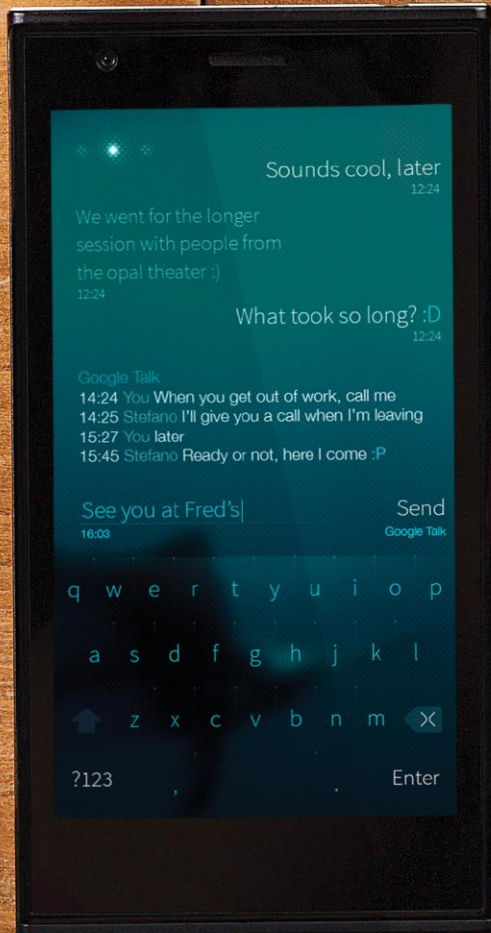




jolla



# THE OTHER HALF

TECHNICAL SPECIFICATION VERSION 1.1 - [WWW.JOLLA.COM](http://WWW.JOLLA.COM)

# TABLE OF CONTENTS

<b>Terms of use</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Functional overview	3
1.2 Contact information	3
<b>2 Hardware specification</b>	<b>4</b>
2.1 Mechanical interfaces and keep out areas	4
2.2 Electrical interfaces	7
2.2.1 I2C	7
2.2.2 Cover power (VDD)	7
2.2.3 External charger 5V DC VIN	8
2.2.4 INT Interrupt GPIO	8
2.2.5 GND	8
2.3 Cover identification interfaces	8
2.3.1 NFC	8
2.3.2 Cover detection switch	9
<b>3 Software architecture</b>	<b>10</b>
3.1 Cover identification	11
3.1.1 The Other Half NFC UID allocation	11
3.2 Hardware control interfaces	12
3.2.1 VDD power control	12
3.2.2 I2C access interfaces	13
3.2.3 INT line handling	13
3.3 Security	14
3.4 Power management	14
<b>4 Application notes</b>	<b>15</b>
4.1 Ambiance cover	15
4.2 External charger cover	17
4.3 Bluetooth cover	18
4.4 Generic I2C device cover	19
<b>5 References</b>	<b>20</b>

## THE OTHER HALF SPECIFICATION COMMUNITY LICENSE

### Copyright

1. Jolla grants you a perpetual, worldwide, non-exclusive, non-sublicensable, no-charge, royalty-free, irrevocable (except as provided in this license) copyright license to reproduce, distribute, display, make available, and translate this specification and to implement the specification and distribute such implementations provided that you do not, directly, or indirectly, charge a fee for your implementations.

### Patents

2. Subject to your continued compliance with the terms of the copyright license, Jolla grants you a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as provided in this license) patent license to make, have made, use, import, and otherwise transfer implementations of this specification where such license applies only to those patent claims, both currently owned by Jolla and acquired in the future, licensable by Jolla that are necessarily infringed by implementation of this specification provided that you do not, directly, or indirectly, charge a fee for your implementations.

### Non-assertion

3. If you, or any third party affiliated (such as via agency, agreement, license, or common ownership or control) with you, threaten or initiate, directly or indirectly via a third party, litigation against any entity alleging that any implementation of this specification (or an adaptation thereof) infringes on your or a third party's rights, then any rights granted to you under this license terminate as of the earliest of the following: i) the date on which you issued a written threat of litigation, or ii) the date on which such litigation is filed.

### Liability limitation

THIS SPECIFICATION, ANY RELATED SOFTWARE AND DOCUMENTATION (REFERRED TO AS “**TOH MATERIALS**”) ARE PROVIDED BY JOLLA “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL JOLLA BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF TOH MATERIALS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1 — INTRODUCTION

This document and attached materials are licensed under The Other Half community license (see “Terms of use”). Based on that license, this specification can be used for evaluating and developing product designs, and creating personal implementations of those designs.

