

# **sysmocom**

sysmocom - s.f.m.c. GmbH

## **sysmoQMOD User Manual**

by Harald Welte and Martin Schramm

Copyright © 2016-2020 sysmocom - s.f.m.c. GmbH

All rights reserved.

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME
v1	February 2017	Initial version, covering QMOD hardware v2	hw
v2	May 2017	Update to QMOD hardware v3	hw
v3	Sep 2020	Update to QMOD hardware v3.3	ms

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose	1
1.2	Designated Use	1
1.3	Intended Audience	1
1.4	Regulatory compliance	2
<b>2</b>	<b>Specifications</b>	<b>2</b>
2.1	Electrical	3
2.2	Environmental Specifications	3
<b>3</b>	<b>sysmoQMOD Hardware</b>	<b>3</b>
3.1	Power Supply	6
3.1.1	On-board voltage rails	6
3.1.2	Main 3.3V plane	6
3.1.3	Auxiliary 3.3V plane	7
3.1.4	1.5V plane	7
3.2	Connectors	7
3.2.1	J801: Barrel-type DC Input	8
3.2.2	K501: USB B connector towards USB host (PC)	9
3.2.3	J803: Internal USB upstream port + supply voltage for daisy-chaining	9
3.2.4	X503: External USB A connector for daisy-chaining	10
3.2.5	J802: Internal USB downstream port + supply voltage for daisy-chaining	10
3.2.6	TC601/TC701: JTAG + SWD connector for SAM3 microcontrollers	11
3.2.7	X601/X701: 3.3V Serial Console of SAM3 microcontroller	12
3.2.8	SIM101/SIM201/SIM301/SIM401: SIM Card Slot	13
3.2.9	X101/X201/X301/X401: PCM interface	14
3.2.10	X102/X202/X302/X402: mPCIe connector	15
3.2.11	Installing mPCIe modems	15
3.3	USB Hub	16
3.3.1	USB hub configuration data EEPROM	18
3.3.2	USB peripheral power switching	18
3.4	Modem Monitoring / Control via SAM3	19
3.5	SIM Card Emulation	19
3.6	Hardware Options	20
3.6.1	Enclosure option for sysmoQMOD	20
3.7	LEDs	20
3.7.1	User-visible LEDs	21
3.7.2	Non-User-visible LEDs	21

<b>4</b>	<b>SAM3 Firmware</b>	<b>22</b>
4.1	Updating the firmware via DFU	22
4.1.1	DFU Basics	22
4.1.2	Entering DFU mode from the runtime	22
4.1.3	Updating the application using <code>dfu-util</code>	23
4.2	Emergency recovery using SAM-BA	24
4.2.1	Using <code>bossac</code> for flashing	24
<b>5</b>	<b>Remote SIM functionality</b>	<b>24</b>
5.1	Remote SIM Transport Channel	25
5.2	Remote SIM Use Cases	25
<b>6</b>	<b>Host Software</b>	<b>26</b>
6.1	<code>simtrace2-list</code>	26
6.2	<code>simtrace2-cardem-pcsc</code>	26
6.3	GSMTAP / wireshark APDU tracing	28
6.4	<code>osmo-remsim</code>	28
<b>7</b>	<b>Glossary</b>	<b>28</b>
<b>A</b>	<b>Bibliography / References</b>	<b>36</b>
	References	36

# 1 Introduction

This manual describes the sysmoQMOD a quad mPCIe modem carrier board with integrated management and remote SIM forwarding capabilities.

The target audience are system integrators who use the sysmoQMOD to build their own products, such as remote nodes for cellular network quality monitoring or roaming testing.

## 1.1 Purpose

The purpose of the sysmoQMOD is to operate a set of up to four cellular (WWAN) modems in mPCIe form factors. Such modems are available from a variety of suppliers, such as Sierra Wireless, SIMcom, Quectel, Ericsson, Huawei and others.

Multiple cellular modems connected to a single USB port / USB host can be used by a systems integrator in a variety of applications, such as

- least-cost routing of voice, SMS or data services to different cellular networks of different operators
- load-balancing of data traffic across multiple cellular connections
- remotely deployed systems for automatic remote roaming probes, providing roaming testing services to operators - particularly in combination with the [?] functionality.
- remotely deployed systems for service / QoS testing of cellular networks
- test equipment for interoperability, load and functional testing of cellular infrastructure equipment, such as in the sysmocom LTHW products for testing the Osmocom Cellular Infrastructure stack.

## 1.2 Designated Use

The designated use of the sysmoQMOD is to be used as a carrier board for one to four mPCIe compliant cellular (WWAN) modems, integrated into a larger customer-specific device/appliance.

It is the customer's responsibility to

- ensure proper integration with the cellular modems he selected
- mount the product in a suitable enclosure providing
  - protection from environmental influences
  - protection from ESD (Electrostatic Discharge) exposure to the bare circuit board assembly
  - EMC (Electromagnetic Compatibility) compliance
  - providing sufficient heat dissipation capability for the product and modems
- connect a suitable compatible power supply
- separately obtain and install the modems and SIM cards intended to be used in the given application
- connect cabling, adapters and antennas compatible with the chosen cellular modems

## 1.3 Intended Audience

The intended audience of this manual is the technical staff of the systems integrator who integrates the sysmQMOD into a customer-specific product/appliance within the designated use stated above.

