap **Save**.



**Note:** 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.

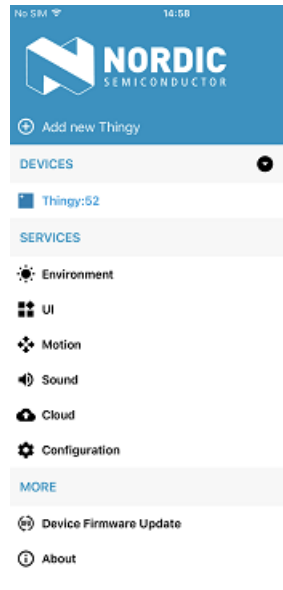
The app opens the **Environment** service, which is the default view for the Thingy app as long as one Thingy has been configured, regardless of its connection state. If Thingy hasn't been configured, the app will fall back to the [initial \(Add Thingy\) empty view](#).

## 4.2 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.

**Note:** The services listed in this chapter are available in the Thingy app. They do not reflect the exact service names or number on the device.

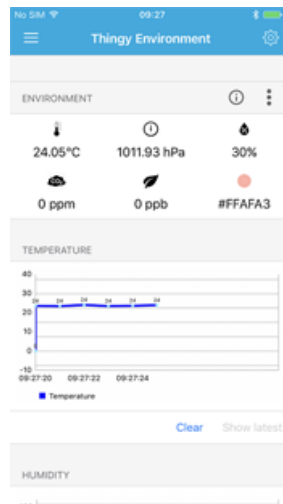
The services offered in the Thingy app are divided into Environment, User Interface (UI), Motion, Sound, and Cloud.



### 4.2.1 Environment

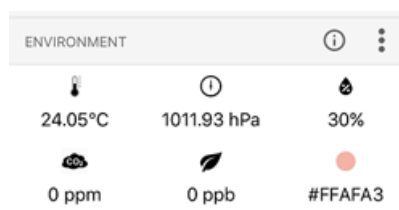
The Environment service provides you with a set of environment-related data, such as temperature, air humidity, air pressure, and light intensity.

When the app is first launched, it displays the **Thingy Environment** with all features enabled and notifying. After you have customized a feature's setting, it is saved and will display the next time you launch the app.

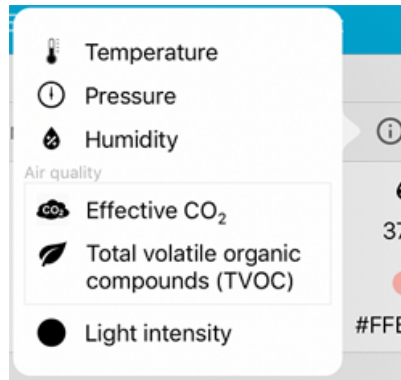


### Environment

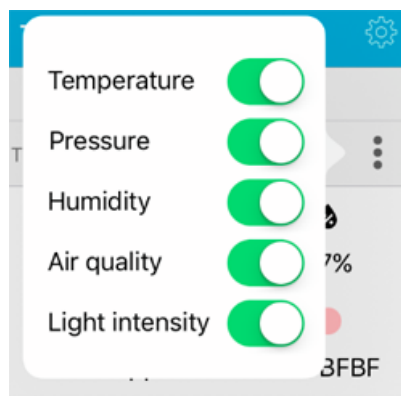
In the Environment section you can see the most current data for all enabled features.



Tap the Info button ⓘ for clarification on what each symbol stands for.



Tap the More options button ⋮ to enable or disable the available features.

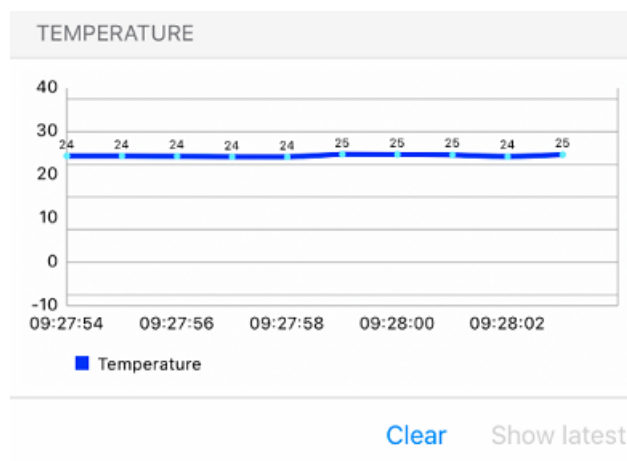


### Live graphs

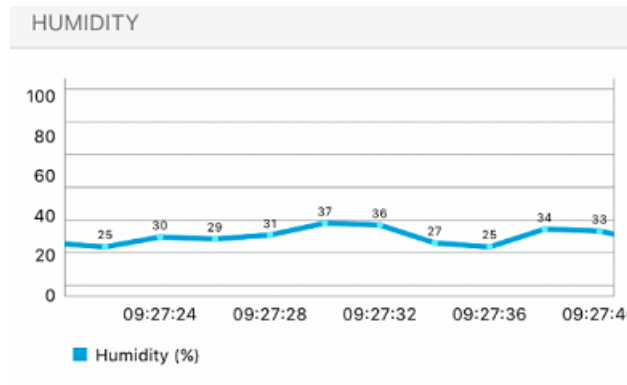
Live graphs are available for the temperature, pressure, and humidity features. Once enabled, data will start populating the graphs in realtime.

Here are some tips when interacting with a live graph.

- Graphs automatically scroll to the latest value with the **Show Latest** button disabled.

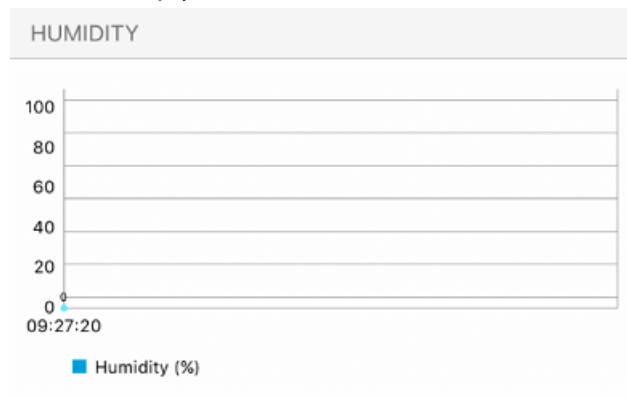


- Tap any part of the graph to see the timestamp and exact value of the entry.
- Tap and hold the graph to stop scrolling. The **Show latest** button will now be enabled.




Clear Show latest

- You can scroll through the graph manually, as with any other scrollable view.
- To resume autoscrolling, tap **Show latest**.
- To clear a graph, tap **Clear**. This will empty all data and will reenable autoscrolling.

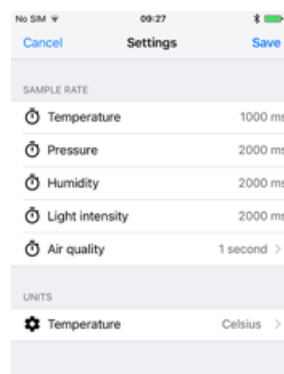


Clear Show latest

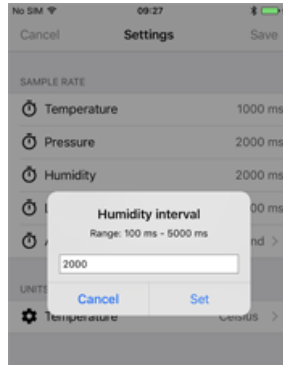
### Configuring the Environment service

Tap the Configuration button  to the right of the application's navigation bar to open configuration settings. In the configuration menu, you can customize several aspects of the Environment service.

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



To change a setting, tap the feature that you would like to change and type in the new value. Tap **Set** to save the changes. Tap **Cancel** to ignore your changes.



### 4.2.2 User interface

The User Interface service allows you to interact with the LED and button on Thingy.

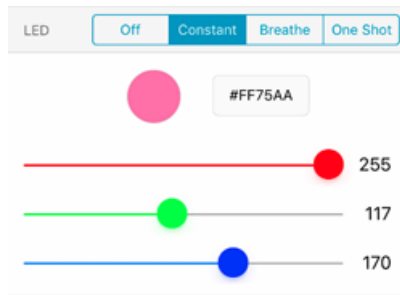
#### LED

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.



In Off mode, the LED is turned off completely and Thingy will remain in Off mode while disconnected. There is no configuration for this mode.

In Constant mode, the LED stays on, but once disconnected the mode turns off. There are red, green, and blue value sliders that allow you to change the color of the LED.



Breathe mode is where the Thingy LED fades in and out at a given interval, but turns off after Thingy is disconnected. You can choose from one of the preset colors, adjust the light intensity, and increase or decrease the breathe delay.



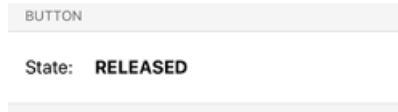
In one-shot mode, the Thingy LED fades in then out once, then remains off indefinitely. In the settings you can select the color of the LED and adjust the light intensity. This mode does not persist while disconnected.

#### Button

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



- Unknown - This state is displayed when the view is initially loaded and the app has no information about the button's state.
- Pressed - This is displayed when you press and hold the button.
- Released - This is displayed when you release the button.

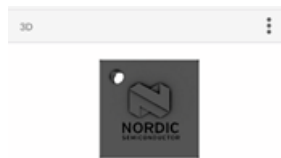


### 4.2.3 Motion


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.

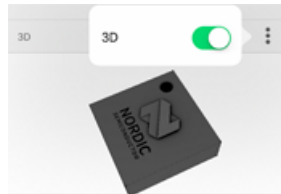
#### 3D Motion

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.



Follow these steps to interact with this feature:

1. Tap the More options button  in the 3D section.
2. Toggle the 3D switch. The model immediately starts simulating Thingy.



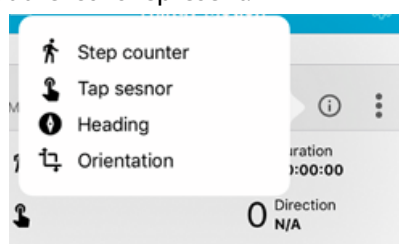
The simulation uses quaternions to rotate the model at a given configured interval. To make the animation smooth, the simulation interpolates between the current and target positions with a duration of the update interval. This makes the animation as smooth as possible.


#### Motion characteristics

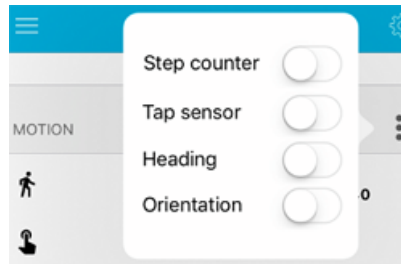
The characteristics in this section come from all motion-related sensors.

- Step counter - 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.)

Tap the Info button  to display what the icons represent.



Tap the More options button  on the Motion bar to enable or disable different characteristics using the toggle.



### Gravity Vector


The Gravity Vector section displays the effect of gravity on each axis in units of  $m/s^2$ . To use this feature, tap the 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.



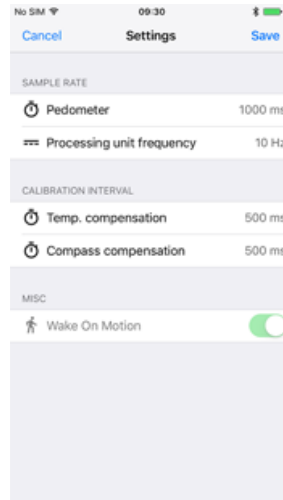
You can interact with the graph in the following way:

- 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.
- 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 reenale autoscrolling.
- Tap **Clear** to clear all entry points and reenale autoscrolling.

### Configuring the Motion service

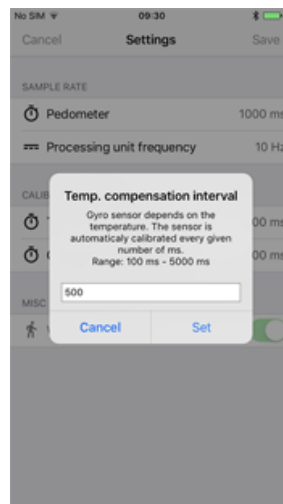
The Configuration button  is located to the right of the application's navigation bar. In the Configuration view, you can customize several aspects of the Motion service.

- Pedometer interval (ms)
- Processing unit frequency (Hz)
- Temperature compensation (ms) - This setting is used for internal calibration of the sensor.
- Compass compensation (ms) - This setting is used for internal calibration of the sensor.
- Wake On Motion - With this option enabled, Thingy will wake up from low-power mode when moved.



To change a setting:

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



#### 4.2.4 Sound

The Sound service lets you control Thingy's microphone and its speaker. Here you can find Bidirectional streaming and visualizer, 8-bit PCM streaming, piano keyboard, and sound samples.

##### Bidirectional streaming and visualizer

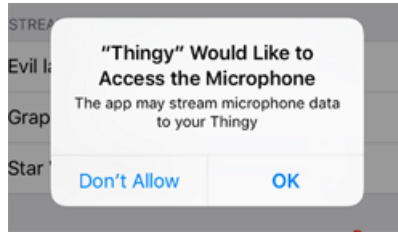
Here, you can control the streaming process from Thingy to the mobile device and vice versa.



Follow these steps to stream sound from the microphone on the mobile device to the Thingy speaker.

1. Tap the microphone icon on the left. If microphone permission is not granted yet, you will receive a permission request. If the permission was declined, iOS might not ask again, leaving this feature useless. (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.

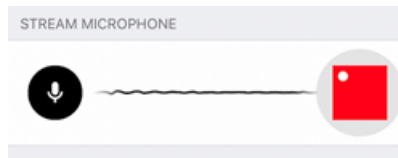


2. To stop streaming, tap the microphone button again. It will turn black and the graph will stop updating.

There are two ways in which you can stream sound from the Thingy's microphone to the mobile device's speaker.

**Method 1:**

1. Tap the Thingy icon on the right. The Thingy icon will turn red and the visualizer will display the sound wave coming from the Thingy. You should hear sound coming out of the mobile device's speakers.



2. To stop streaming, tap the Thingy icon. It will turn black and the graph will stop updating.

**Method 2:**

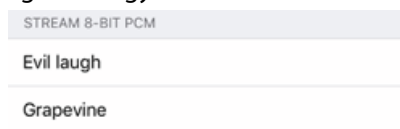
1. Navigate to the Sound view and open it. This type of streaming will only work when the Sound view is open.
2. Press the Thingy button located on the side of the casing (not the one in the app).
3. To stop streaming, press the Thingy button again.

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

**8-bit PCM streaming**

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.

Thingy is capable of streaming 8-bit PCM audio from the mobile device to its speaker. Selecting one of the example audio files will begin streaming to Thingy.

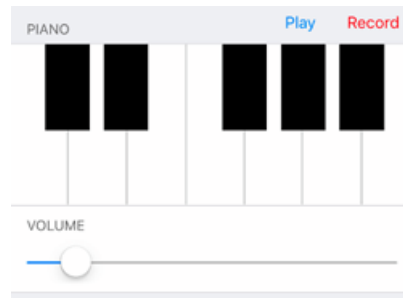


**Piano keyboard (Frequency mode)**

This mode allows you to play specific frequencies simulated by a keyboard. Only one note can be played at a time. **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.

**Playing the piano**

To play a note, tap the respective piano key.