If you wish to sell The Other Half covers / products you have implemented, you must contact Jolla Ltd to agree on licensing terms. See chapter “Contact information“ for more details.

The Other Half exchangeable covers for Jolla devices offer opportunities for a wide array of accessory and application innovations. This document will describe Jolla’s The Other Half hardware, mechanical and software interface specification for third party partners.

The document describes the mechanics for reference The Other Half covers for the Jolla device and the additional electrical functional connections towards The Other Half.

With the information in this document and the mechanics 3D file that comes with it a third party should be able to produce The Other Half functional cover that matches the Jolla device.



## 1.1 — FUNCTIONAL OVERVIEW

The Other Half hardware interface consists of dedicated hardware pin connectors, dedicated NFC tag reader, and a mechanical switch for cover attach / detach detection. The main features supported by the interface are:

- Cover specific content and software download based on NFC UID
- Cover attach / detach detection via cover switch
- Wireless or any other external 5V DC charger
- Any i2c slave smart device
- INT input line for dedicated button or wake up signal functions
- 3,3V power supply for any smart device
- Bluetooth or WLAN connected smart device powered with own battery or via 3,3V VDD pin

## 1.2 — CONTACT INFORMATION

For business, manufacturing, distribution, NFC UID allocation, licensing, and co-development project inquiries, please contact our business development crew at

- [info@jolla.com](mailto:info@jolla.com)

For technical questions please contact

- [developer-care@jolla.com](mailto:developer-care@jolla.com)

For community co-creation projects and community support, refer to

- <http://together.jolla.com>

In this chapter we describe the functionality of The Other Half hardware interface and mechanical interface and dimensions in Jolla devices.

### 2.1 — MECHANICAL INTERFACES AND KEEP OUT AREAS

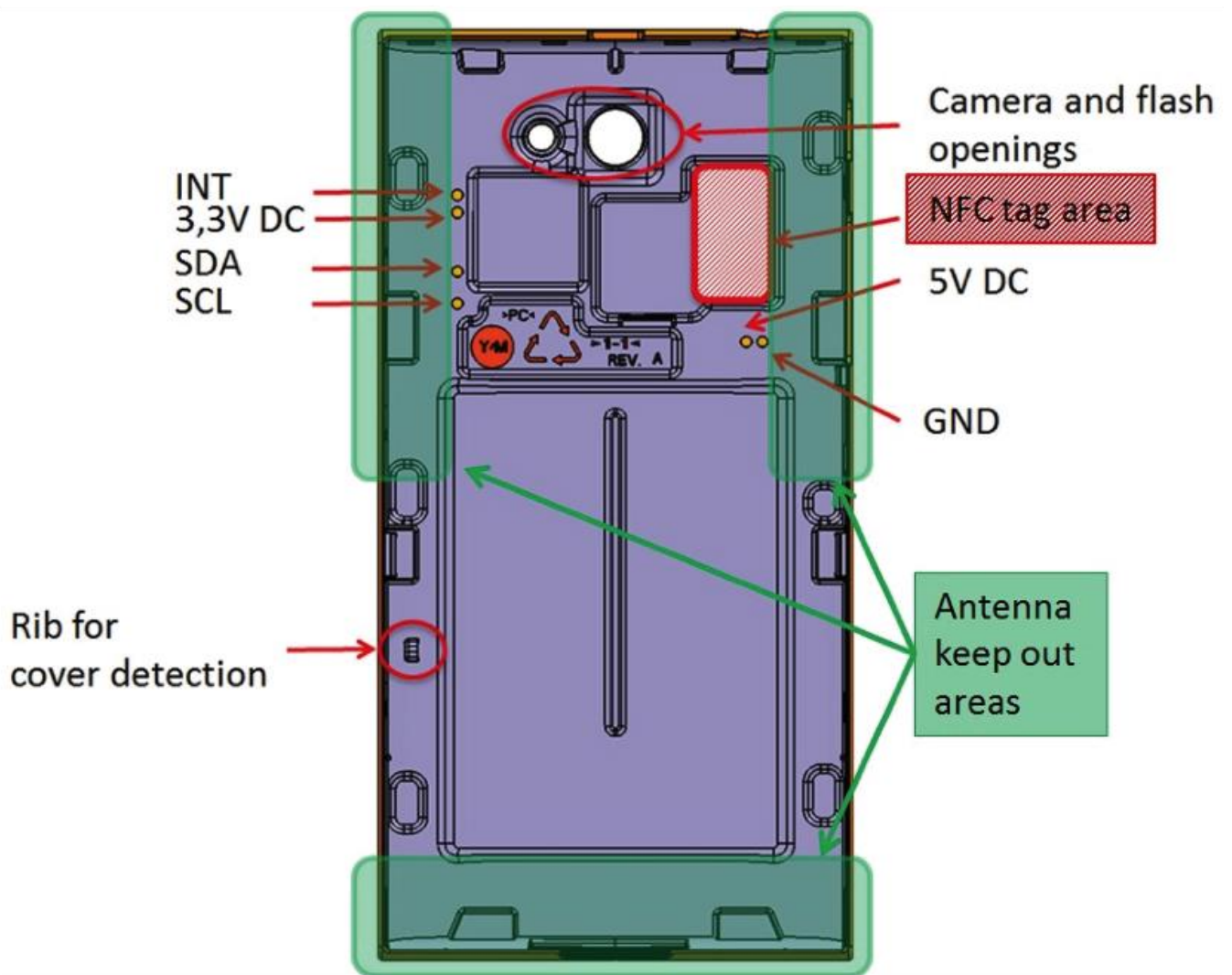
The mechanical 3D-files (attached) and main details are described in the pictures below. In mechanical design it is important to keep the cover attachment hook as designed to maintain the cover attach/removal feeling close to original Jolla The Other Half covers. Also all “keep out” areas and recesses must be kept as in original mechanical file; outer surface and thickness of The Other Half may change.