## 1.4 Regulatory compliance

The product is designed to be conforming with all applicable harmonized standards in the EU. As such, the series-produced units will be accompanied with a declaration of conformity for the European market.

However, only a given reference configuration of the product is submitted to related conformance testing. This reference configuration includes

- a set of four CE conforming cellular modems
- a typical shielded metal enclosure (19" 1U style)
- a CE conforming AC power supply for supplying 12V
- a set of U.FL to SMA pigtails
- a set of antennas compatible with the requirements of the modem
- a CE conforming personal computer connected to the USB B port

Due to the many variable parameters of any customer-specific appliance built from the sysmoQMOD, it is the responsibility of said system integrator to test and declare conformity with all applicable norms and standards on his final product.

## 2 Specifications

- 160x120mm four-layer PCB assembly with two-sided component placement
- mounting holes (M3) for mounting on PCB spacers/stands
- 4 miniPCIe slots for cellular / WWAN Modems
- 4 SIM/UICC slots for cards in ETSI/3GPP 2FF form-factor
- integrated 7-port USB 2.0 High Speed hub (6 ports used internally)
- 1 USB device port on USB-B connector or 2.54mm pin header (factory option)
- 1 USB host port on USB-A connector or 2.54mm pin header (factory option)
- 2 Atmel SAM3 microcontrollers for monitoring and control
  - temperature monitoring
  - individual reset of modems
  - power cycling (all 4 modems together)
  - SIM card forwarding
  - remote WWAN LED reading of each modem
- on-board WWAN LED for each modem
- on-board programmable LED for each modem (typical use: Remote SIM activity)

Table 1: (continued)

## 2.1 Electrical

Table 1: Electrical Specifications

Parameter	Min	Typ	Max	Unit
Supply Voltage	6	12	14	Vdc
Supply Current			5	Adc

The current rating is determined almost exclusively by the type of modems used and above figure is the worst-case peak consumption with all four modems currently transmitting a GSM burst at 2W RF output power (each). The board itself doesn't draw a significant amount of power, typically 200-250mA only.

## 2.2 Environmental Specifications

The sysmoQMOD has been designed exclusively from parts specified for the **full industrial temperature range from -40 to 85 centigrade**.

### NOTE

The actual usable temperature range might be limited by system integration. Care must be taken to ensure sufficient heat dissipation.

Please also observe the environmental specification of the mPCIe modems you intend to use. They might have a more limited range than the sysmoQMOD board itself.

## 3 sysmoQMOD Hardware

The sysmoQMOD hardware consists of a two-sided quad-layer printed circuit board assembly (PCBA), sized 120x160mm.

From v3 onwards, the board features edge-launch sim slots as well as provisions for light guides. This means that SIM cards can be accessed from outside of an enclosure via a front panel.