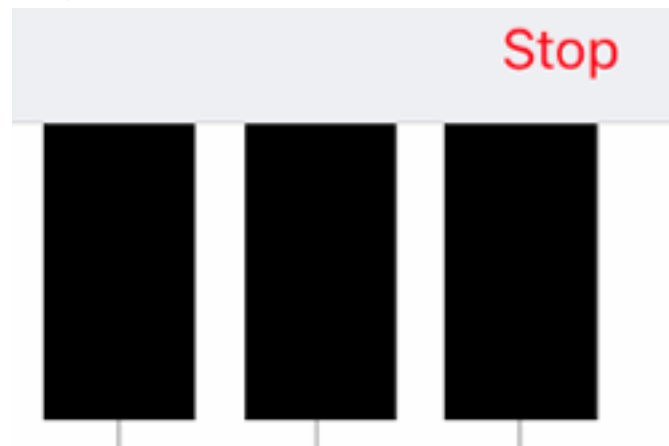


### Recording input for later playback

1. Press the **Record** button.



2. Play some notes.
3. Press the **Stop** button. The keystrokes are now stored.



### Playing back recorded keystrokes

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

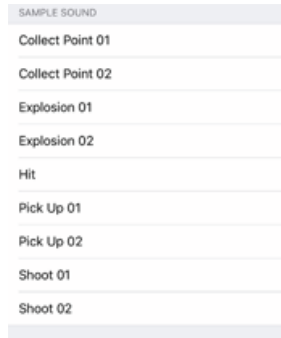
- Tap **Play** to start playing back the recorded keystrokes. The notes will start playing on Thingy.
- Playback will stop automatically when done, or when you tap **Stop**.

### 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.

### 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..



To use this feature, tap any of the samples in the Sample Sound view. The sound will be immediately played.

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

### Speaker and microphone modes

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 (such as streaming), the current mode will quit and the new mode will start. The speaker can function in only one mode at a time.

Microphone mode:

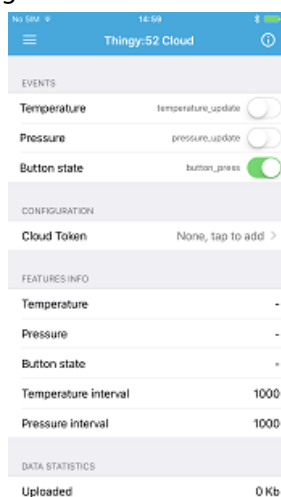
- ADPCM mode (only supported mode within the app)

### 4.2.5 Cloud

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, tap the Info icon  in the upper right corner of the Cloud service view. A short tutorial is displayed on how to obtain the Maker Webhooks Token that is necessary for IFTTT integration. Paste this token in the **Cloud Token** configuration field.

The Cloud view is divided into the following sections.



### Events

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

Paste the Cloud Token in this field.

### Features Info

If a particular type of event from the **Events** 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.

## 4.3 Thingy Configuration

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

### Configuration settings

You can configure the following settings in the **Thingy Configuration** view:

- Name - Sets the name used when advertising the Thingy device.
- Advertising parameters
  - Advertising interval - How often Thingy advertises.
  - Advertising timeout - How long Thingy advertises for a connection request before timing out.
- Connection parameters
  - Minimum connection interval - The minimum interval in milliseconds at which the mobile device asks for data from Thingy.
  - Maximum connection interval - The maximum interval in milliseconds at which the mobile device asks for data from Thingy.
  - Slave latency - This is the number of events between when the mobile device asks Thingy for data and when Thingy actually sends 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.
  - Supervision timeout - A timeout 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 timeout to detect it quickly.
- Eddystone Beacon URL - Stores a URL in the Eddystone URI format.





Additionally, you can view the current firmware version that is running on Thingy.

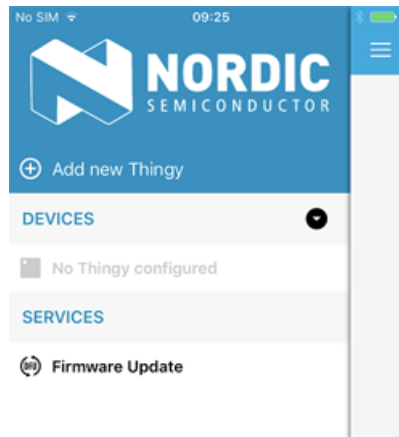
### Changing the configuration

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.

## 4.4 Managing Thingy devices

The mobile application supports connections to multiple Thingy devices, meaning you can have more than one Thingy configured in the app.

From the main menu of the application, you can add, remove, connect, or disconnect a Thingy.



### 4.4.1 Add or remove a Thingy

Adding or removing a Thingy is done in the main application menu.

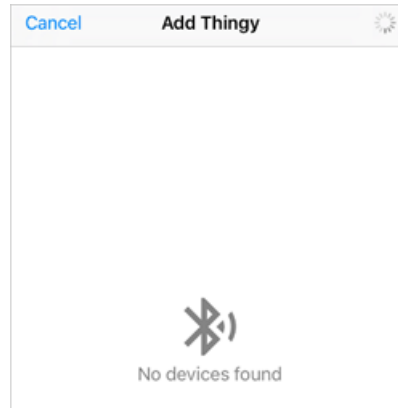
#### Add a Thingy

1. From the top left of the screen, open the main application menu.
2. Tap **Add new Thingy**.



The Scanner view appears and automatically starts scanning.

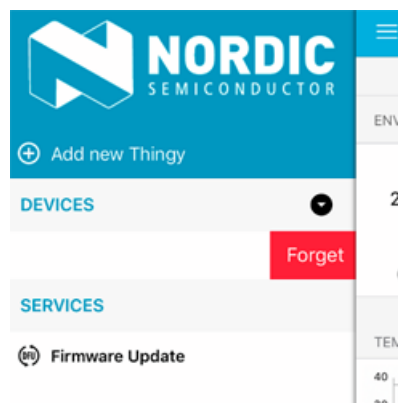




3. Once the Thingy device is discovered, select it from the menu and follow the wizard. The procedure is the same as during the first launch.

### Remove a Thingy

1. From the top left of the screen, open the main application menu.
2. Swipe left on the Thingy which you want to remove. A **Forget** button appears.
3. Tap the **Forget** button. The Thingy will be removed. If the device was connected, it will first be disconnected.

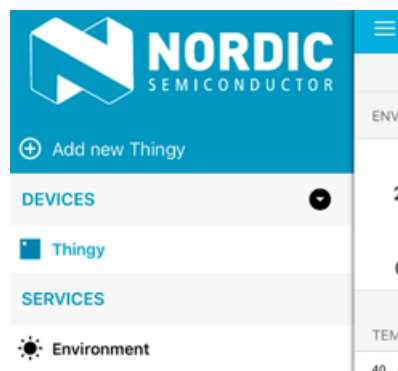


### 4.4.2 Connect or disconnect a Thingy

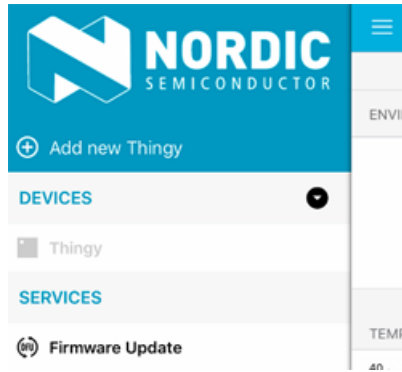
The Thingy app supports configuring and storing more than one Thingy, but only one device can be connected at a time.

To switch to an active Thingy, tap its icon to make it the active one. The currently connected one will automatically disconnect and the newly selected Thingy will connect. Only one Thingy can be active at a time.

1. Tap a disconnected Thingy from the main menu. The device will connect and its icon turns blue, indicating it's connected.



2. To disconnect a Thingy, tap a connected Thingy from the main menu. The device will disconnect and its icon turns grey, indicating that it's not connected.



## 4.5 Over-the-air device firmware update (OTA-DFU)

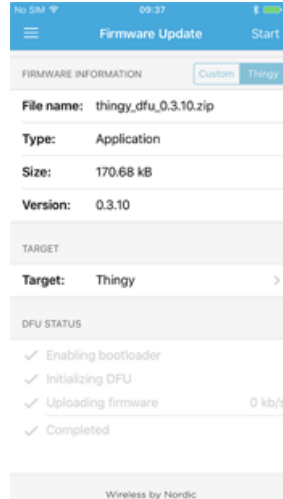
The mobile app provides support for over-the-air device firmware updates. This allow for seamless updates to the Thingy firmware application through the Thingy mobile application.

### OTA-DFU outline

When there is a feature addition 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.

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

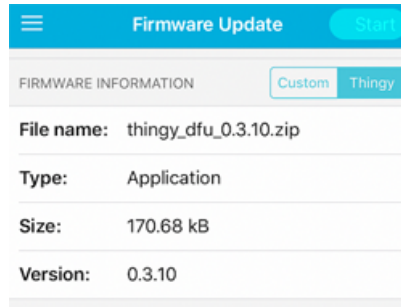


### Updating firmware on a connected Thingy

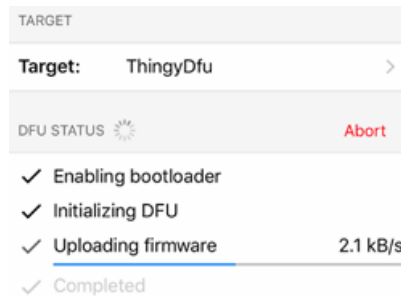
This scenario is a default use case.

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

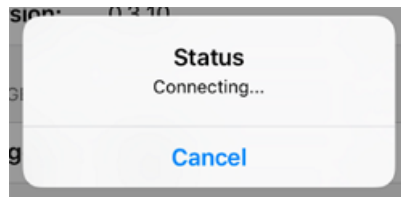
1. Connect to Thingy.
2. Navigate to the Firmware Update menu.
3. Tap **Start** in the top right of the navigation bar to start the DFU process on the device.



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.

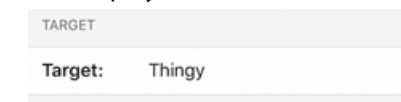


### Updating firmware on a Thingy that is in bootloader mode

This scenario is not a default use case.

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

1. Power Thingy off and then on while holding the button on top of Thingy to put it into bootloader mode. The device starts advertising with a new name "ThingyDFU". The MAC address is always incremented by 1 when a Thingy device is in bootloader mode.
2. If Thingy is in bootloader mode and not connected yet, tap **Target** to enter the Scanner view. 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.



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

---

# Chapter 5

## Android app

---

Download and install the Thingy app for Android from Google Play to control Thingy using an Android device. To enjoy the functionalities of the Android app that are described in this chapter, you need to complete the following steps.

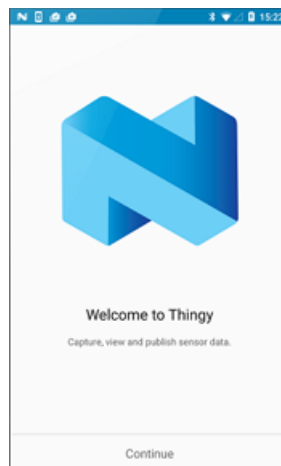
1. On your Android device, download and install the Thingy app for Android from Google Play.
2. Power on Thingy and make sure that it is not connected to any other mobile device. If Thingy is already connected to another device, its LED will breathe in blue color, by default.

**Note:** If Thingy went to sleep mode, you can shake it or press the button to wake it up.

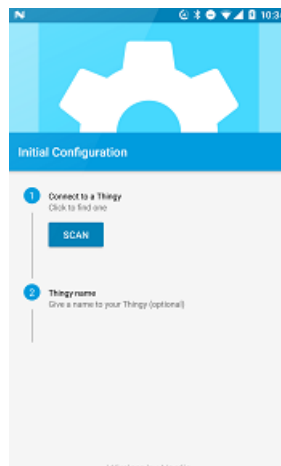
### 5.1 Connecting Thingy

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

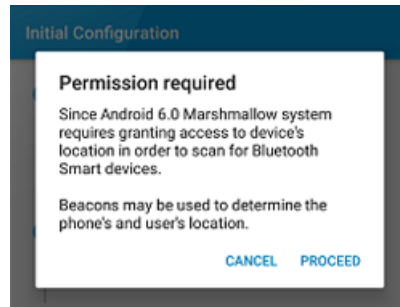
1. Tap the icon to launch the app. A welcome screen is displayed.
2. Tap **Continue**.



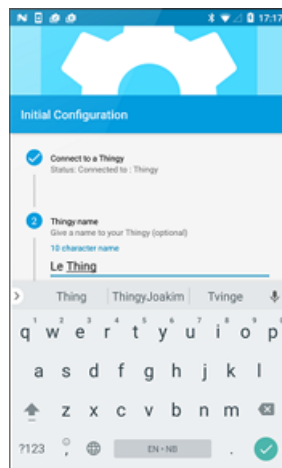
3. The **Initial Configuration** screen appears where you can set up Thingy. Tap **Scan**. The Scanner starts the discovery process and lists any Thingy that is nearby.



4. Tap **Proceed** to grant the Location Permission which is required for Android 6.0 Marshmallow and later.

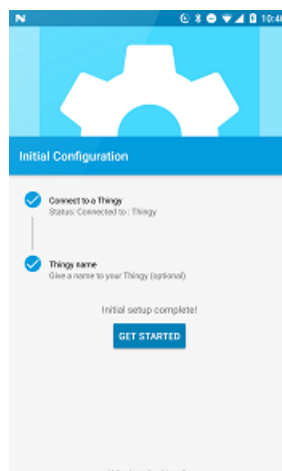


5. Tap **Scan** again to start scanning.
6. Select Thingy from the **Available Devices** list. If Thingy does not appear in the list within a couple of seconds, check the following:
  - a) Is Thingy powered on? The Power button is located under the rubber cover, to the left of the USB port.
  - b) Is Thingy connected to another mobile device? By default, the LED will breathe in cyan color to indicate that it is already connected to another device.
  - c) Is Thingy in sleep mode? Press the button or shake the device to make sure that it is not in sleep mode.
  - d) Has Thingy run out of battery? Connect Thingy to a charger. Make sure that the power switch is on when charging.
7. Add a name for your device under **Thingy name** if you want to change the name from the default one.



**Note:** 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.

8. After you complete the configuration steps, 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.



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.

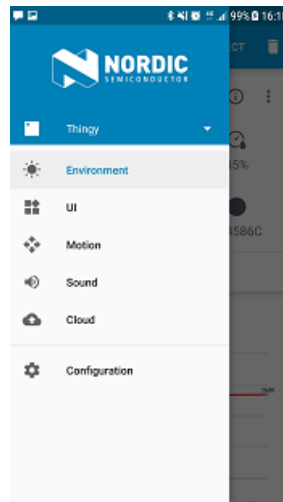
In the top right of the screen are the Connect/Disconnect options that change the connection state and a trash icon that removes the Thingy from the app. These options are available when at least one Thingy is configured in the app. If there are no devices configured or all of them have been deleted, the app falls back to the **No Thingy configured** view.

## 5.2 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.

**Note:** The services listed in this chapter are available in the Thingy app. They do not reflect the exact service names or number on the device.

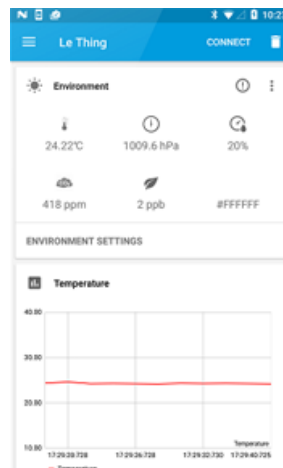
The services offered in the Thingy app are divided into Environment, User Interface (UI), Motion, Sound, and Cloud.