Pogo pins are used for electrical connections, working range and area of connection pads will be provided (once all data received).

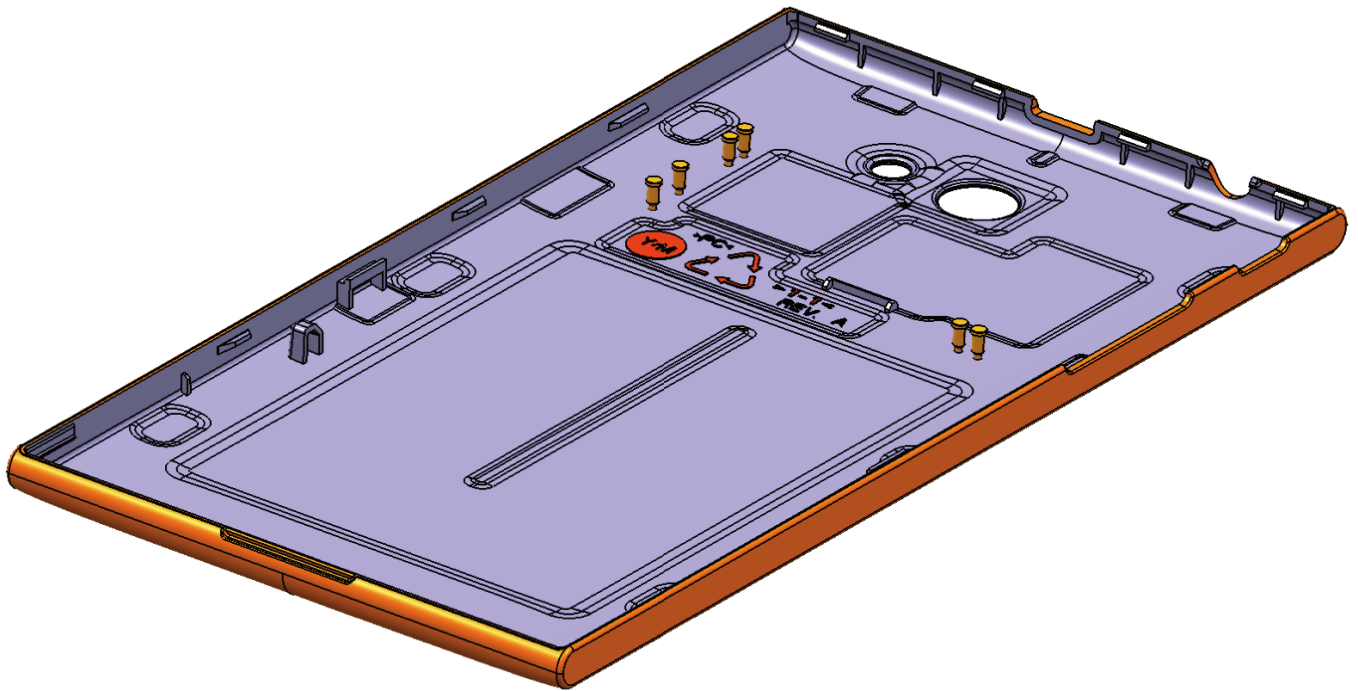
Covering mechanical issues:

- Cover detection mechanical “rib” (pushes micro switch on the Jolla device side)
- NFC tag area (preferred NFC tag type / manufacturer will be provided)
- Contact pins and functions gross ref (locations and working range)
- Camera keep out cone (details in 3D mechanical file)
- Flash keep out area (details in 3D mechanical file)
- Cover hooking to attach to Jolla device (details in 3D mechanical file)
- Antenna keep out areas (details in 3D mechanical file) avoid anything else than pure cover plastic on those areas to keep antenna performance intact.

Picture 1. Mechanical view inside the The Other Half cover



Picture 2. Cover attaching hooks/ribs and connection pogo-pins shown for reference





## 2.2 — ELECTRICAL INTERFACES

Jolla device has 6 functional contact pins towards The Other Half. The contacts use pogo-pins for conducted connection method. Pins have the following functionality:

- 5V DC input pin (1000mA max)
- VDD - 3,3V DC output pin (300mA max)
- SDA – I2C data (1,8V level)
- SCL – I2C clock, frequency 400kHz (1,8V level)
- INT – Signal for interrupt host system (1,8V level)
- GND – Ground of signal and power

### 2.2.1 — I2C

The I2C bus that the cover is connected to shall be operated in Fast mode

(400KHz) clock speed. 1.8V voltage is used on SCL and SDA pins. I2C can be used for multiple purposes in The Other Half side when communicating with the Jolla device.

All I2C devices in the cover must operate in slave mode.

Several devices may be connected to the I2C bus of the cover connector, as long as their addresses do not conflict.

It is expected that standard I2C addressing conventions are followed for any smart I2C devices in the covers. 7-bit addressing mode is recommended.

### 2.2.2 — COVER POWER (VDD)

VDD 3,3V DC output can be used for example for The Other Half powering, I2C powering, charging a smart device in The Other Half etc. Current limitation from that pin is maximum 300mA but 150mA is recommended max current drawn from this pin to avoid any thermal issues. Also it should be taken into account that this will be drawn from the Jolla device battery and will decrease battery life directly.

The 3,3V DC VDD power is controlled via the linux kernel regulator framework. By default the power is OFF. The kernel driver or userspace software accompanied with the Cover has to enable and disable the VDD as needed.

## 2.2.3 — EXTERNAL CHARGER 5V DC VIN

Wireless or any other external charger can be attached over the 5V / 1000mA input power pin (VIN) and ground (GND) pin. This pin can be used for feeding the current from The Other Half towards the Jolla device. The Other Half shall have all other electronics needed for functionality, pin can be used only to feed current towards the Jolla device's power management chip.

The 5V pin is connected to DC charger circuit of the Jolla device. Management of the charging is done automatically inside the Jolla device. The voltage applied over VIN and GND should be 5V. Lower voltage will cause the charging to slow down. Charging effectively stops if VIN voltage drops below 4,3V.

## 2.2.4 — INT INTERRUPT GPIO

The INT general purpose input/output pin can be used to wake up the Jolla device to read smart I2C peripheral data, or for dedicated button / switch function. The gpio line is edge triggered, and can be configured to trigger both on lowering and rising edge.

The INT line has a pull up to 1,8V. The polarity line is active low.

## 2.2.5 — GND

GND pin for ground connection of signals and powers. This pin is always needed when cover has any conducted electrical functionality.

## 2.3 — COVER IDENTIFICATION INTERFACES

For identifying The Other Half covers, NFC and a mechanical cover switch are used.

### 2.3.1 — NFC

The internal NFC functionality of the Jolla device is dedicated to detecting NFC tags in the covers. The NFC is enabled only for short period of time when a cover attach event is detected, and once during system boot to read the tag on the cover that is attached.