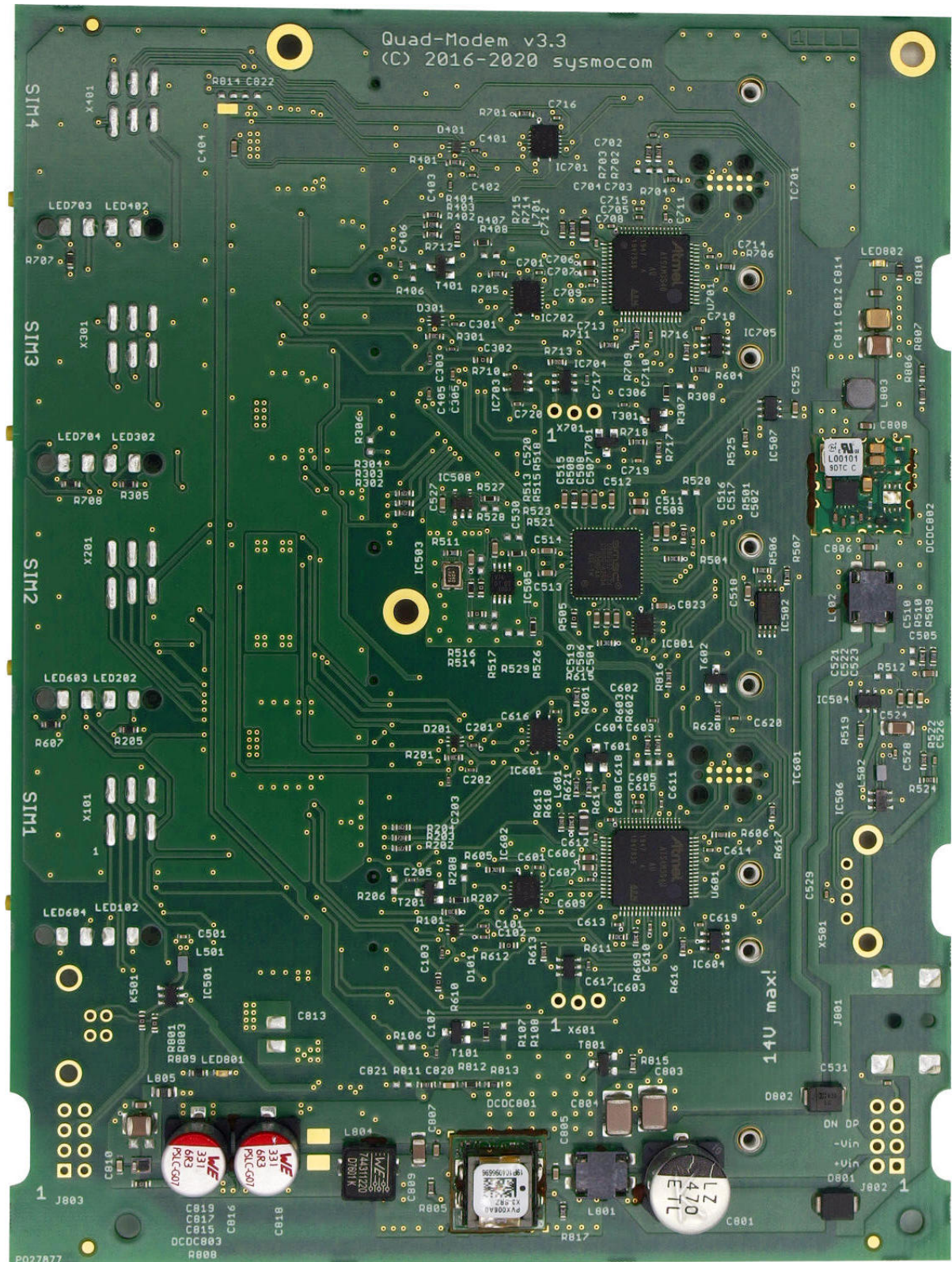


Figure 1: sysmoQMOD PCBA top side



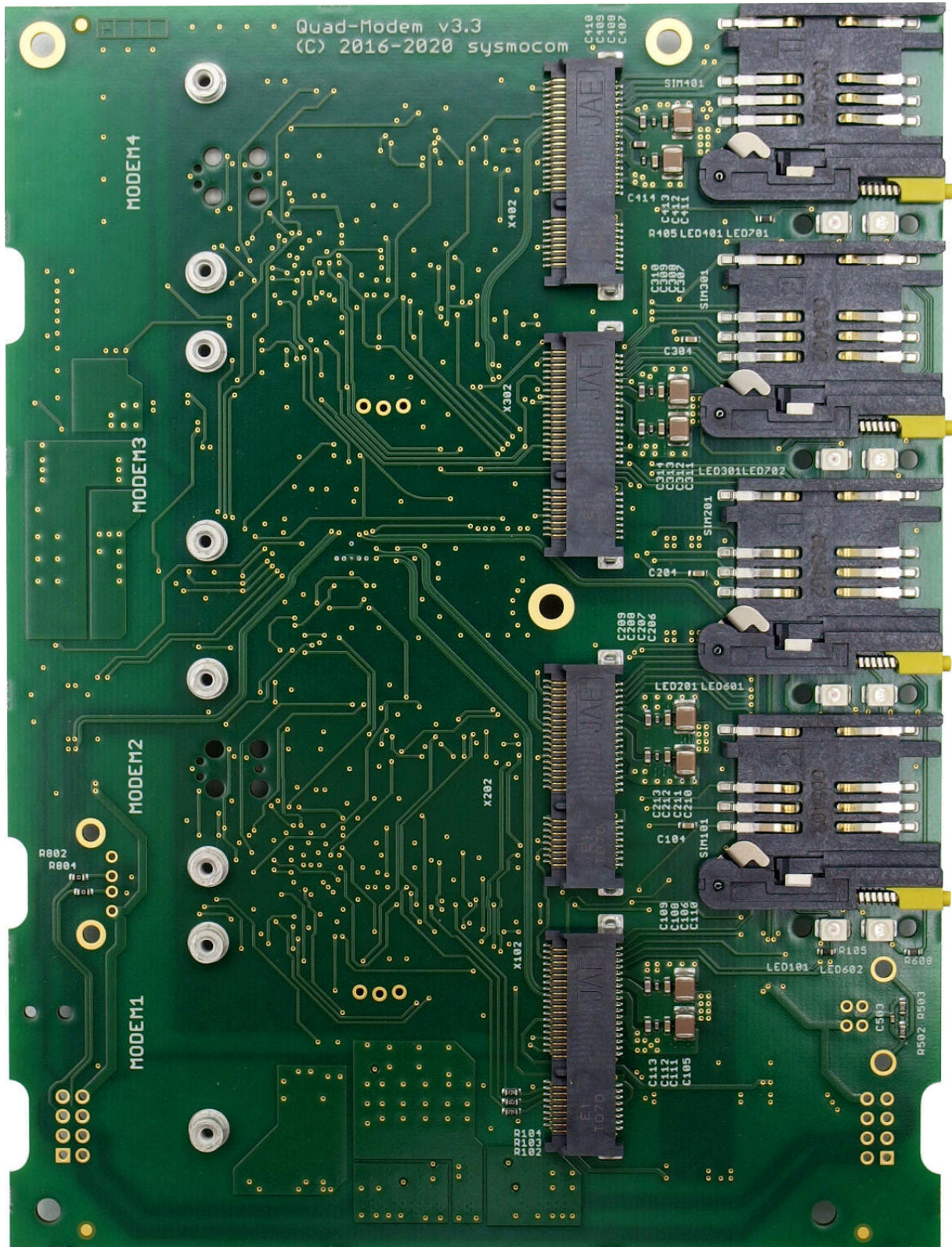


Figure 2: sysmoQMOD PCBA bottom side

**NOTE**

Pinheader, USB sockets and DC input are subject of customization options; shown here is a PCBA before populated connector options (see following sections).