### 5.2.1 Environment

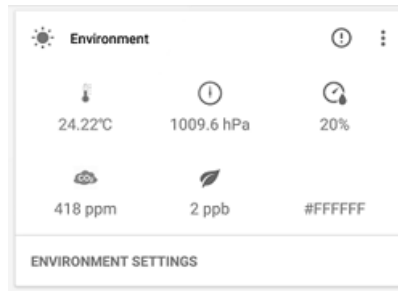
The Environment service provides you with environmental data, such as temperature, air humidity, air pressure, and light intensity.

When the app is first launched, it displays the **Environment** view with all characteristics enabled and notifying. After you have customized a feature's setting, it is saved and will display the next time you launch the app.

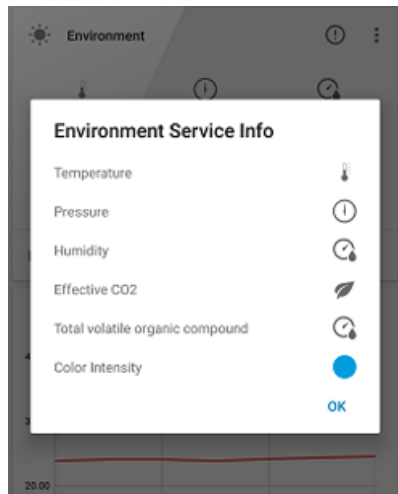


## Environment

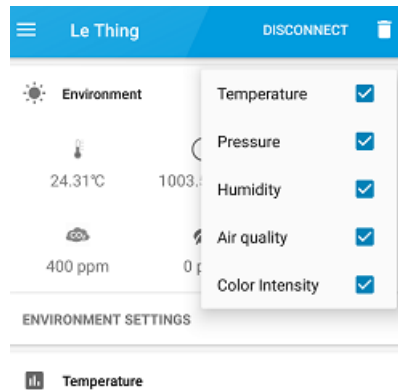
In the Environment section you can see the most current data for all enabled features.



Tap the Info button ⓘ for clarification on what each symbol stands for.



Tap the Action button ⋮ to enable or disable the available features.



## Live graphs


Live graphs are available for the temperature, pressure, and humidity features. Once enabled, data will start populating the graphs in realtime.

Here are some tips when interacting with a live graph.

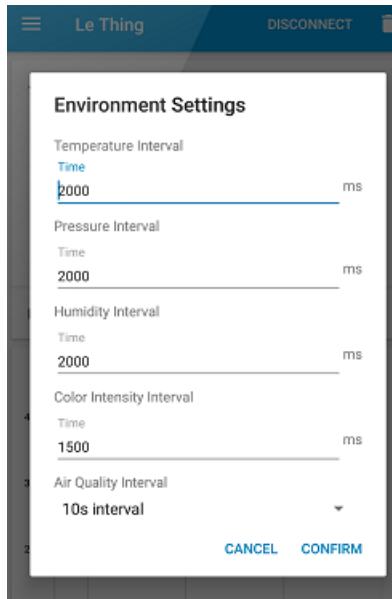
- Graphs automatically scroll to the latest value.
- Tap any part of the graph to see the timestamp and exact value of the entry.
- You can scroll through the graph manually, as with any other scrollable view.



## Configuring the Environment service

Tap the Configuration button  to right of the application's navigation bar to open configuration settings. In the configuration menu, you can customize several aspects of the Environment service.

- Temperature sampling interval (ms)
- Pressure sampling interval (ms)
- Humidity sampling interval (ms)
- Light intensity sampling interval (ms)
- Air quality sampling interval - This setting uses predefined values of 1, 10, or 60 seconds only.



To change a setting, tap the feature that you would like to change and enter the new value. Tap **Confirm** to save the changes. Tap **Cancel** to ignore your changes.

### 5.2.2 User interface

The User Interface service allows you to interact with the LED and button on Thingy.

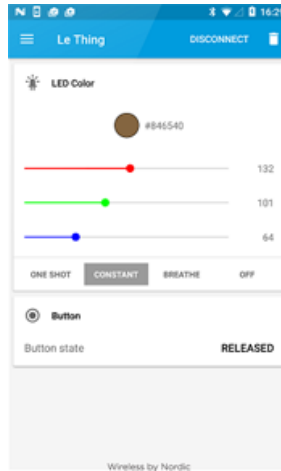
#### LED Color

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.

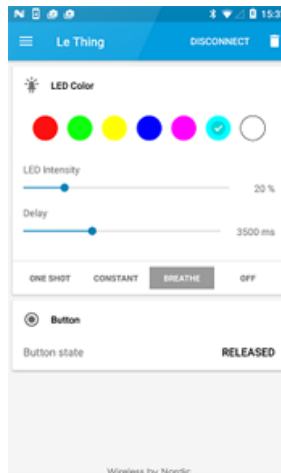


In Off mode, the LED is turned off completely and Thingy will remain in Off mode while disconnected. There is no configuration for this mode.

In Constant mode the LED stays on, but once disconnected the mode turns off. There are red, green, and blue value sliders that allow you to change the color of the LED.



Breathe mode is where the Thingy LED fades in and out at a given interval, but turns off after Thingy is disconnected. You can choose from one of the preset colors, adjust the light intensity, and increase or decrease the breath delay.



In one-shot mode, the Thingy LED fades in then out once, then remains off indefinitely. In the settings you can select the color of the LED and adjust the light intensity. This mode does not persist while disconnected.

### Button

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

- Unknown - This state is displayed when the view is initially loaded and the app has no information about the button's state.
- Pressed - This is displayed when you press and hold the button.
- Released - This is displayed when you release the button.



### 5.2.3 Motion


The Motion service displays the step counter, 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.

#### 3D Motion

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



Follow these steps to interact with this feature:

1. Tap the Action button  in the 3D section.
2. Toggle the 3D switch. The model immediately starts simulating Thingy.

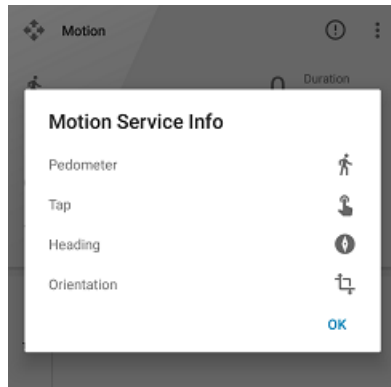
The simulation uses quaternions to rotate the model at a given configured interval. To make the animation smooth, the simulation interpolates between the current and target positions with a duration of the update interval. This makes the animation as smooth as possible.

#### Motion characteristics

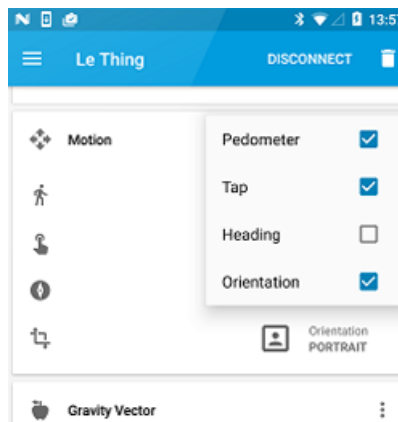
The characteristics in this section come from all motion-related sensors.

- Step counter - 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.)

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

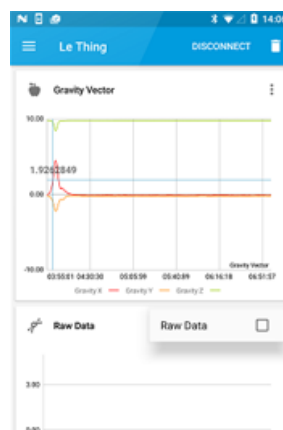


Tap the Action button ⋮ on the Motion section to enable or disable different characteristics by selecting or deselecting a characteristic.



### Gravity Vector

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 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.



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.

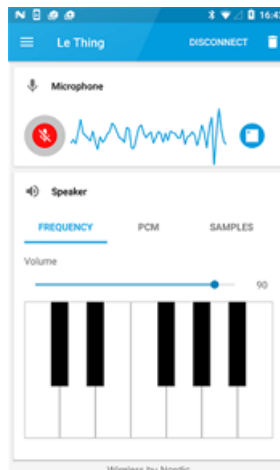
## 5.2.4 Sound

The Sound service lets you control Thingy's microphone and its speaker.

### Microphone

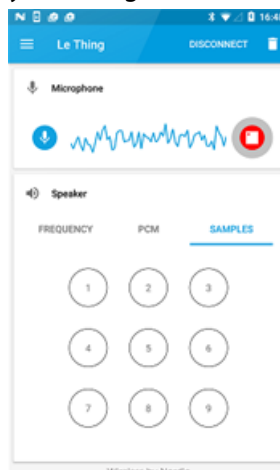
Here, you can control the streaming process from Thingy to a mobile device and vice versa. Follow these steps to stream sound from the mobile device's microphone to the Thingy speaker.

1. Tap the microphone icon on the left. If microphone permission is not granted yet, you will receive a permission request. If the permission was declined, Android might not ask again, leaving this feature useless. (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. It will turn black and the graph will stop updating. Follow these steps to stream sound from the Thingy's microphone to the mobile device's speaker.

1. Tap the Thingy icon on the right. The Thingy icon will turn red. The visualizer displays the sound wave coming from Thingy and audio will be played through the mobile device's speakers.

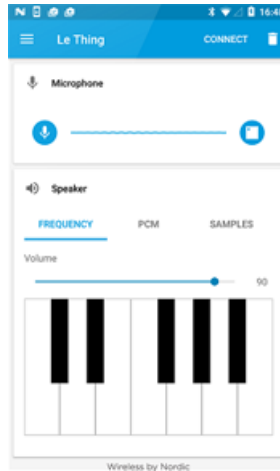


2. To stop streaming, tap the Thingy icon. It will turn black and the graph will stop updating.

### Speaker

This section contains three audio streaming modes.

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



- PCM mode - 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.



**Important:** 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.

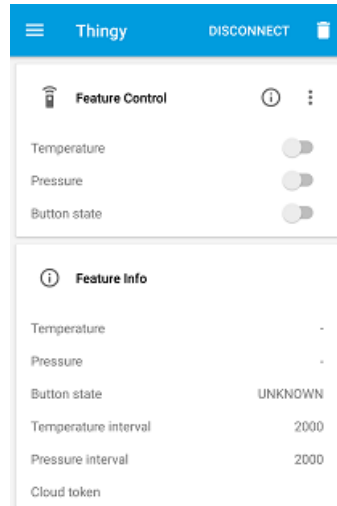
- Samples mode - Thingy comes bundled with short sample sound effects, stored in Thingy's firmware. Tap buttons 1-9 to directly play the sound samples stored in Thingy. Note that this is not actually streaming, as the audio files are on the device already.

### 5.2.5 Cloud

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, tap Info ⓘ in the upper right corner of the Cloud service view. A short tutorial is displayed on how to obtain the Cloud Token that is necessary for IFTTT integration. Paste this token in the **Add / Edit IFTTT token** configuration field that is available under the Action ⋮ icon.

The Cloud view is divided into the following sections.



### Feature Control

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**.

### Feature Info

If a particular type of event from the **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 B.



## 5.3 Thingy Configuration

The Configuration service contains a **Basic** tab, where you can configure the Thingy peripheral, and an **Advanced** tab where you can configure notification time intervals for each sensor.

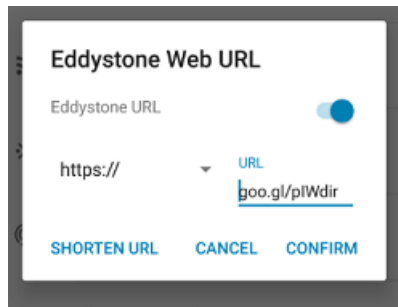
### Basic Configuration

You can configure the following settings found in the **Basic** tab:

- Name - Sets the name used when advertising the Thingy device.
- Advertising parameters
  - Advertising interval - How often Thingy advertises.
  - Advertising timeout
- Connection parameters
  - Minimum connection interval - The minimum interval in milliseconds at which the mobile device asks for data from Thingy.
  - Maximum connection interval - The maximum interval in milliseconds at which the mobile device asks for data from Thingy.

- Slave latency - This is the number of events between when the mobile device asks Thingy for data and when Thingy actually sends 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.
- Supervision timeout - A timeout 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 timeout to detect it quickly.
- Eddystone Beacon URL - Stores a URL in the Eddystone URI format. 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.

**Note:** 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.



- Cloud Token - Set this characteristic to store a token for any cloud service.
- Firmware version - Displays the firmware version.

### Advanced configuration

Here, you can change the interval settings of the Thingy notification updates for Environment and Motion sensors. 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.

## 5.4 Managing Thingy devices


The mobile application supports connections to multiple Thingy devices, meaning you can have more than one Thingy configured in the app. The Android OS imposes a limit of seven devices that can be concurrently configured.

When browsing through the different service views within the app, you can select a Thingy from the main application menu to view the data obtained by that particular Thingy. If multiple devices are configured in the app, it automatically switches to the next Thingy on the list. If not, the app will display the **No Thingy configured** view.

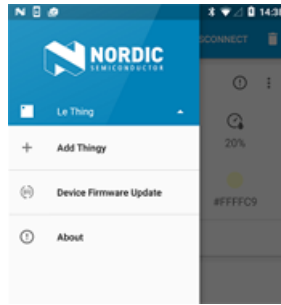
### 5.4.1 Add or remove a Thingy

Adding or removing a Thingy is done in the main application menu.

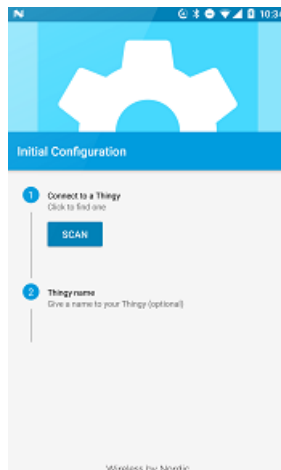
#### Adding a Thingy

1. From the top left of the screen, open the main application menu. Tap the arrow  next to the Thingy name to switch to the menu section that allows for device management.
2. Tap **Add Thingy**.

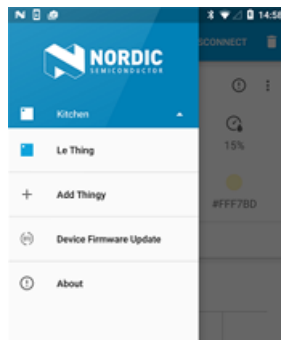





3. Tap **Scan** in the Initial Configuration to scan for a nearby Thingy. Once you have discovered the one you want, you can connect to it, add a name, and share the location. The procedure is the same as during the first launch. You can tap X any time to cancel the process.

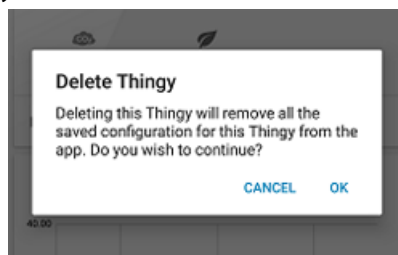


Once you add a Thingy to your app, the main menu will contain a new device. The currently selected Thingy is always shown at the top of the list.