Every The Other Half smart cover must contain a NFC tag with unique identifier (UID). The NFC UID is used to identify the cover and to pair content in Jolla store to The Other Half. The tag shall be attached at the inner surface at the location described in mechanical interface part of the document to ensure the functionality and reliability of this identification method.

For best compatibility, it is recommended to use SMARTRAC Midas NTAG203 12x19 mm tags [1].  
Dumb covers with no smart functionality may omit the NFC tag.

## **2.3.2 — COVER DETECTION SWITCH**

When a cover is attached to the Jolla device, a mechanical switch is used to detect this event. The switch is operated via a mechanical “rib” on The Other Half. Please refer to the mechanical part of this document for the correct location of the rib.

# 3 — SOFTWARE ARCHITECTURE

The Other Half smart cover software architecture is geared towards combining correct content with each cover. The content may be software, media, or ambiances. The key of the concept is that each cover is uniquely identifiable. The Jolla Store back end maps each identifier to specific downloadable content that can operate the cover.

The main software components related to The Other Half operations are linux kernel, The Other Half daemon (tohd), Store client, and Ambience daemon (ambienced). The relations of different components are illustrated in Figure 1.

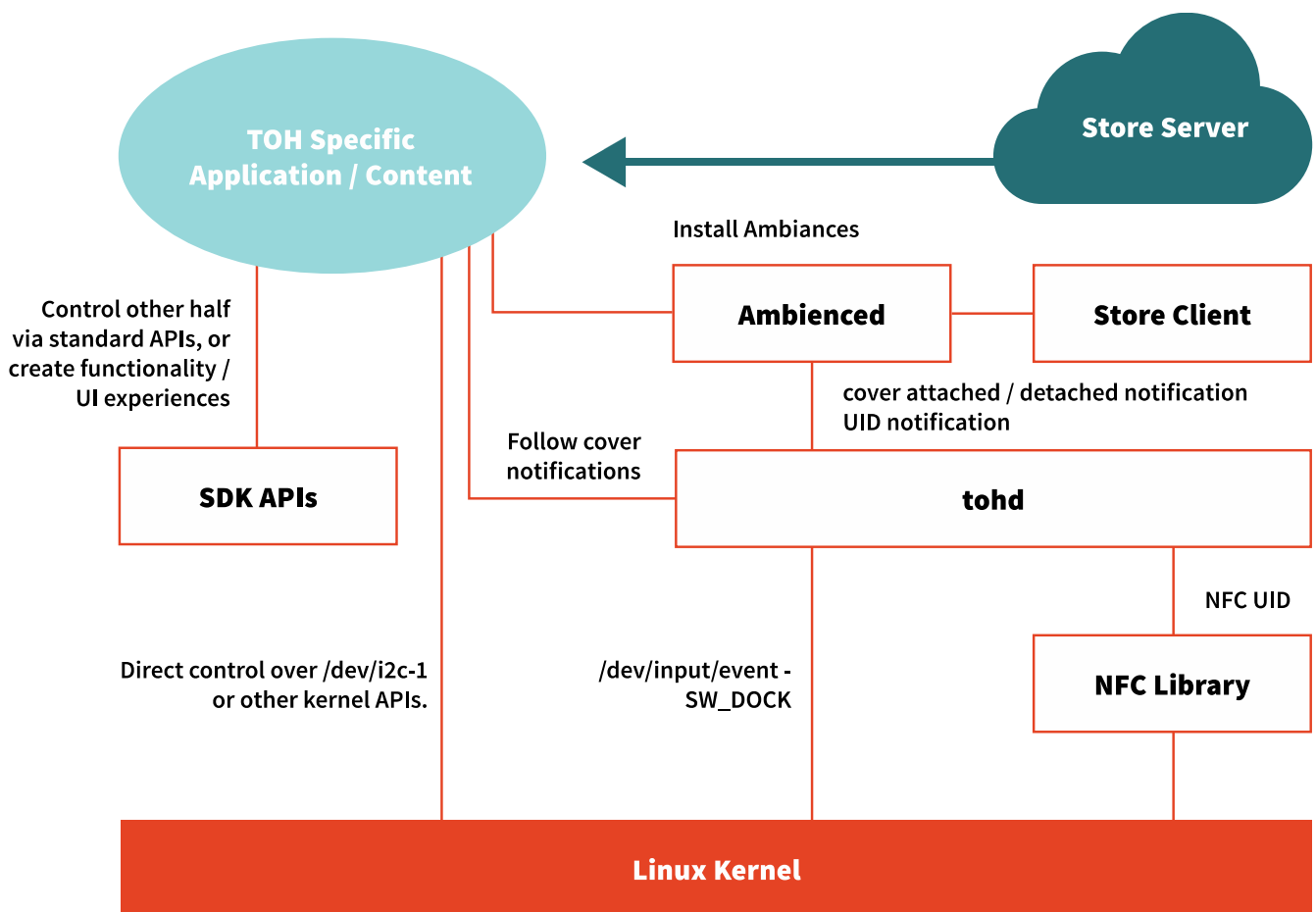


Figure 1: The Other Half software components

## 3.1 — COVER IDENTIFICATION

The identification of The Other Half covers is done via NFC tag UID. Each time the cover is attached to the Jolla device, the NFC tag is read from the UID field of the tag. If a UID is found, it is passed to the store client, which checks the store back-end server for a matching content RPM [2] package. If there is content found in the store server matching the UID, it is downloaded and installed to the Jolla device.

Pulling in multiple packages is possible by adding dependencies from the main content package to other RPM packages.

Please note that any content installed after The Other Half attachment will remain on the Jolla device even if The Other Half is removed or exchanged. Hence it is recommended that any Other Half related active software should follow the UID change notifications, to act only when the right UID is detected.

Installed The Other Half content can be removed manually from the Jolla Store application and reinstalled again by re-attaching TOH.

To follow UID and cover status indications, please refer to tohd D-bus API “com.jolla.tohd”. To view the latest API using introspection, run following command on the Jolla device command line:

```
[root@localhost nemo]# dbus-send --system --type=method_call --print-reply=y --dest=com.jolla.tohd /com/jolla/tohd org.freedesktop.DBus.Introspectable.Introspect
```

That will print out an XML formatted introspect of the API. The properties exported by the API can be read for example via Qdbus and at time of writing they are:

```
[root@localhost nemo]# /usr/lib/qt5/bin/qdbus --system com.jolla.tohd /com/jolla/tohd org.freedesktop.DBus.Properties.GetAll com.jolla.tohd
Docked: 1
Ready: 1
TOHID: 045b5e6a7b2b80
```

The most important properties in the tohd API are

TOHID: This is the NFC UID of the cover that has been attached.

Ready: This signal tells when cover has been fully attached and can be used.

### 3.1.1 — THE OTHER HALF NFC UID ALLOCATION

It is expected that you can provide a list of the NFC UIDs you intend to use in your The Other Half application as a ASCII text file with one UID on each line. To allocate UID ranges in Jolla Store for your The Other Half implementation, please contact us. See contact information in chapter 1.2.

## 3.2 — HARDWARE CONTROL INTERFACES

In this section we describe the software APIs for controlling The Other Half hardware resources.

### 3.2.1 — VDD POWER CONTROL

Active covers which require the 3,3V VDD power, can request the activation of the power via the linux kernel regulator framework user space consumer API :

- `/sys/devices/platform/reg-userspace-consumer.0/state`
- NOTE: This API is not supported in initial Jolla device software. It will be available from 1.0.2.\* and later Sailfish OS releases.

To enable write “1” to the file, to disable write “0” to the file. Below an example how to do that from developer mode command line as root user:

```
# Enable VDD
echo 1 > /sys/devices/platform/reg-userspace-consumer.0/state
# Disable VDD
echo 0 > /sys/devices/platform/reg-userspace-consumer.0/state
```

Kernel level drivers may access the VDD power via normal regulator framework consumer kernel API [3]. The name of the consumer supply for the VDD pin is “**toh\_vdd**”. Example below:

```
#include <linux/regulator/consumer.h>
...
/* Get the struct regulator * pointer for using it in the driver */
mydriver->toh_vdd = regulator_get(mydriver->dev, "toh_vdd");
...
/* Enable TOH VDD */
regulator_enable(mydriver->toh_vdd);
...
/* Disable TOH VDD */
regulator_disable(mydriver->toh_vdd);
...
/* Release the regulator handle on module / driver exit */
regulator_put(mydriver->toh_vdd);
```

The VDD should be turned ON only when it is really needed, and turned OFF otherwise. The NFC UID and cover switch status signals can be tracked to only enable the power when the right cover is attached.