### 3.1 Power Supply

The sysmoQMOD has one nominal 12V DC power input which is used to power all on-board circuits. This 12V DC input can be supplied via *either* one of the following options:

- The barrel-type power jack **J801** (center positive)
- Pins 1,2 (negative) and 8,7 (positive) of **J802**
- Pins 1,2 (negative) and 8,7 (positive) of **J803**

The power input has on-board protection diodes to prevent polarity reversal (D801) as well as an over-voltage zener (D802) diode. However, there **is no internal on-board fuse**, so the power supply should have reasonable current limits in place, or an external fuse shall be used.

#### 3.1.1 On-board voltage rails

On-board step-down DC/DC converters provide several different internal voltage rails from this

- **Main 3.3V plane** for the four mPCIe modems
- **1.5V plane** for the four mPCIe modems
- **Auxiliary 3.3V plane** (for USB hub and SAM3 microcontrollers)

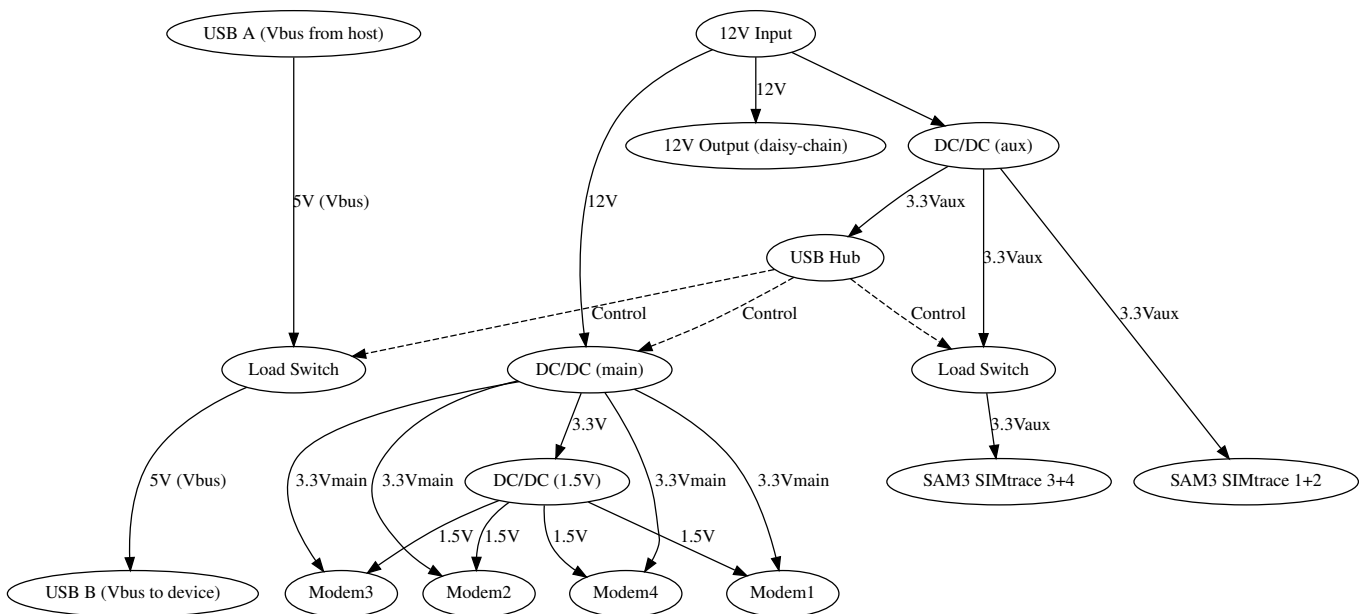


Figure 3: Power Diagram of the sysmoQMOD board

#### 3.1.2 Main 3.3V plane

This power plane is provided to the four mPCIe modems. The power supply can be enabled/disabled via the on-board USB hub, in order to power-cycle the modems.

The total maximum current available on this plane is 6A.

### 3.1.3 Auxiliary 3.3V plane

This power plane is provided to the on-board USB hub, SAM3 micro-controllers, clock-generator and related digital and analog circuits. It is permanently enabled.

The total maximum current available on this plane is 1A.

### 3.1.4 1.5V plane

The 1.5V plane is generated by a DC/DC converter from the Main 3.3V plane. It is made available to all four modems, to ensure mPCIe spec compliance which requires a 1.5V supply in addition to the 3.3V supply.

The total maximum current available on this plane is 600mA.

## 3.2 Connectors

The QMOD has the following physical connections:

- General
  - J801: Barrel-type DC input
  - K501: USB B connector towards USB host (PC)
  - X501: USB A connector for daisy-chaining
  - TC601: JTAG + SWD connector for primary SAM3 microcontroller
  - X601: 3.3V Serial Console of primary SAM3 microcontroller
  - TC701: JTAG + SWD connector for secondary SAM3 microcontroller
  - X701: 3.3V Serial Console of secondary SAM3 microcontroller
  - J803: 12V DC IN/OUT and USB device port towards USB host
    - \* The USB traces are either routed to K501 or J803
  - J802: 12V DC IN/OUT and USB host port for daisy-chaining
    - \* The USB traces are either routed to X501 or J802
- Modem 1
  - SIM101: SIM card slot
  - X101: PCM interface (only available on select modems)
  - X102: mPCIe connector for modem
- Modem 2
  - SIM201: SIM card slot
  - X201: PCM interface (only available on select modems)
  - X202: mPCIe connector for modem
- Modem 3
  - SIM301: SIM card slot
  - X301: PCM interface (only available on select modems)
  - X302: mPCIe connector for modem
- Modem 4
  - SIM401: SIM card slot