### Removing a Thingy

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



### 5.4.2 Connect or disconnect a Thingy

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.

#### Connecting and disconnecting a Thingy device

1. 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's connected.
2. 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's not connected.

## 5.5 Over-the-air device firmware update (OTA-DFU)

The mobile app provides support for over-the-air device firmware updates. These allow for seamless updates to the Thingy firmware application through the Thingy mobile application.

#### OTA-DFU outline

When there is a feature addition 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 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.

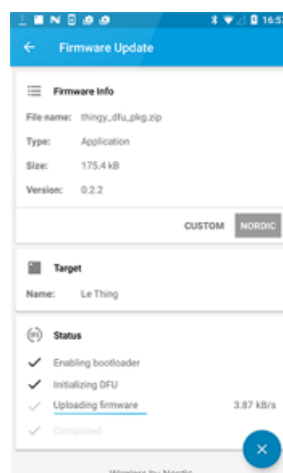
#### Updating firmware on a Thingy that is already connected

This scenario is a default use case.

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

1. Connect to Thingy.
2. Navigate to the Firmware Update menu.
3. 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.

**Note:** 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.

### Updating firmware on a Thingy that is in bootloader mode

This scenario is not a default use case.

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

1. Power Thingy off and then on while holding the button on top of the Thingy to put it into bootloader mode. The device starts advertising with a new name "ThingyDFU". The MAC address is always incremented by 1 when a Thingy device is in bootloader mode.
2. If Thingy is in bootloader mode and not connected yet, tap the blue DFU icon to enter the Scanner view. 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.

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

---

# Chapter 6

## Web app

---

Use the web application to interact with your Thingy without the need to download and install any software on your mobile device.

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

### 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

### Accessing the web app

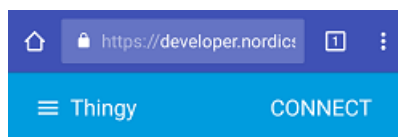
The default way to open the Thingy web application is through the Eddystone beacon that is advertised by Thingy. Use an app that allows for detecting nearby beacons, such as the Physical Web app, and go to the URL that is advertised by Thingy. On Android devices, you can also use the built-in Nearby functionality. Make sure that the link is opened by a supported browser.

An alternative way to access the app is to open the URL <https://developer.nordicsemi.com/thingy/52/> in a supported browser.

## 6.1 Connecting Thingy

After opening the web application in a browser, you must connect Thingy to your mobile device.

1. Tap Connect to scan for nearby Thingy devices.



2. From the list that is displayed, select the Thingy that you want to connect to and tap **Pair**. If Thingy does not appear in the list within a couple of seconds, check the following:
  - a) Is Thingy powered on? The Power button is located under the rubber cover, to the left of the USB port.

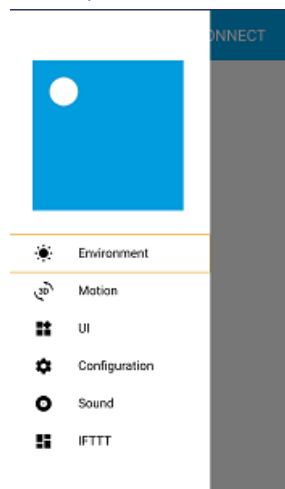
- b) Is Thingy connected to another mobile device? By default, the LED will breathe in cyan color to indicate that it is already connected to another device.
- c) Is Thingy in sleep mode? Press the button or shake the device to make sure that it is not in sleep mode.
- d) Has Thingy run out of battery? Connect Thingy to a charger. Make sure that the power switch is on when charging.

## 6.2 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.

**Note:** The services listed in this chapter are available in the Thingy app. They do not reflect the exact service names or number on the device.

The services offered in the Thingy web app are divided into Environment, Motion, User Interface (UI), Configuration, Sound, and IFTTT (If This Then That).

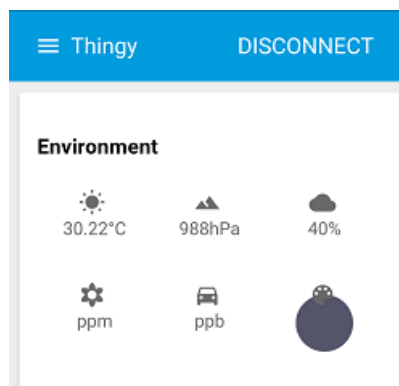


### 6.2.1 Environment

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

#### Environment

In the Environment section, you can see the most current data for all enabled features.



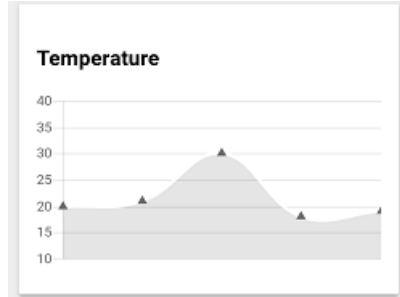
From left to right, the symbols represent the following environmental data:

- Temperature
- Pressure
- Humidity

- CO2
- TVOC (total volatile organic compounds)
- Color intensity

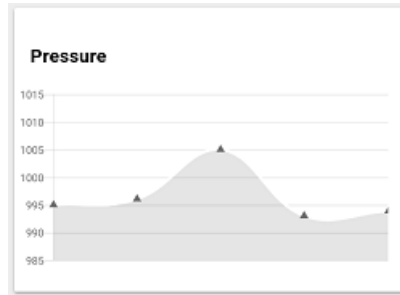
**Temperature**

This section contains a live graph that represents the current and historical temperature measurements.



**Pressure**

This section contains a live graph that represents the current and historical pressure measurements.



**6.2.2 Motion**

The Motion service displays the pedometer (step counter), tap sensor, accelerometer, and other motion-related sensor data received from Thingy. It also displays a 3D model of your Thingy.

**3D**

In the 3D section, you can see a simulation of the current rotation attitude of the Thingy on a 3D model in the app. Rotating the Thingy on any of the axes will immediately be reflected onto the simulation.

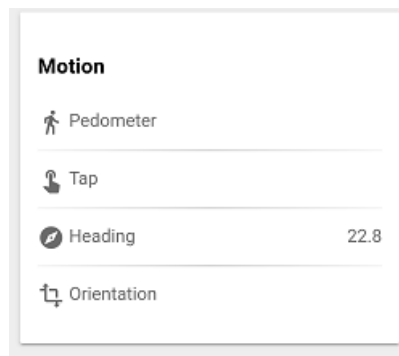


The simulation uses Euler angles to rotate the model at a given configured interval. To make the animation smooth, the simulation interpolates between the current and target positions with a duration of the update interval. This makes the animation as smooth as possible.

### Motion

The motion characteristics in this section come from all motion-related sensors.

- Pedometer (Step counter) - A standard step counter 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.)

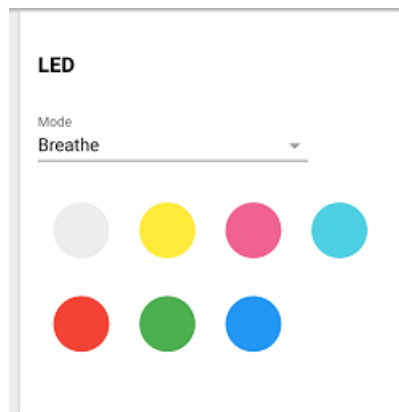


### 6.2.3 User interface

The User Interface service allows you to interact with the LED and button on Thingy.

#### 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.



In Off mode, the LED is turned off completely and Thingy will remain in Off mode while disconnected. There is no configuration for this mode.

In Constant mode the LED stays on, but once disconnected the mode turns off.

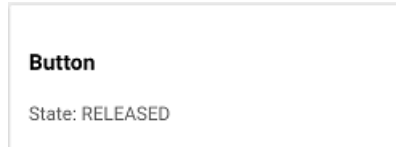
Breathe mode is where the Thingy LED fades in and out at a given interval, but turns off after Thingy is disconnected.

In one-shot mode, the Thingy LED fades in then out once, then remains off indefinitely. This mode does not persist while disconnected.

## Button

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

- Pressed - This is displayed when you press and hold the button.
- Released - This is displayed when you release the button.

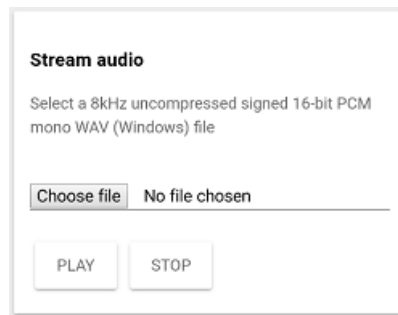


## 6.2.4 Sound

The Sound service lets you control the Thingy's speaker.


### Stream audio

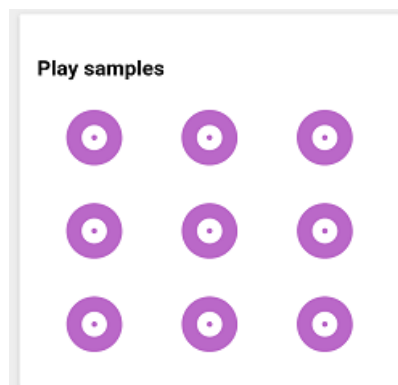
Thingy is capable of streaming 8-bit PCM audio from the mobile device to its speaker. Choose the file located on your mobile device and tap **Play** to start streaming.



**Note:** 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.

### Play samples

Thingy comes bundled with short sample sound effects, stored in Thingy's firmware. Tap the  icons to directly play the sound samples stored in Thingy. Note that this is not actually streaming, as the audio files are on the device already.





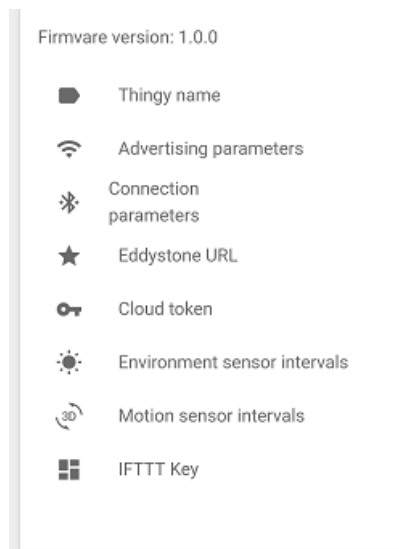
## Play tones

This is a frequency mode in which you can tap the  icons to play notes of a certain frequency. Only one note can be played at a time.



## 6.3 Thingy Configuration

The Configuration service allows you to customize various Thingy settings.



You can configure the following settings of your Thingy:

- Firmware version - Displays the firmware version.
- Thingy name - Sets the name used when advertising the Thingy device.
- Advertising parameters
  - Advertising interval - How often Thingy advertises.
  - Advertising timeout
- Connection parameters
  - Minimum connection interval - The minimum interval in milliseconds at which the mobile device asks for data from Thingy.
  - Maximum connection interval - The maximum interval in milliseconds at which the mobile device asks for data from Thingy.
  - Slave latency - This is the number of events between when the mobile device asks Thingy for data and when Thingy actually sends 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.

- Supervision timeout - A timeout 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 timeout to detect it quickly.
- Eddystone URL - Stores a URL in the Eddystone URI format. 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.
- Cloud Token - Set this characteristic to store a token for any cloud service.
- Environment sensor intervals - Interval settings of the Thingy notification updates for environment sensors.
- Motion sensor intervals - Interval settings of the Thingy notification updates for motion sensors.
- IFTTT key - Paste the IFTTT key in this field to integrate your Thingy with the IFTTT web service.

## 6.4 IFTTT

You can integrate your Thingy with the IFTTT (If This Then That) web service.

**IFTTT**

Key:

Event 0/20

---

**Instructions**

1. Save your IFTTT key in Configuration
2. Type an event to trigger
3. Press the Thingy button to trigger

Before you can start using this service, you must register with IFTTT on its website and obtain a Maker Webhooks access key. Paste this key in [Thingy Configuration](#) on page 49.

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

# Chapter 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 Hardware figures

The following figures present various elements of the Thingy PCB.

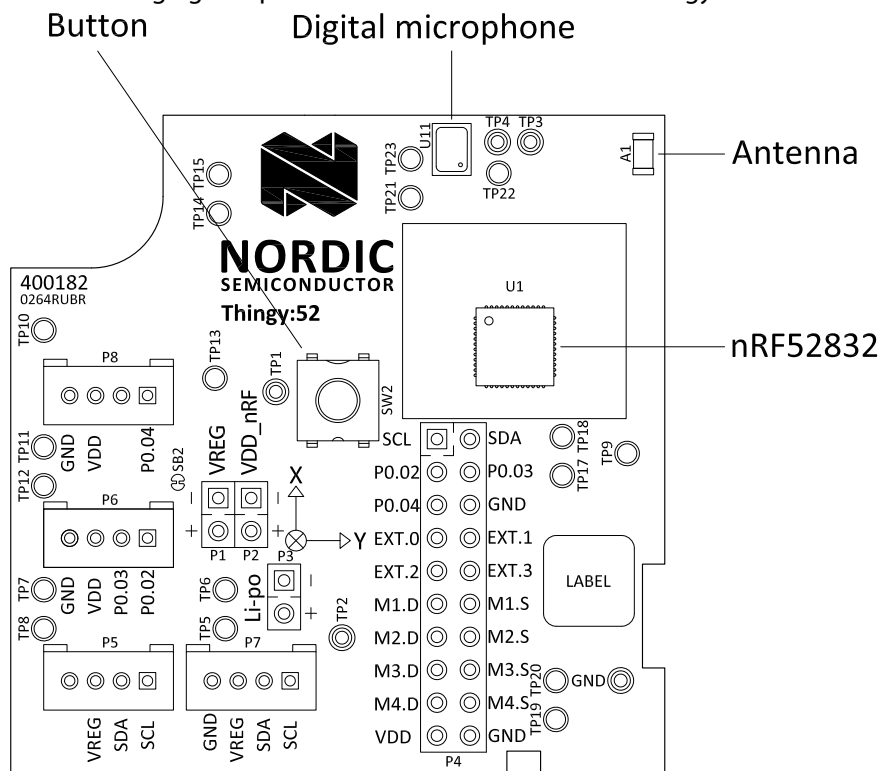
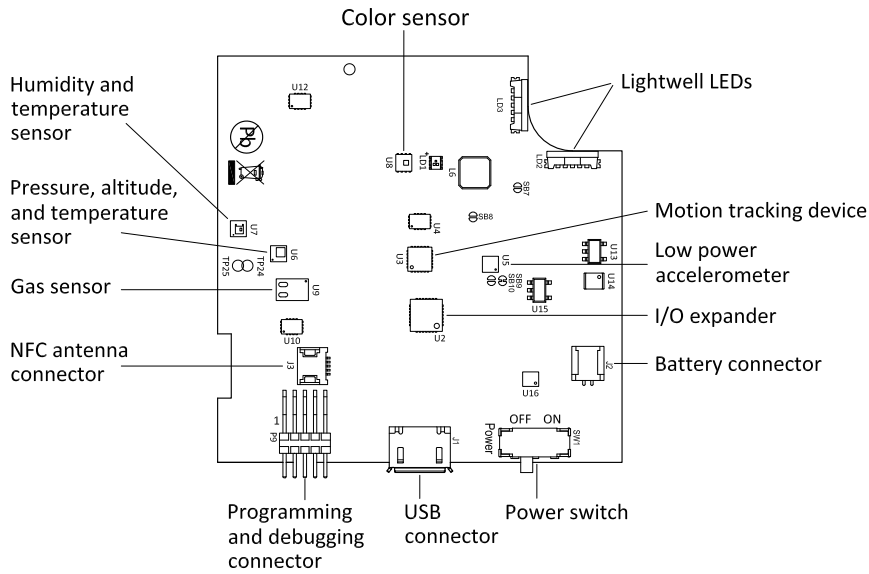