The VDD power will NOT turn OFF automatically when cover is removed. It must be explicitly turned OFF when not needed anymore.

## 3.2.2 I — 2C ACCESS INTERFACES

Kernel space device drivers can use The Other Half I2C connector via normal kernel I2C APIs (<https://www.kernel.org/doc/Documentation/i2c/writing-clients>). For user space control the i2c-dev interface (<https://www.kernel.org/doc/Documentation/i2c/dev-interface>) should be used.

The Other Half is configured to use I2C bus number 1 in the Jolla device. The user space i2c-dev device node can be accessed from:

- /dev/i2c-1

Any kernel space drivers should be probed only if the right cover which has the functionality is attached. This can be achieved by tracking cover NFC UID and Ready signals, and then instantiating the right i2c device to the kernel from i2c sysfs interface [4]:

- /sys/class/i2c-adapter/i2c-1/new\_device

Once the cover is removed, the the Jolla device must be removed as well so that the driver will exit:

- /sys/class/i2c-adapter/i2c-1/delete\_device

Please note that there is no kernel space API for tracking NFC UID changes.

## 3.2.3 — INT LINE HANDLING

The INT line is connected to GPIO 67 in the Jolla device SoC. To use the int line from userspace, you can export it from linux kernel gpio sysfs. To follow the GPIO state changes, you can enable the edge triggering by setting “edge” setting to “falling” and then using for example C function poll() [5] on “value” entry to get notified of the state changes (poll for POLLPRI events). See below a command line example to configure the GPIO:

```
# Export INT gpio 67 to the sysfs
echo 67 > /sys/class/gpio/export
# Configure INT gpio67 to generate events on falling edge.
echo falling > /sys/class/gpio/gpio67/edge
# Check current value
```

```
cat /sys/class/gpio/gpio67/value
```

To use the INT interrupt in kernel space drivers, normal kernel interrupt functions from `<linux/interrupt.h>` can be used.

### 3.3 — SECURITY

It is recommended to use non-sequential NFC tag UIDs, if possible.

### 3.4 — POWER MANAGEMENT

Any applications that interrupt the Jolla device to wake up, should take care to not wake up the host processor when the Jolla device is suspended. During suspend, the power consumption cost of single interrupt is several magnitudes bigger than when the system is up and running.

Kernel space drivers can use normal `.suspend` and `.resume` handlers to handle suspend state. User space implementations can follow display status for similar effect. The display state can be followed from Mode Control Entity's (MCE) D-bus API: <https://github.com/nemomobile/mce-dev/tree/master/include/mce>

Preferably any wake up interrupts should be initiated to respond to user activity. Periodic polling should be avoided when possible.

## 4 — APPLICATION NOTES

This section describes technical requirements for the most common use cases for creating The Other Half applications.



## 4.1 — AMBIANCE COVER

Feature	Requirement	Comments
NFC tag	Must have	The NFC UID needed to download right Ambiance package from store
VDD power	-	
VIN charger pin	-	
I2C	-	
INT input line	-	

The ambiance cover is a implementation of The Other Half, where attaching the cover initiates a download and installation of new ambiance and other content for the Jolla device. The ambiance can customize following parameters in the look and feel of Jolla:

- Wall paper image
- UI colors
- Ring and event tones
- Ring tone volume

The ambiance is specified in a “.ambiance” file format. An example of an ambiance file that lists the customizable elements related to ambiance is shown below:

```
{  
  "translationCatalog"      : "ambiance-template",
```

```

"displayName"      : "ambience-template-name",
"ringerToneFile"  : { "file": "template-ringtonewav", "enabled": 1 },
"messageToneFile" : { "file": "template-message.wav", "enabled": 1 },
"mailToneFile"    : { "file": "template-email.wav", "enabled": 1 },
"internetCallToneFile" : { "file": "template-im.wav", "enabled": 1 },
"chatToneFile"    : { "file": "template-chat.wav", "enabled": 1 },
"calendarToneFile" : { "file": "template-calendar.wav", "enabled": 1 },
"clockAlarmToneFile" : { "file": "template-alarm.wav", "enabled": 1 },
"wallpaper"       : "template.jpg",
"highlightColor"  : "#FF9D80",
"secondaryHighlightColor" : "#FF9D8099",
"primaryColor"    : "#FFFFFF",
"secondaryColor"  : "#FFFFFF99",
"ringerVolume"    : 70,
"favorite"        : true
}

```

To create an ambience, the media files required by the ambience file also need to be packaged into a RPM package (either the main ambience RPM or a content RPM the main RPM depends on).