- X401: PCM interface (only available on select modems)
- X402: mPCIe connector for modem

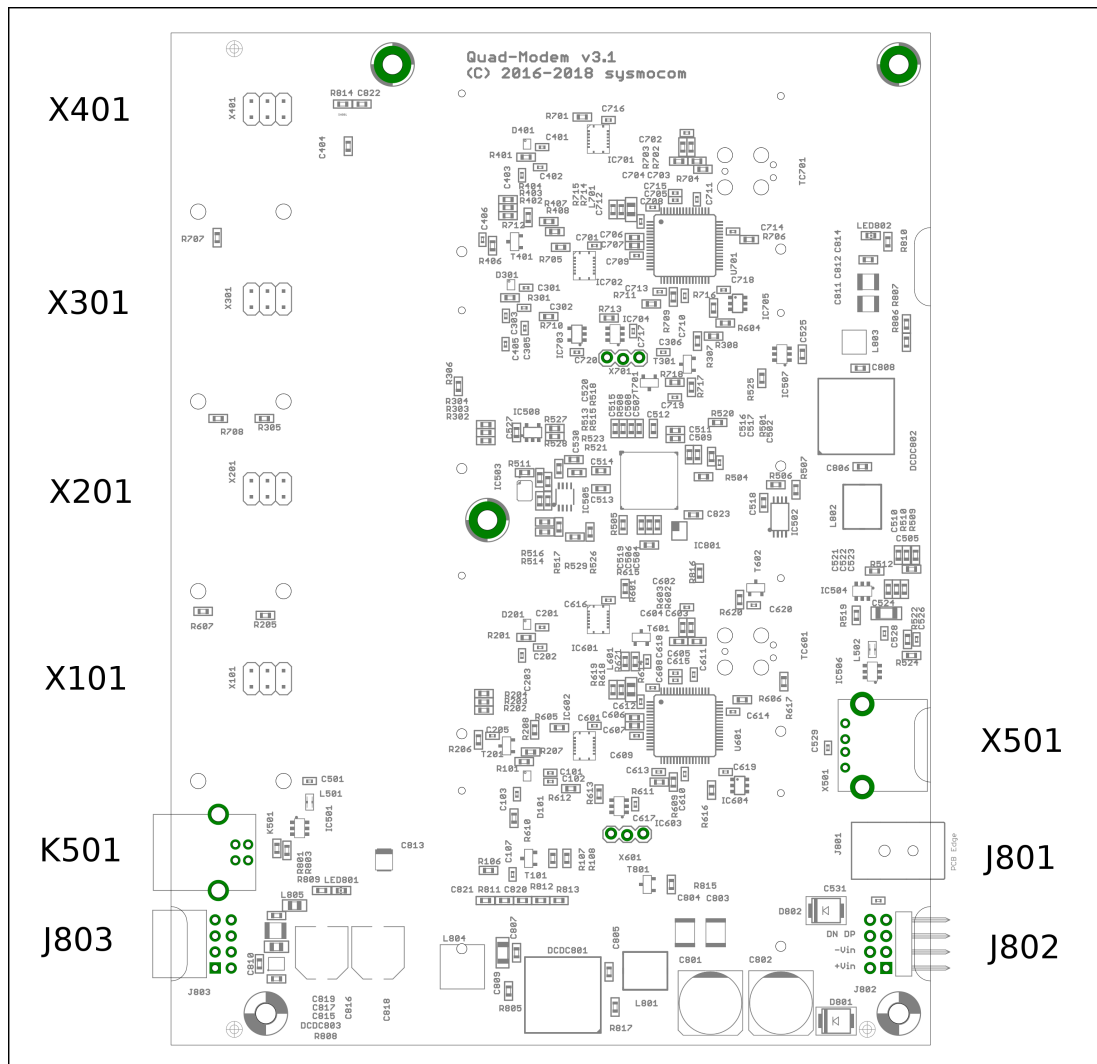


Figure 4: connector names and positions on sysmoQMOD board, top view

### 3.2.1 J801: Barrel-type DC Input

This is a 5.5mm diameter barrel-type DC input connector with 2.5mm pin diameter and the following pin-out:

Table 2: K501 USB B Pinout

Pin	Function	Type
Center	+VIN	12V DC Input (positive)
Outer	-VIN	12V DC Input (ground, <b>not chassis ground</b> )



### 3.2.2 K501: USB B connector towards USB host (PC)

This is a standard USB-B type connector, exposing the USB device port of the on-board USB hub towards an external USB host (PC).

Table 3: K501 USB B Pinout

Pin	Function	Type
1	VBUS	Input
2	D-	Bi-Directional
3	D+	Bi-Directional
4	GND	Reference

VBUS is only used for detecting the presence of a USB host, it is not used to power the sysmoQMOD board. However, it is passed-through to X501, please see Section 3.2.4.

#### NOTE

By assembly option, the USB port can be exposed on **either** K501 (USB B) or J803 (2.54mm header), but not both!