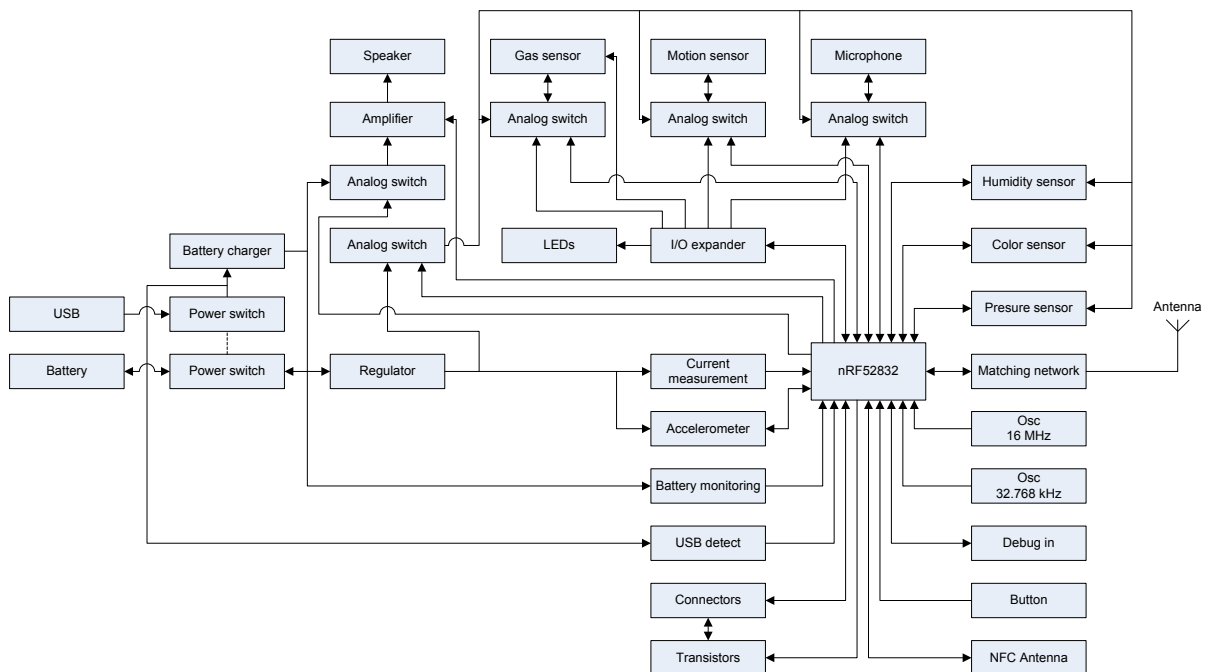
Figure 5: Thingy PCB, top



**Figure 6: Thingy PCB, bottom**

## 7.2 Block diagram

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



**Figure 7: Thingy hardware block diagram**

### 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*<sup>®</sup> low energy radio and a 32-bit ARM<sup>®</sup> Cortex<sup>®</sup>-M4F CPU.

The nRF52832 features 512 kB of flash memory and 64 kB of RAM. It supports DSP instructions, Floating Point Unit (FPU), and energy efficient processing of computationally complex operations, making it a great basis for Thingy. The nRF52832 SoC supports NFC-A tag which can be used for Out-of-Band (OOB) pairing between two *Bluetooth*<sup>®</sup> devices. The SoC has 32 GPIOs, most of which are used internally on Thingy. However, some of them are available externally on the connectors for the user to connect additional hardware for custom usage.

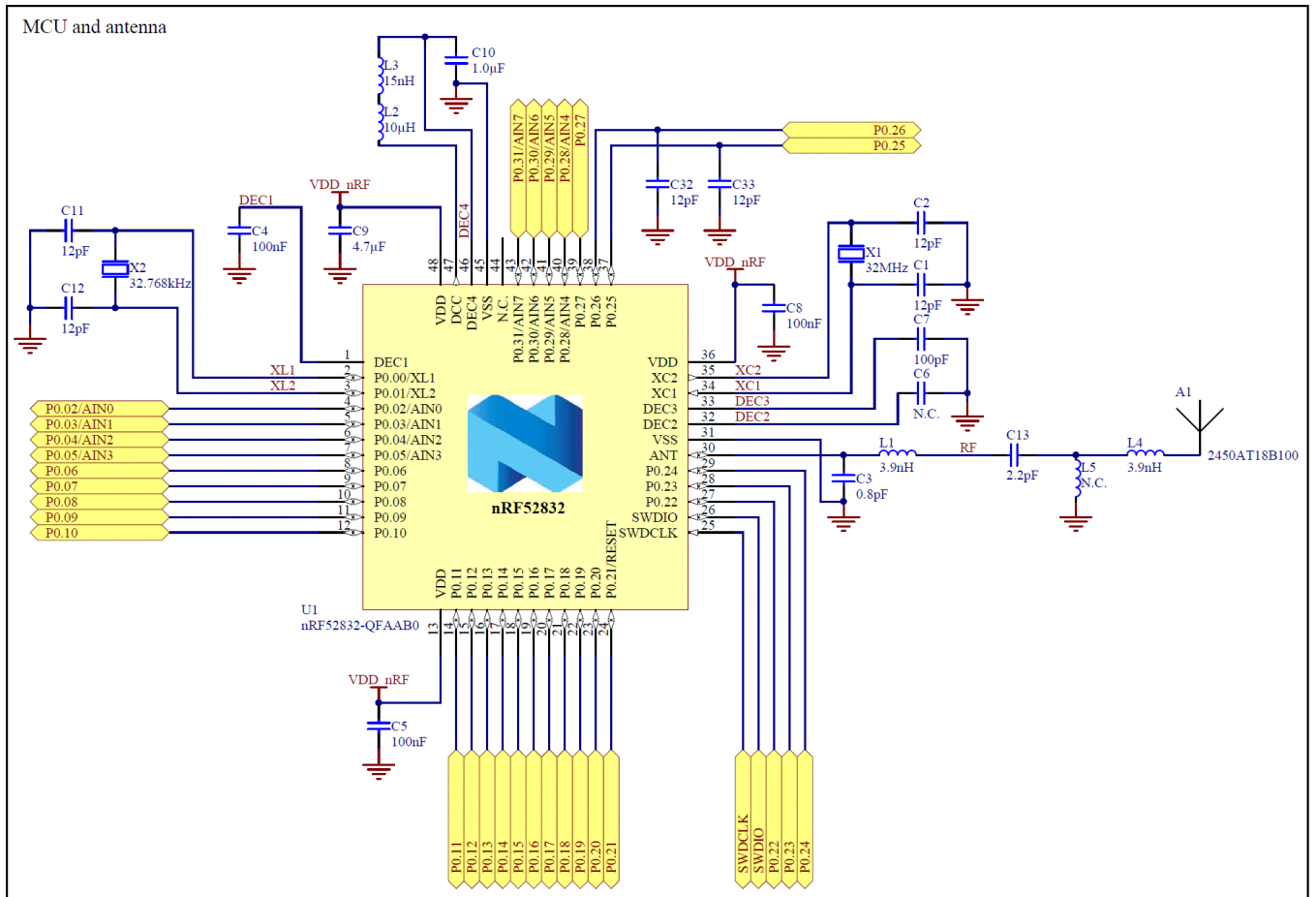


Figure 8: MCU schematic

### 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.

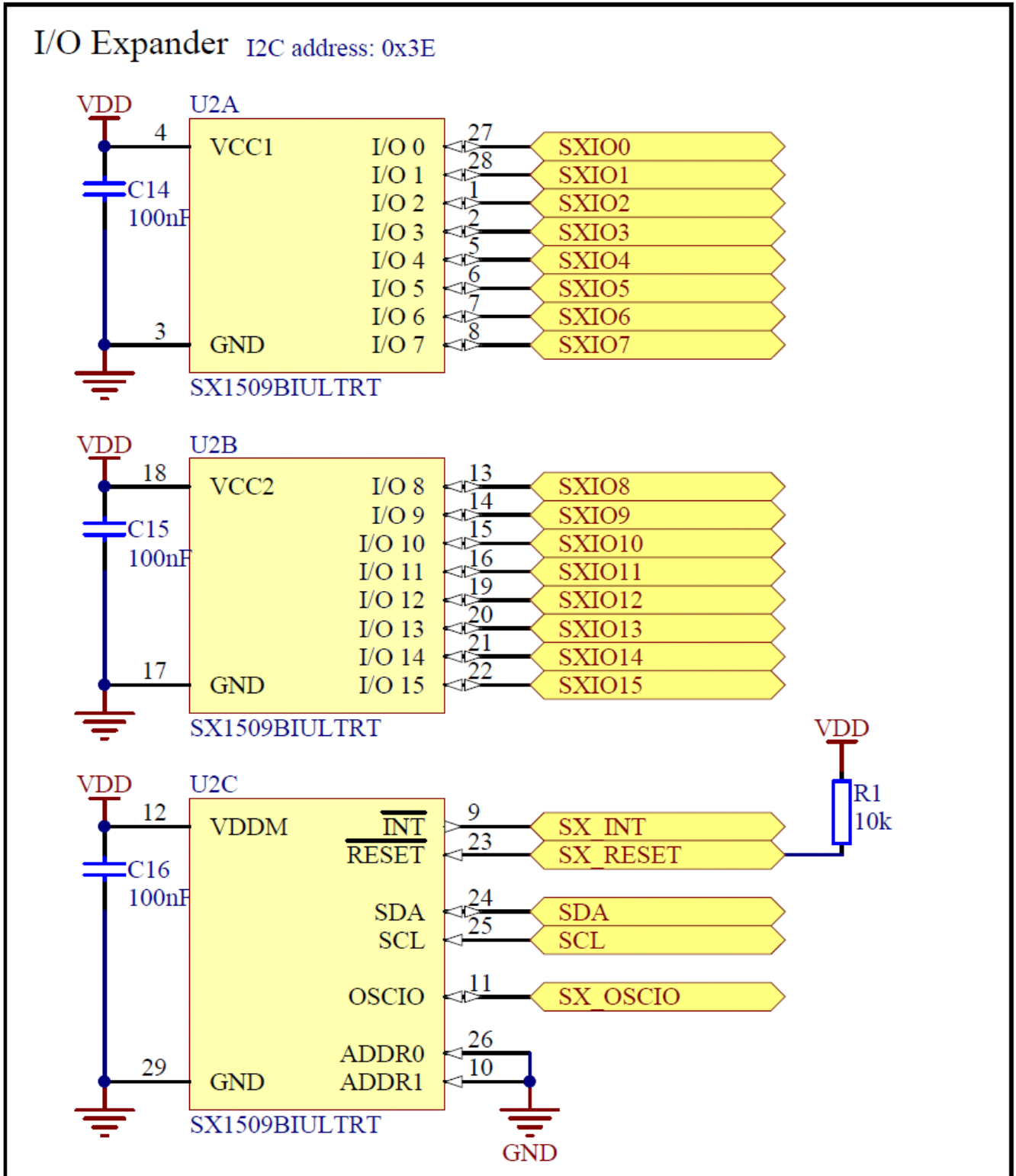


Figure 9: I/O expander schematics

### 7.5 Pin maps

The following tables present the pin assignments for the nRF52832 SoC and for the I/O expander.

**Table 1: nRF52832 pin map**

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
P0.07	SDA	I2C data line
P0.08	SCL	I2C 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 I2C data line
P0.15	SCL_EXT	External and low power accelerometer I2C 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

I/O	Label	Description
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 2: I/O expander 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

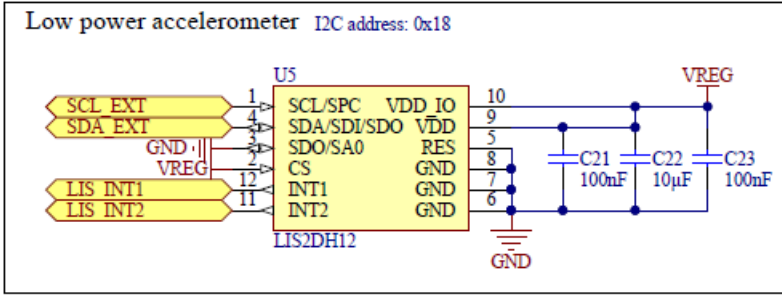
## 7.6 Motion sensors

Thingy includes a low power accelerometer and a 9-axis motion sensor.

### Low power accelerometer

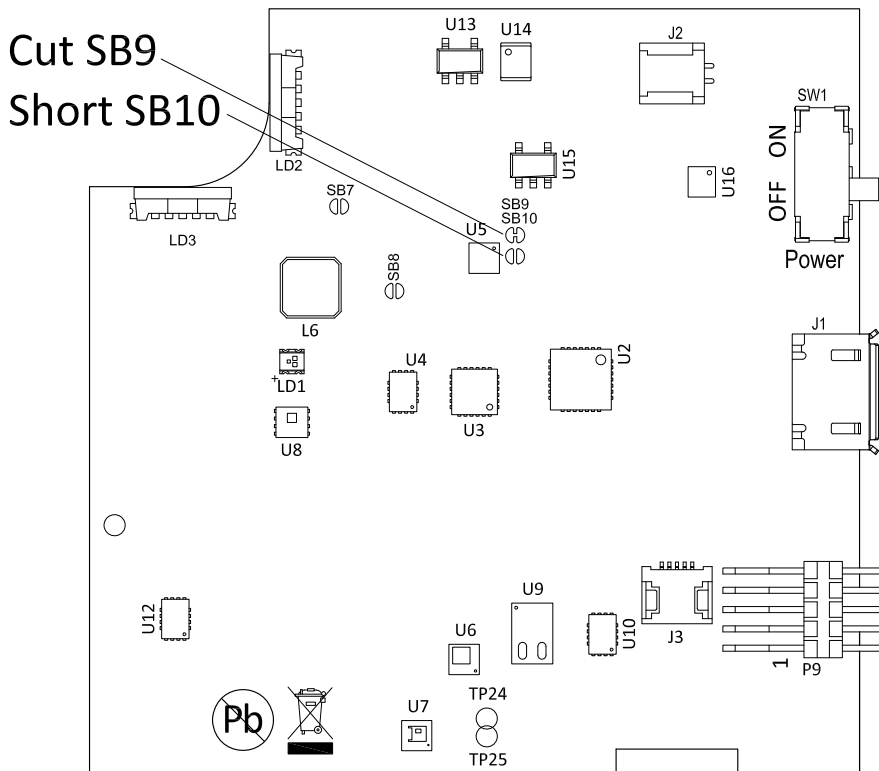
A dedicated low power accelerometer is used to wake Thingy if it is asleep. The accelerometer features an I2C interface and can detect motion on three axes. It is connected directly to the voltage regulator output so it is powered regardless of the VDD power control (see [Power supply](#) on page 62 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).