In addition to the ambience itself, the RPM package may contain any other media content related to The Other Half, like music files, video files, pictures, applications, desktop shortcuts to web URLs.

## 4.2 — EXTERNAL CHARGER COVER

Feature	Requirement	Comments
NFC tag	-	

VDD power	-	
VIN charger pin	Must have	The charging circuit must have 5V VIN and GND pads to connect to corresponding pins on the Jolla device
I2C	-	
INT input line	-	

The external charger Other Half is a cover that can charge the main battery of the Jolla device. The voltage can be obtained for example from a wireless charger circuit, external battery pack, or anything else that can provide 5V.

The only thing the external charger needs to do in terms of functionality towards the Jolla device is to provide 5V over the VIN and GND pins of the Jolla device. The DC charger circuitry inside the Jolla device will draw at maximum 1000mA at 5V. The charging is automatically stopped once the Jolla device's battery is full.

Please note that charging effectively stops if the VIN voltage drops below 4,3V.

It is recommended to test thoroughly the thermal behavior of the cover over a full charging period.

## 4.3 — BLUETOOTH COVER

Feature	Requirement	Comments
NFC tag	Optional	Required software can be packaged and downloaded

		from Jolla Store according to NFC UID.
VDD power	Optional	Can be used to directly power the cover, or to charge the cover if it has its own battery power.
VIN charger pin	-	
I2C	-	
INT input line	-	

Any supported bluetooth profile [6] peripheral may be used in The Other Half. Sailfish OS uses Bluez linux bluetooth stack. At the time of writing the version of Bluez was based on 4.101 version [7]. The latest packaged Bluez in Sailfish OS can be found at <https://github.com/mer-packages/bluez>.

Automatic bluetooth pairing via NFC is not supported.

The VDD pin can be used to directly power the device. Also a dedicated battery may be used in the cover to power it. This battery may be charged via the VDD pin, or using a cover specific external charger.

If VDD power is used to power the cover, it should only be enabled when the right UID, and cover attach event is detected, to avoid leaking power.

## 4.4 — GENERIC I2C DEVICE COVER

Feature	Requirement	Comments
NFC tag	Optional	Can be used to download correct software package

VDD power	Optional	Can be used to directly power the cover, or to charge the cover if it has its own battery power.
VIN charger pin	-	
I2C	-	<p>The cover acts as a I2C slave, and can be accessed via the i2c-dev device at /dev/i2c-1</p> <p>Can be accessed also directly from a kernel space device driver. Default I2C clock rate in kernel is set at 400kHz.</p>
INT input line	-	Can be used to wake up the Jolla device for reading data.

Any I2C peripheral device designed to work in slave mode can be used in The Other Half.

## 5 — REFERENCES

1. **SMARTRAC Midas NFC NTAG203 data sheet.**  
[http://w3.smartrac-group.com/upm/internet/Web\\_RFID\\_ProductSpecification.nsf/ALLBYSALESCODE/3002363/\\$file/tech\\_speck\\_3002363.pdf](http://w3.smartrac-group.com/upm/internet/Web_RFID_ProductSpecification.nsf/ALLBYSALESCODE/3002363/$file/tech_speck_3002363.pdf)
2. **RPM package management.**

<http://www.rpm.org/>

3. **Linux Kernel Regulator Framework Consumer API.**

<https://www.codeaurora.org/cgit/quic/la/kernel/msm/tree/include/linux/regulator/consumer.h?id=b6c36b91b2ed506374d7c526867ec3194fa6c6bf>

4. **Linux Kernel I2C device instantiation.**

<https://www.kernel.org/doc/Documentation/i2c/instantiating-devices>

5. **poll() manual page.**

<http://linux.die.net/man/2/poll>

6. **Jolla supported bluetooth profiles.**

[https://www.bluetooth.org/tpg/QLI\\_viewQDL.cfm?qid=21869](https://www.bluetooth.org/tpg/QLI_viewQDL.cfm?qid=21869)

7. **BlueZ 4.101 documentation.**

<http://git.kernel.org/cgit/bluetooth/bluez.git/tree/doc?id=4.101>