### 3.2.3 J803: Internal USB upstream port + supply voltage for daisy-chaining

This is a dual-row 8pos 2.54mm connector that can be used to interconnect two sysmoQMOD boards in daisy-chaining fashion using either direct board-to-board connectors or an interconnection cable.

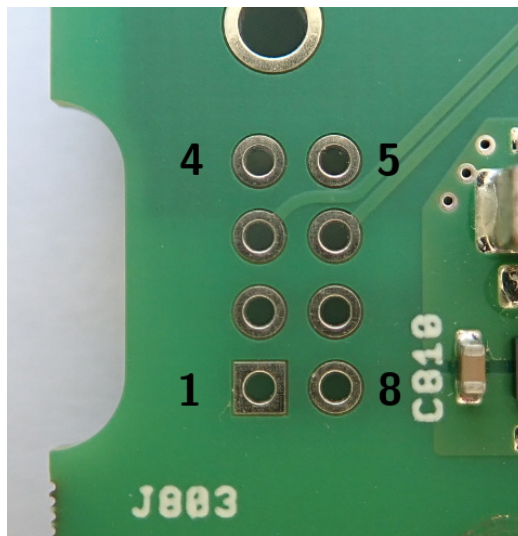


Figure 5: pin assignment J803

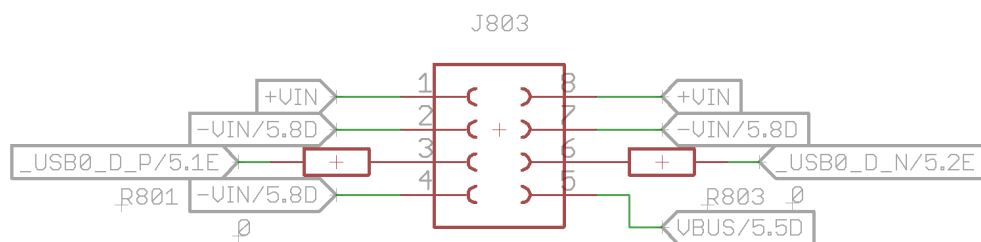


Table 4: J803 Pinout

Pin	Function	Type
1	+VIN	12V Supply Power (in or out)
2	-VIN	GND of Supply Power (in or out)
3	D+	Positive USB differential signal
4	-VIN	GND of Supply Power (in or out)
5	V_BUS	+5V USB Bus voltage output towards client
6	D-	Negative USB differential signal
7	-VIN	GND of Supply Power (in or out)
8	+VIN	12V Supply Power (in or out)

If you are using a cable attached to J803, you must make sure to use an impedance-matched symmetric pair of twisted pairs with an impedance of 90 Ohms, just like all USB cabling. A simple ribbon cable or individual wires will **not** work and cause signal integrity issues on USB.

**NOTE**

By assembly option, the USB port can be exposed on **either** K501 (USB A) or J003 (2.54mm header), but not both!

**3.2.4 X503: External USB A connector for daisy-chaining**

This is a standard USB-A type connector, exposing one of the onboard USB hub downstream ports to an external device. Its primary function is intended for daisy-chaining multiple sysmoQMOD boards without the need for an external hub.

Table 5: X501 USB A Pinout

Pin	Function	Type
1	VBUS	Output
2	D-	Bi-Directional
3	D+	Bi-Directional
4	GND	Reference

**NOTE**

Despite sysmoQMOD being a self-powered USB peripheral, X503 is powered by the USB host, and **not** by sysmoQMOD itself. Thus, be very careful not to overload the USB current budget of the USB host!

**NOTE**

By assembly option, the USB port can be exposed on **either** X501 (USB A) or J802 (2.54mm header), but not both!

**3.2.5 J802: Internal USB downstream port + supply voltage for daisy-chaining**

This is a dual-row 8pos 2.54mm connector that can be used to interconnect two sysmoQMOD boards in daisy-chaining fashion using either direct board-to-board connectors or an interconnection cable.