**Figure 10: Low power accelerometer schematic**

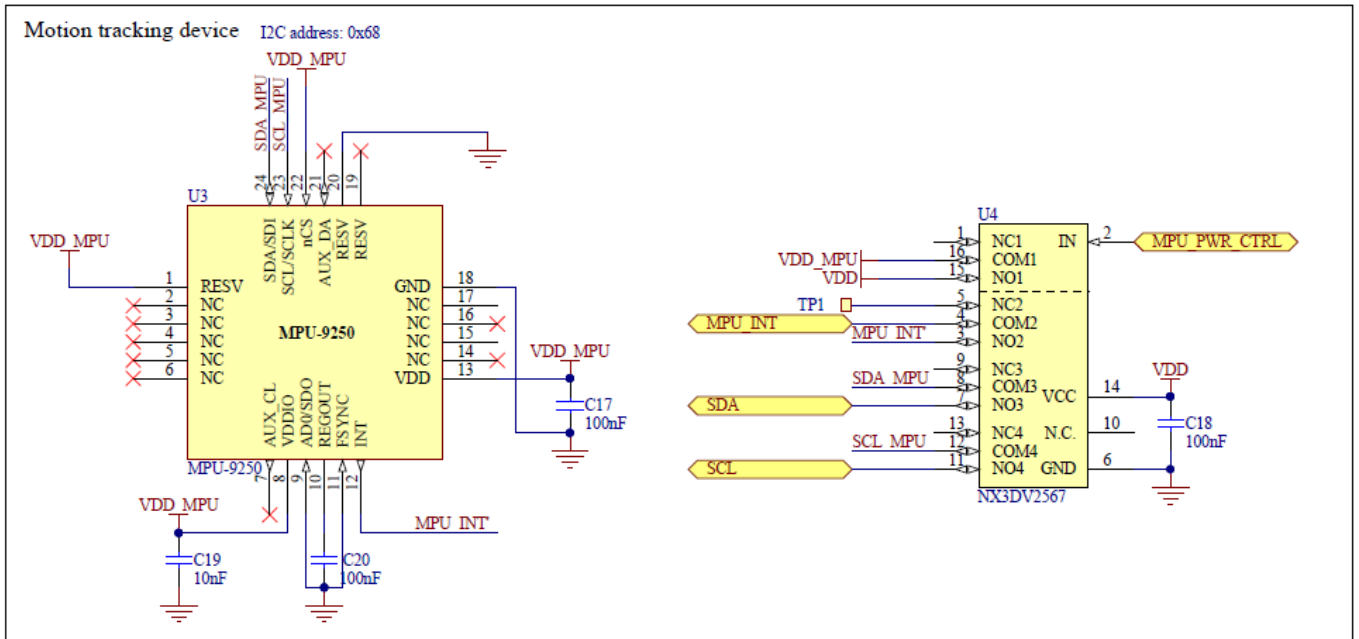
By default the INT1 line of the accelerometer is connected to nRF52832. If you want to use the INT2 line instead, cut the short on **SB9** and solder **SB10**.



**Figure 11: Low power accelerometer interrupt line selection**

### 9-axis motion sensor

For advanced motion tracking features, Thingy uses a 3-axis gyro, 3-axis accelerometer, and 3-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**.



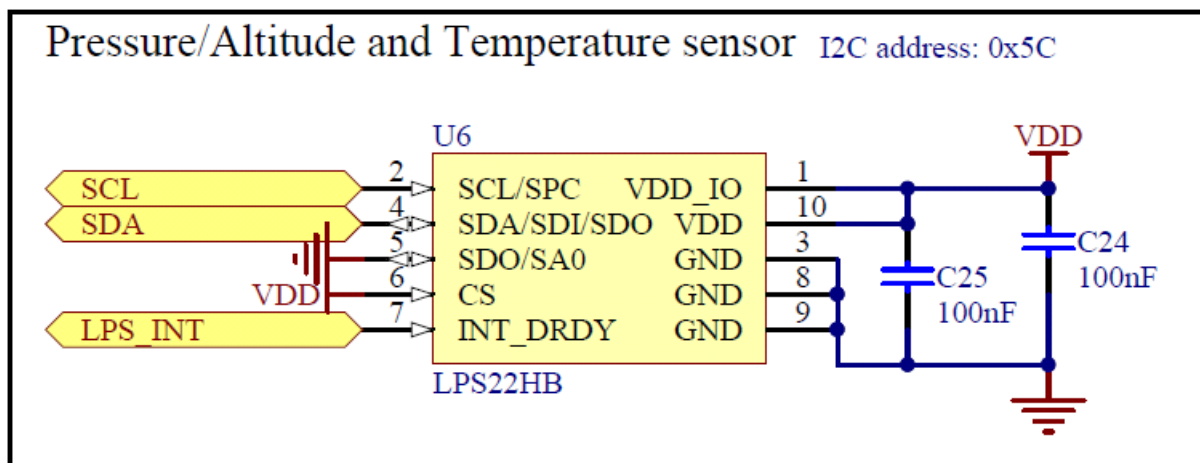
**Figure 12: Motion sensor schematic**

## 7.7 Environment sensors

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

### Pressure sensor

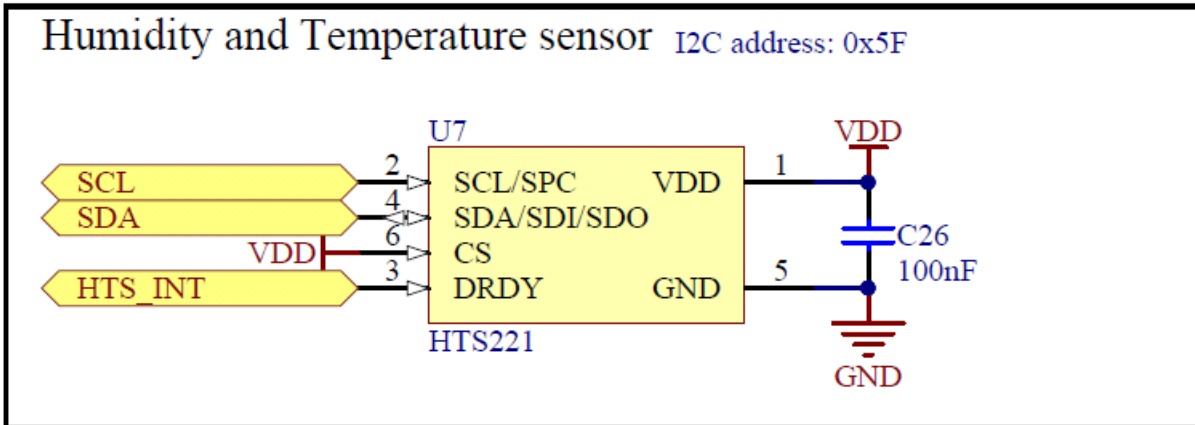
The pressure sensor (**U6**) onboard Thingy is capable of measuring 260 to 1260 hPa absolute pressure with 24-bit data output. The sensor has built-in temperature compensation, and as a secondary function, it provides 16-bit temperature data output. The pressure sensor is interfaced through I2C (slave address 0x5C).



**Figure 13: Pressure sensor schematic**

### Humidity sensor

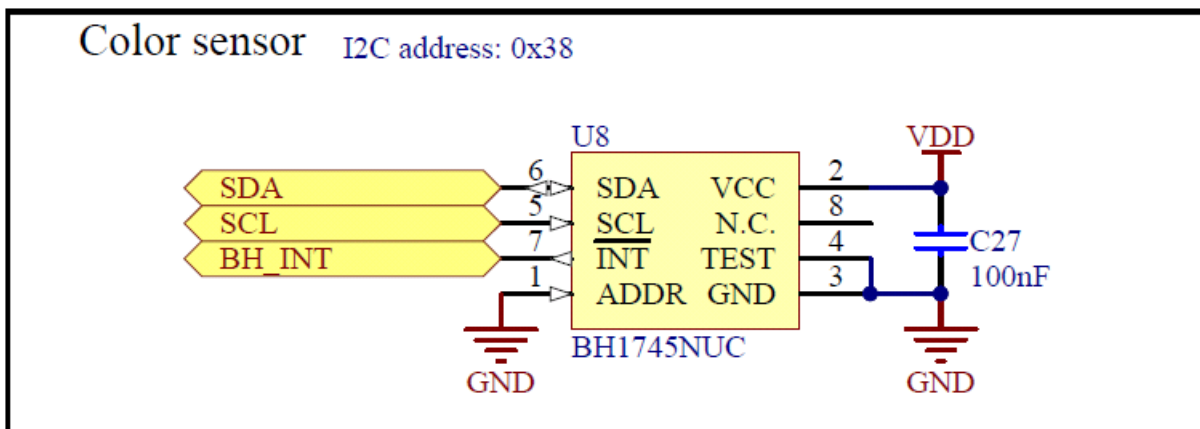
For measuring humidity, Thingy has a combined humidity and temperature sensor onboard (**U7**). It has a 0 to 100% relative humidity range and provides 16-bit humidity and temperature data output. The humidity sensor accesses the MCU through I2C (slave address 0x5F).



**Figure 14: Humidity sensor schematic**

**Color sensor**

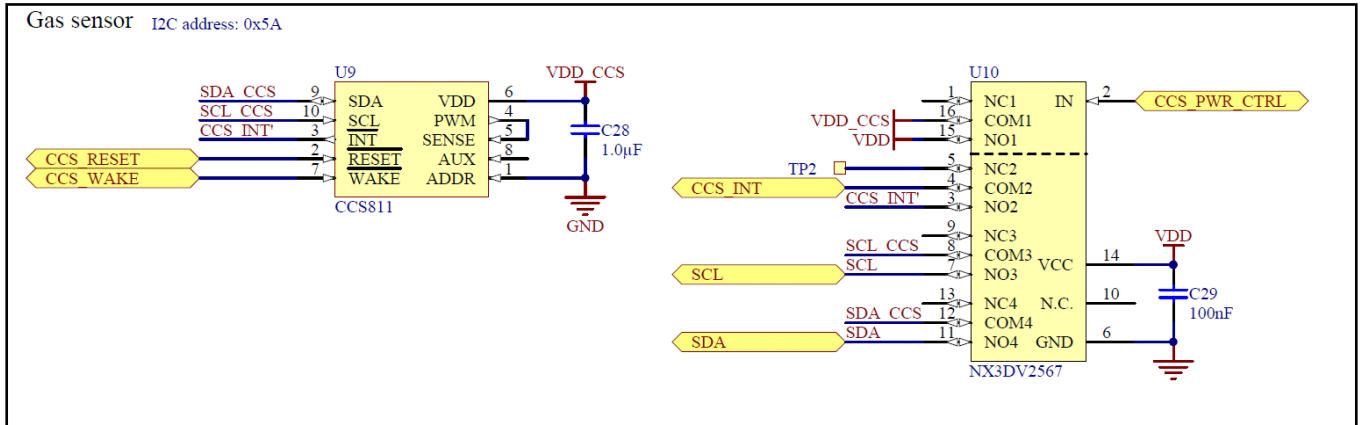
The color sensor (**U8**) onboard Thingy senses red, green, and blue light with a 0.005 – 40k lx dynamic range. The sensor faces towards the blue transparent bottom case with light pipes guiding the light towards the sensor. To measure the color on a surface, the color sensor is accompanied with an RGB 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).



**Figure 15: Color sensor schematic**

**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**.



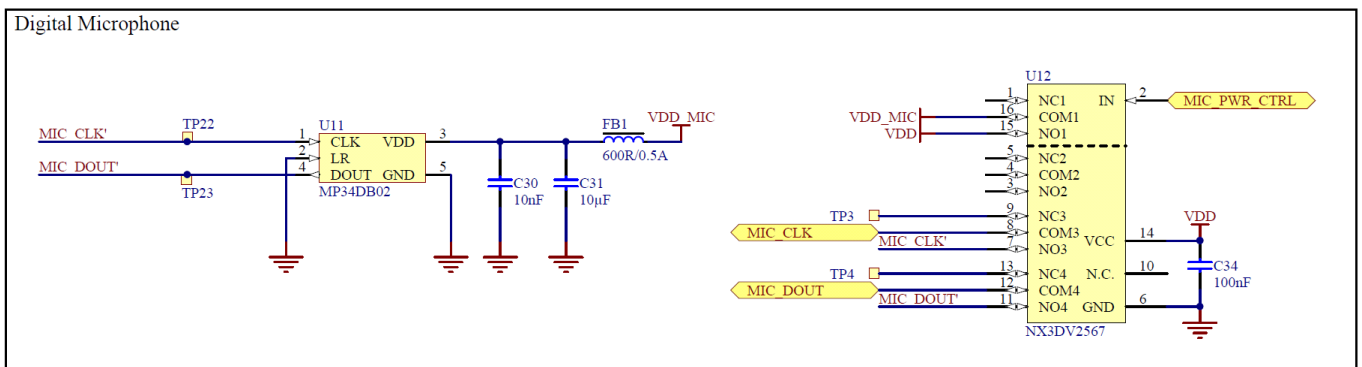
**Figure 16: Gas sensor schematic**

## 7.8 Sound

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

### Digital microphone

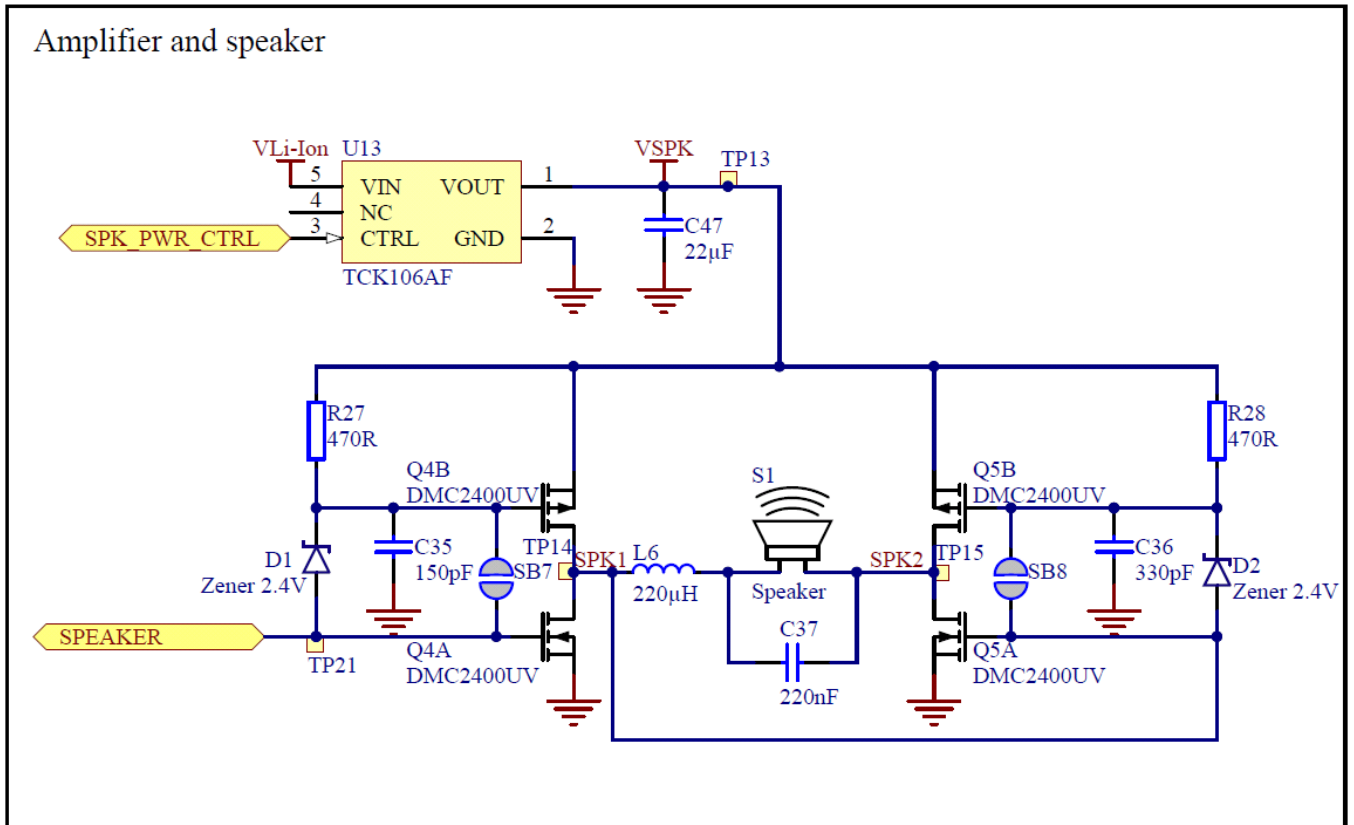
For audio input, Thingy is equipped with a digital output PDM microphone (**U11**). The microphone's power and signal lines are routed through an analog switch (**U12**) to save power when the microphone is not in use. When the microphone is switched off, the GPIOs used for the PDM signal are available on two test points (**TP3** and **TP4**). This allows the user to connect other hardware if the microphone is not in use.



**Figure 17: Digital microphone schematic**

### 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.



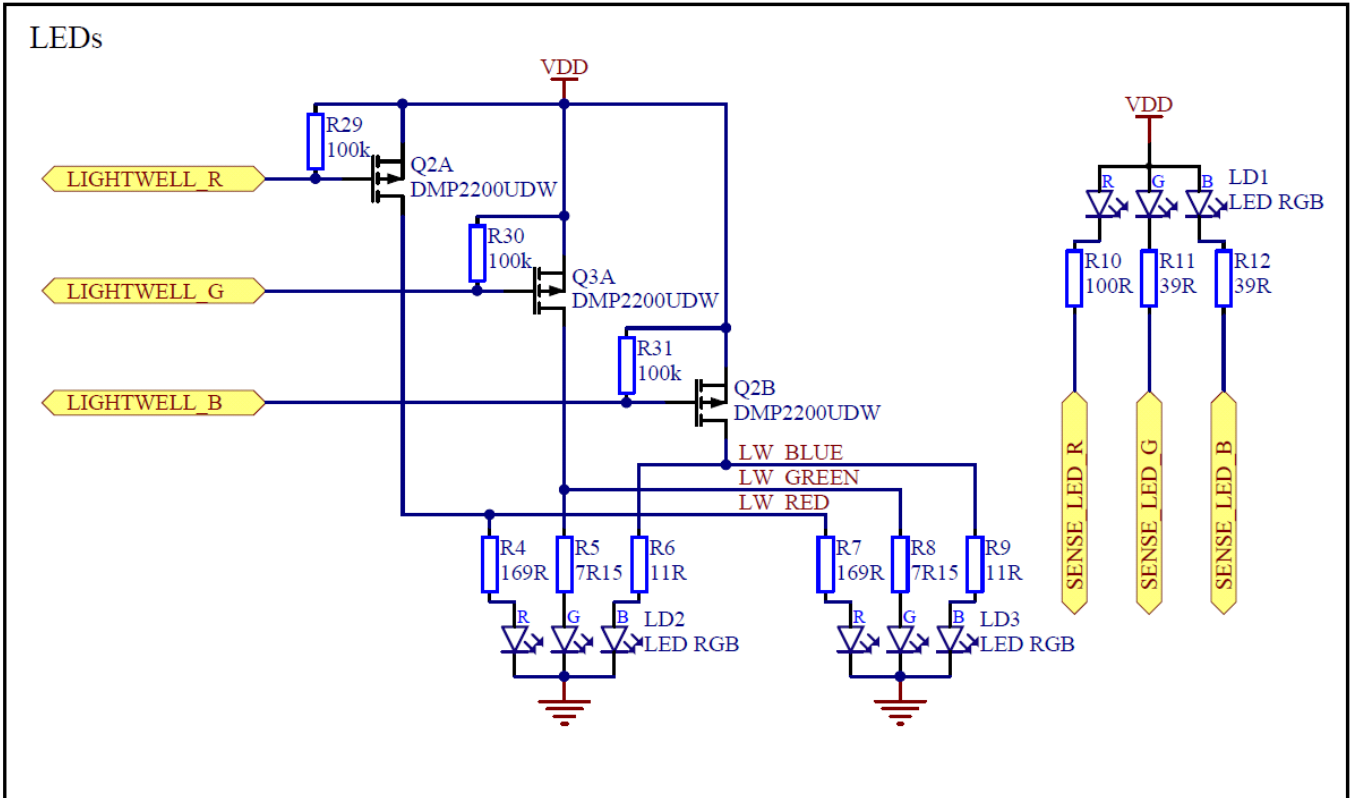
**Figure 18: Amplifier and speaker schematic**

## 7.9 LEDs and button

Thingy's user interface consists of RGB LEDs and a button.

### LEDs

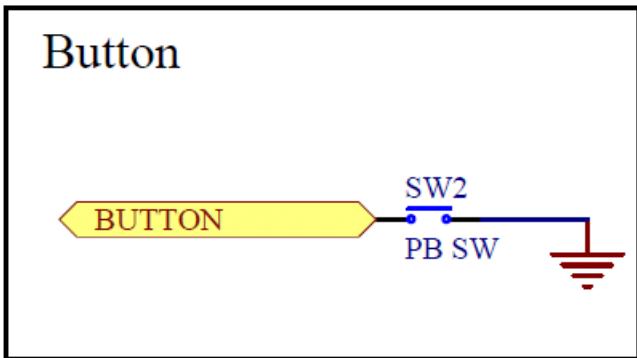
Thingy is equipped with three RGB LEDs. Two of the LEDs are used to light up the light well and are controlled by the same signals using transistors as switches. The third LED is located near the color sensor and is used as auxiliary light for color measurements.



**Figure 19: LED schematic**

**Button**

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

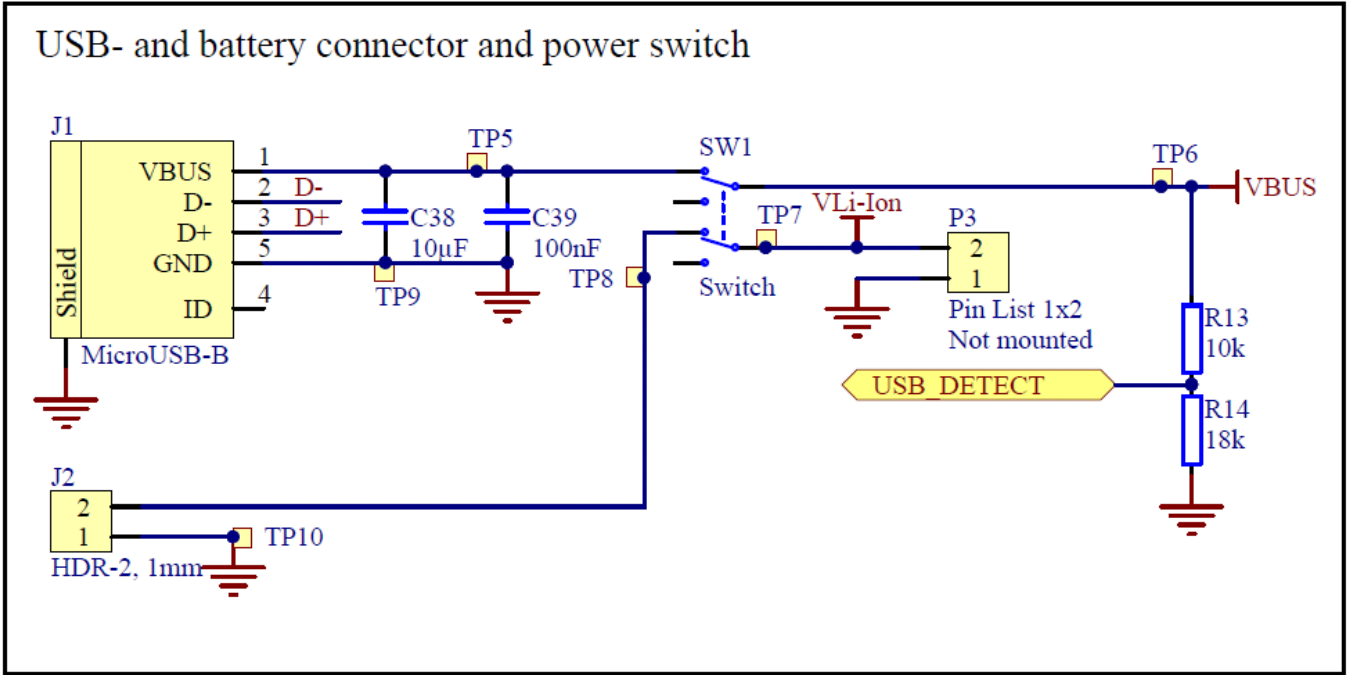


**Figure 20: Button schematic**

**7.10 Power supply**

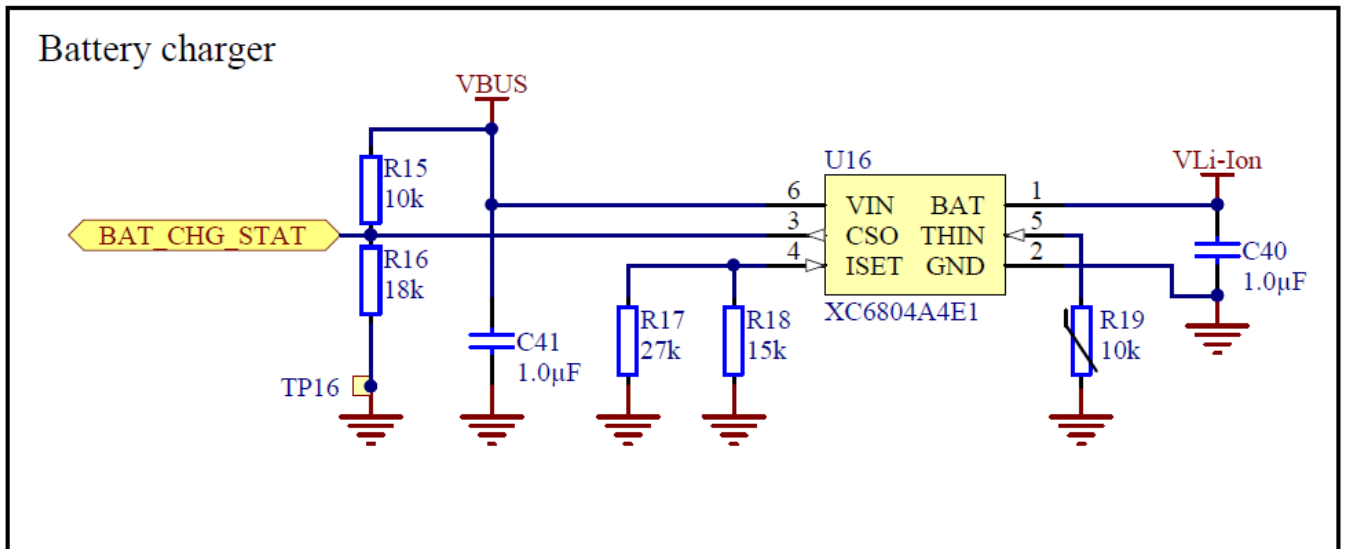
This section contains information about the power supply circuitry of Thingy.

A rechargeable Lithium-ion Polymer battery is used as the main power source. The battery has a nominal capacity of 1440 mAh and 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.



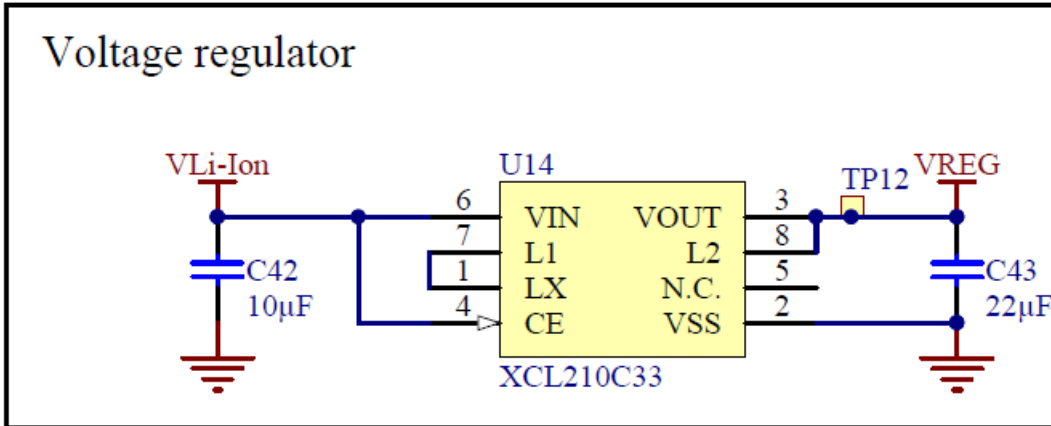
**Figure 21: Schematics for USB and battery connectors, and the power switch**

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.



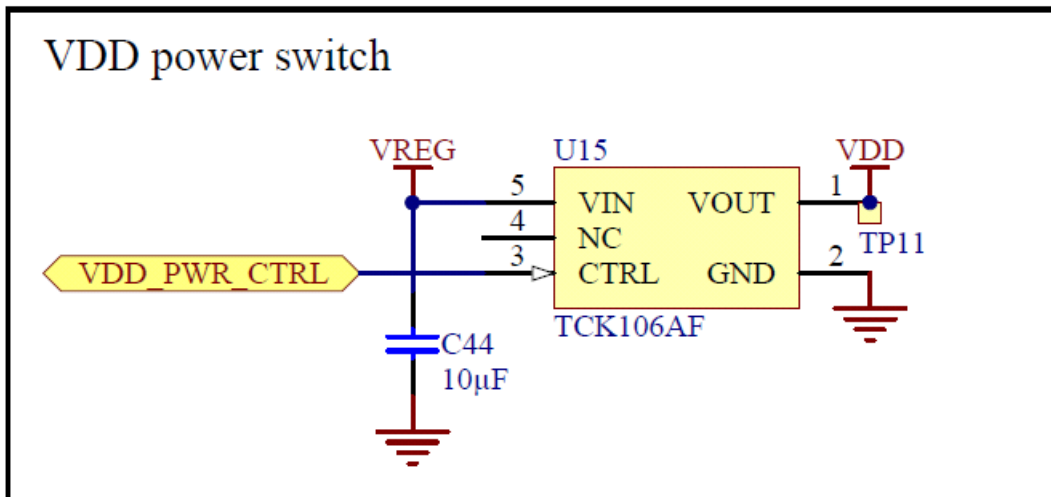
**Figure 22: Battery charger schematic**

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



**Figure 23: Voltage regulator schematic**

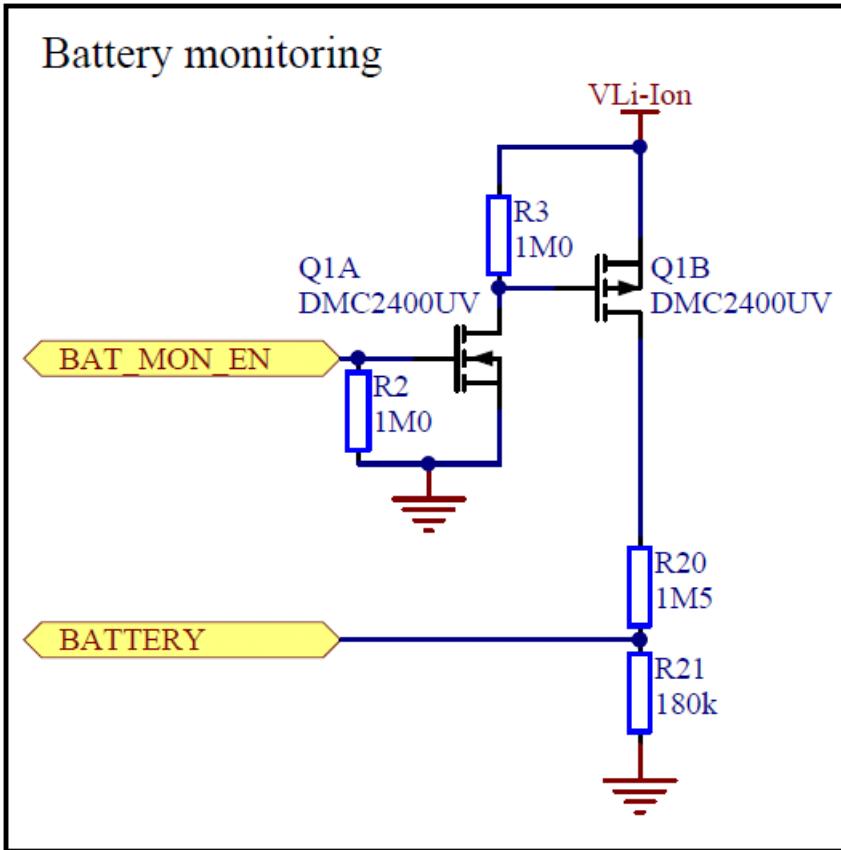
To save power during sleep mode, an analog switch is used to turn off the VDD power net. The low power accelerometer and nRF52832 are powered directly from VREG and will always be powered to wake up Thingy.