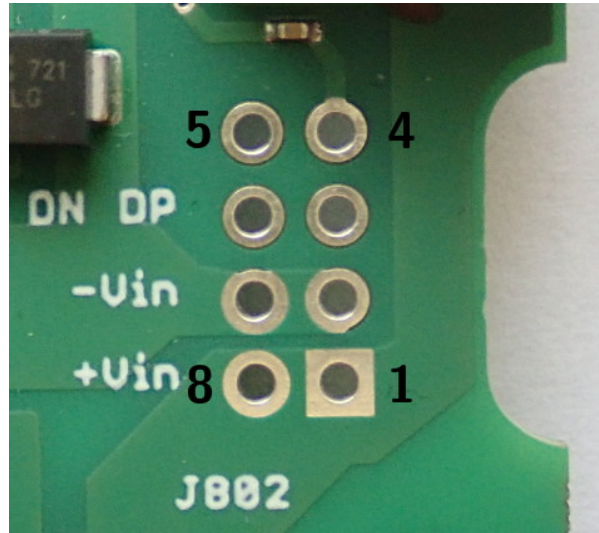


Figure 6: pin assignment J802

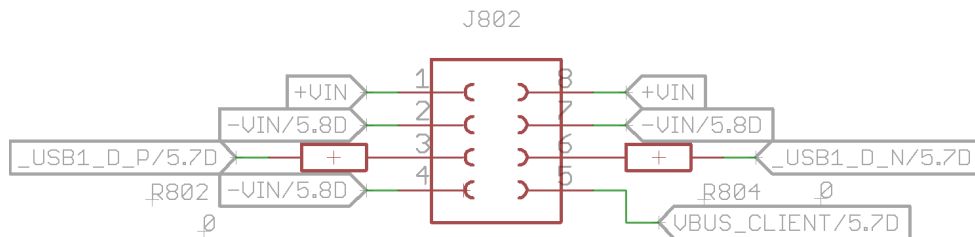


Table 6: J802 Pinout

Pin	Function	Type
1	+VIN	12V Supply Power (in or out)
2	-VIN	GND of Supply Power (in or out)
3	D+	Positive USB differential signal
4	-VIN	GND of Supply Power (in or out)
5	V_BUS	+5V USB Bus voltage output towards client
6	D-	Negative USB differential signal
7	-VIN	GND of Supply Power (in or out)
8	+VIN	12V Supply Power (in or out)

If you are using a cable attached to J802, you must make sure to use an impedance-matched symmetric pair of twisted pairs with an impedance of 90 Ohms, just like all USB cabling. A simple ribbon cable or individual wires will **not** work and cause signal integrity issues on USB.

#### NOTE

By assembly option, the USB port can be exposed on **either** X501 (USB A) or J802 (2.54mm header), but not both!

### 3.2.6 TC601/TC701: JTAG + SWD connector for SAM3 microcontrollers

This is a 10-pin TagConnect connector, compatible with the TC2050 plug. Using this connector has the advantage of having zero cost impact on the PCB side, as the "connector" is just a PCB footprint.

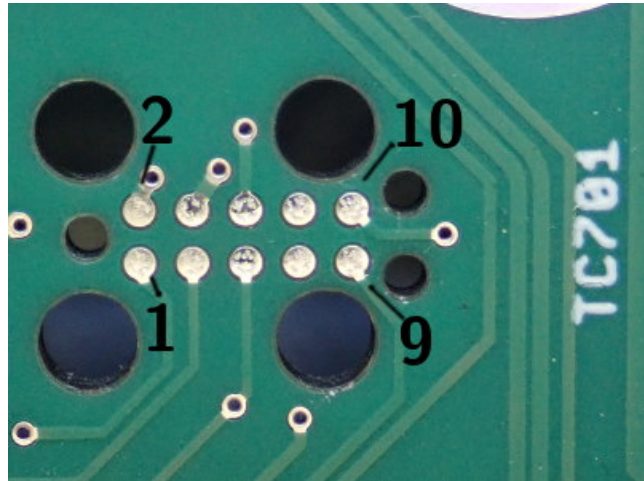


Figure 7: pin assignment TC601 and TC701 (shown TC701)

The pin-out is as follows:

Table 7: TC601/TC701 Pinout

Pin	Function	Type
1	TCK	Input
2	GND	Reference
3	TDO	Output
4	VCC	Reference
5	TMS	Input
6	!RESET	Input
7	NC	
8	NC	
9	TDI	Input
10	GND	Reference

JTAG can be used for flash programming, factory testing and firmware development/debugging of the SAM3 microcontroller.

### 3.2.7 X601/X701: 3.3V Serial Console of SAM3 microcontroller

This connector exposes the serial debug UART of the respective SAM3 microcontroller. During development and testing, this can be used for debugging and firmware updates.

Table 8: TC601/TC701 Pinout

Pin	Function	Type
1	RXD	Input
2	TXD	Output
3	GND	Reference



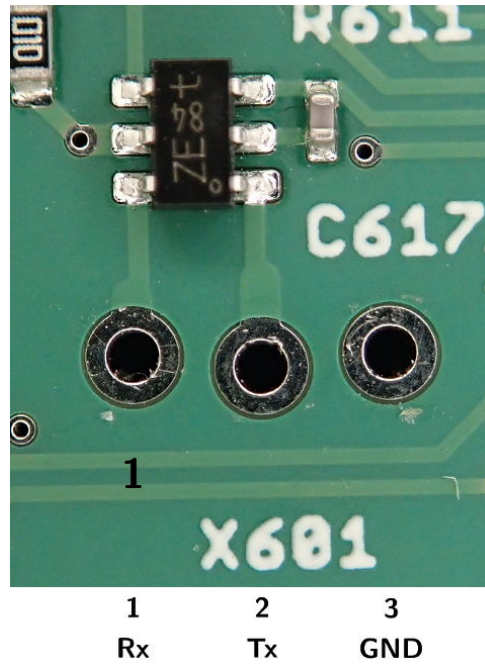


Figure 8: pin assignment X601

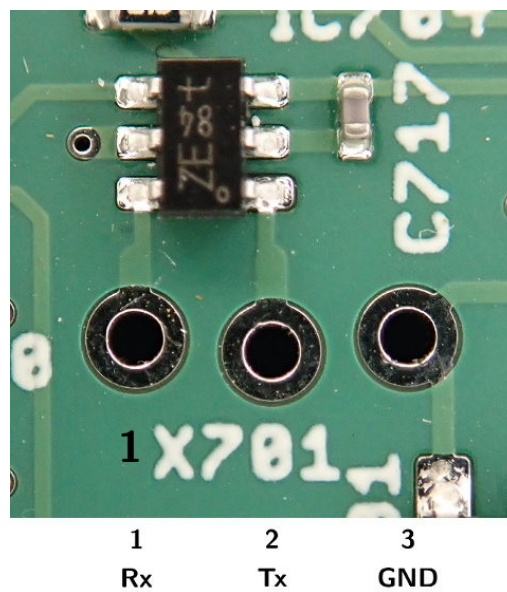


Figure 9: pin assignment X701

An adapter cable to the *Osmocom standard* 2.5mm stereo jack for 3.3V UART is available from sysmocom. The adapter cable needs to have its **black** cable on the **GND** pin (3) of the header, i.e. the black cable facing to the side of the USB-A (host) port.