**Figure 24: VDD power switch schematic**

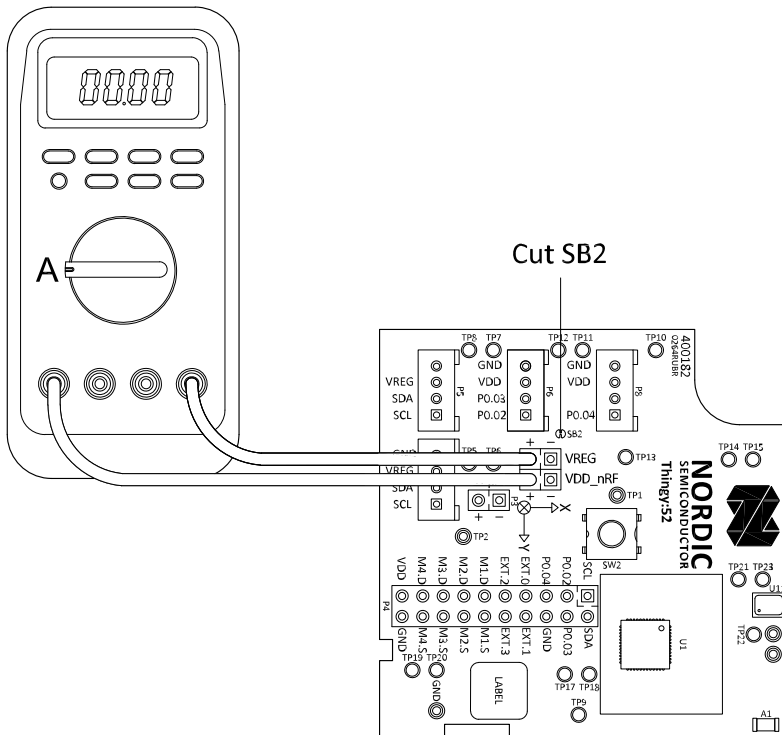
To monitor the battery voltage, a voltage divider circuit is connected to the battery. nRF52832 will use the ADC to read out the voltage level. To avoid unnecessary current drain, the voltage divider is turned off with a transistor when not in use.



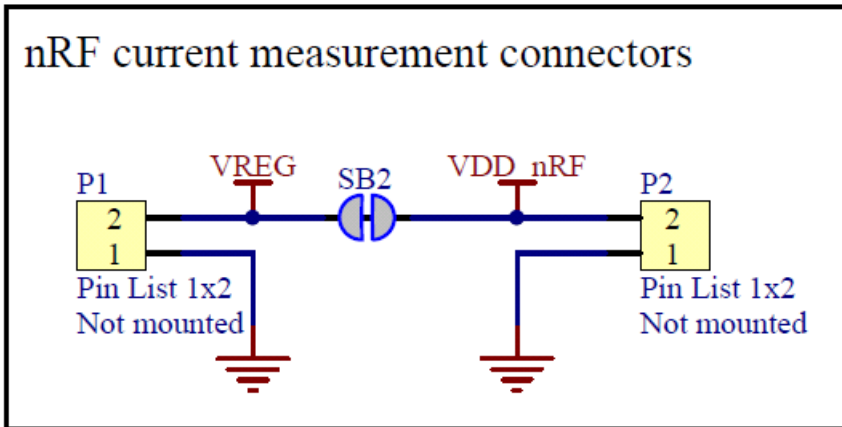


**Figure 25: Battery monitoring schematic**

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



**Figure 26: Measuring current to the nRF52832**

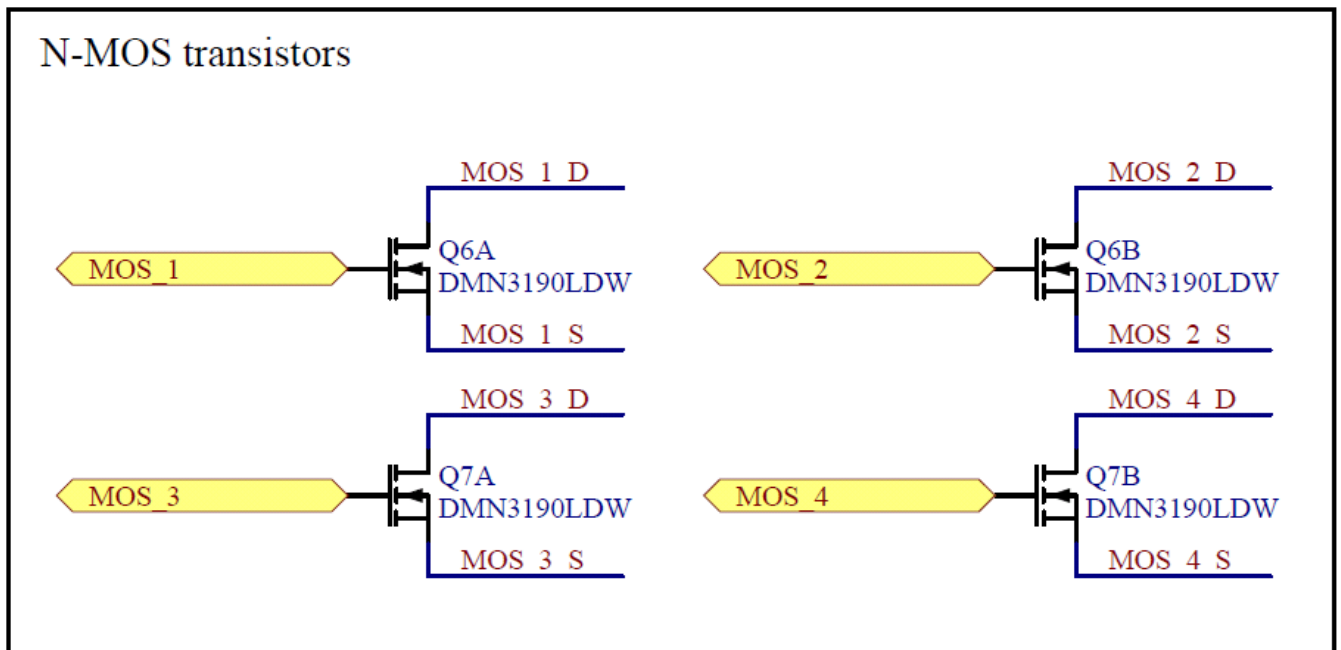


**Figure 27: Current measurement connectors schematic**

### 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.

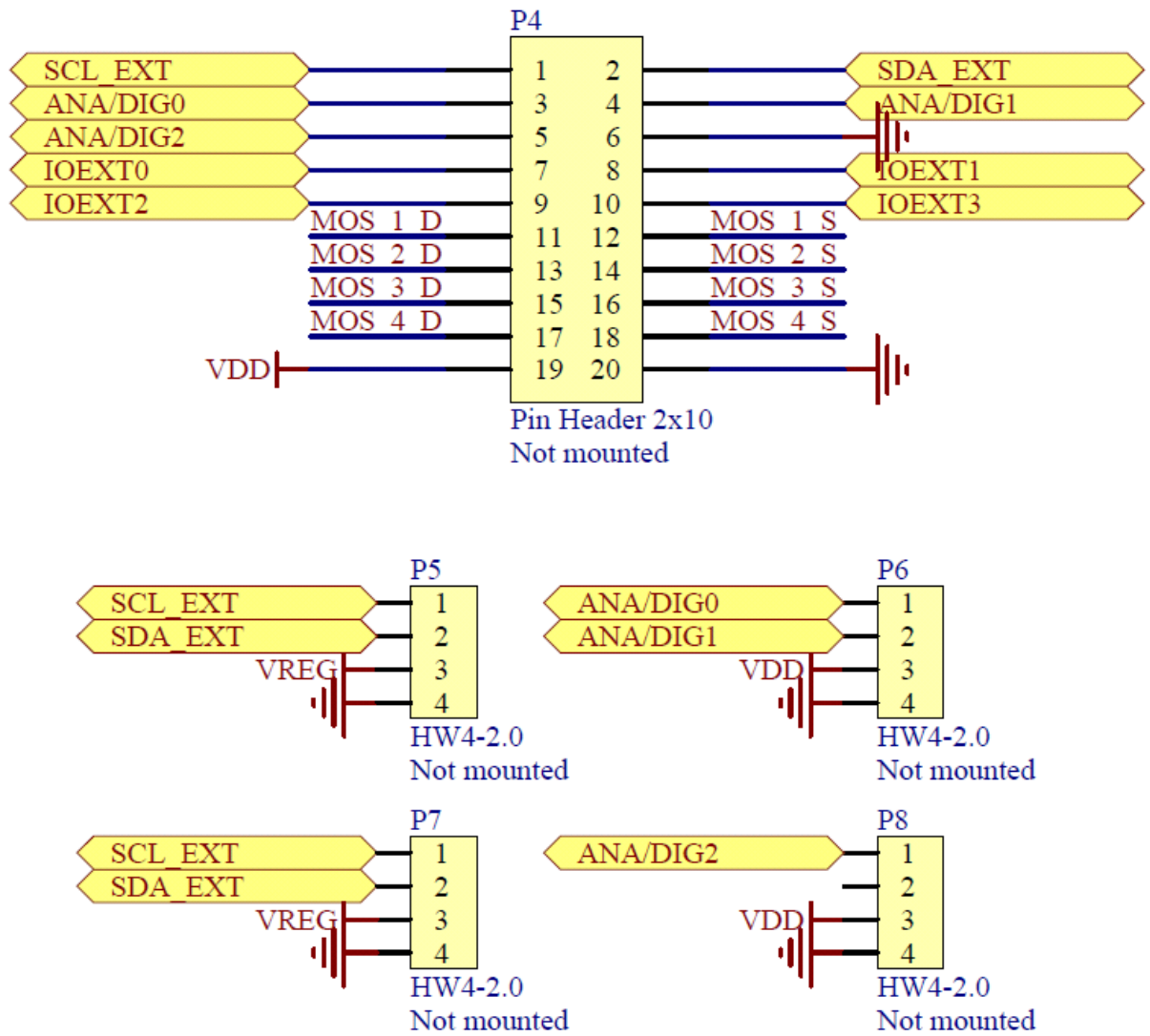
For more flexibility, Thingy is equipped with four N-MOS transistors that can be used to drive small DC motors or LEDs. The drain and source of the transistors are available on external connectors and the gate is connected directly to the nRF52832.



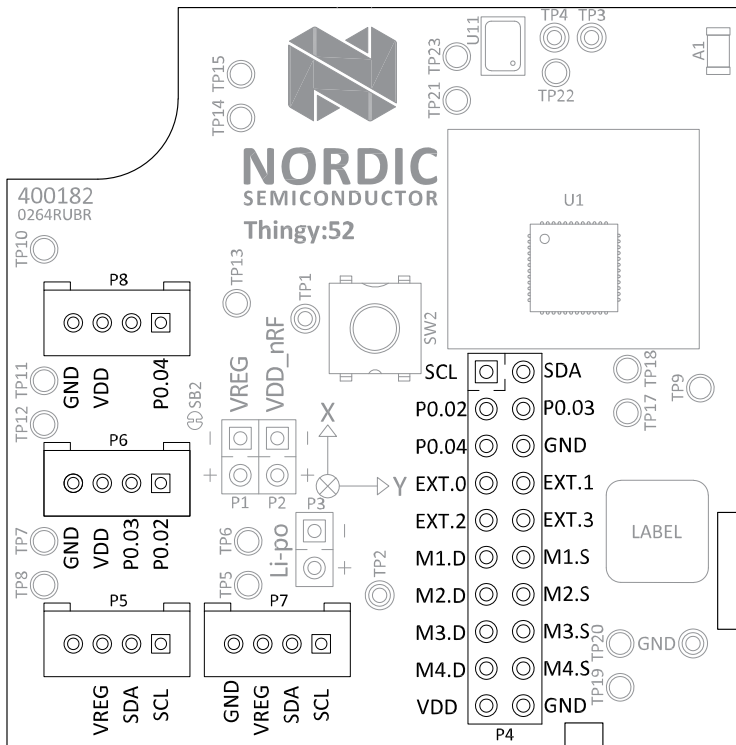
**Figure 28: Schematics for the N-MOS transistors**

Extra GPIOs can be found on connector **P4** (see [Pin maps](#) on page 54). In addition to **P4**, some of the GPIOs are available on connectors **P5-P8**. 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.

## Interface connectors



**Figure 29: Interface connectors**



**Figure 30: Thingy external connectors**

**Table 3: Pinout of connector P4**

Pin	Signal	Description
1	SCL_EXT	External I2C clock
2	SDA_EXT	External I2C 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

Pin	Signal	Description
19	VDD	Power supply, controlled by the VDD power switch
20	GND	Ground

**Table 4: Pinout of connectors P5 and P7**

Pin	Signal	Description
1	SCL_EXT	External I2C clock
2	SDA_EXT	External I2C data
3	VREG	Power supply, no power control
4	GND	Ground

**Table 5: Pinout of connector P6**

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 6: Pinout of connector P8**

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

# 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 2002/95/EC of the European Parliament and of the Council* on the Restriction of Hazardous Substances (RoHS) and the requirements of the *REACH* regulation (EC 1907/2006) on Registration, Evaluation, Authorization and Restriction of Chemicals.

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



# Mouser Electronics

Authorized Distributor

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

[Nordic Semiconductor:](#)

[nRF6936](#)