### 3.2.8 SIM101/SIM201/SIM301/SIM401: SIM Card Slot

This is a standard SIM card holder for a SIM card in ETSI 2FF form-factor.

For the pin-out, please refer to the ETSI/3GPP specifications on SIM/UICC cards.

The card slot type is a Molex 91228-3001.



Figure 10: Molex 91228-3001 SIM Slot

It requires that SIM cards are inserted into the Molex Card Holder 91236-0001.



Figure 11: Molex 91236-0001 SIM Holder

### 3.2.9 X101/X201/X301/X401: PCM interface

Some select mPCIe modules make available the digital audio interface (PCM bus) on non-standard pins of the mPCIe socket. This connector exposes pins 45, 47, 49 and 51 of the respective mPCIe socket. It is intended to be attached to some external peripheral capable of interfacing the PCM bus.

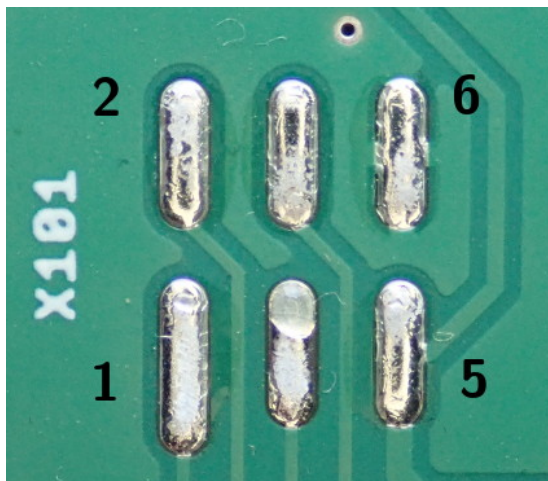


Figure 12: pin assignment X101/X201/X301/X401 (shown X101)

Table 9: X101/X201/X301/X401: PCM interface pinout

Pin	Function
1	mPCIe Pin 45
2	mPCIe Pin 47
3	mPCIe Pin 49
4	mPCIe Pin 51
5	VCC
6	GND

### 3.2.10 X102/X202/X302/X402: mPCIe connector

This is a mPCIe connector suitable for interfacing cellular modems (sometimes called WWAN modems) in mPCIe form-factor. It provides supply power on the 3.3V and 1.5V rails, as well as the electrical USB interface to the on-board hub and the SIM card slots and SAM3 for SIM emulation.

For the pin-out, please refer to the relevant mPCIe specification.

#### Note

For different reasons, not all modem manufacturer decided to comply electrically to PCI-SIG standardized "PCIexpress mini card" specification. One affected modem seen so far is the ZTE ZM8620 V2/A; it is not advised to run this on an unmodified QuadModem\_v3 PCBA. System integrators shall run own testing for usability of a particular modem prior to purchase.

### 3.2.11 Installing mPCIe modems

Beginning with version 3.3, the miniPCIe card modem are mounted with M2 screws in threaded SMT standoffs, before (3.2 and earlier) brackets with fast mount clips were used. Eight large-head M2 screws (PH1, 3mm) are included in delivery per QuadModem.

#### Note

During testing, the proper soldering of all SMT standoffs is verified. However, we can not be held responsible for broken standoffs because of too much torque. When tightening the screws, do not apply more than 0.5 Nm torque. Use a PH1 screwdriver, not a wrench - except it is a precision torque wrench displaying the force. We strongly suggest rather using less torque and an additional usage of a threadlocker, e.g. Loctite 222 or 243.

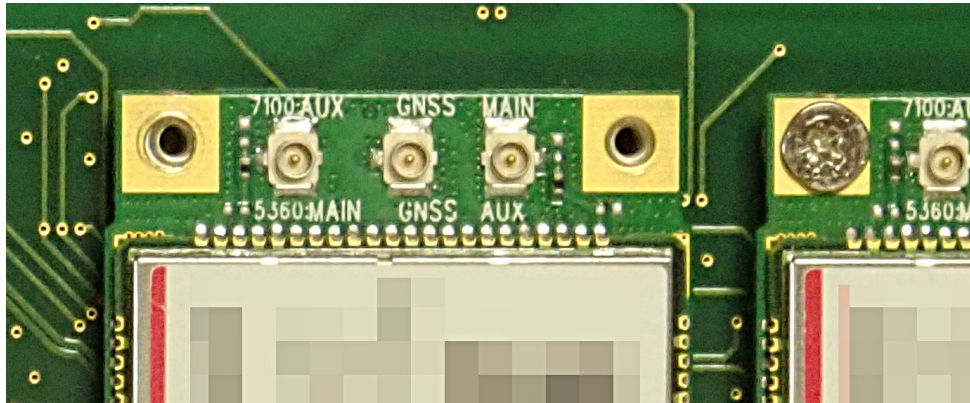


Figure 13: proper alignment before placing screws

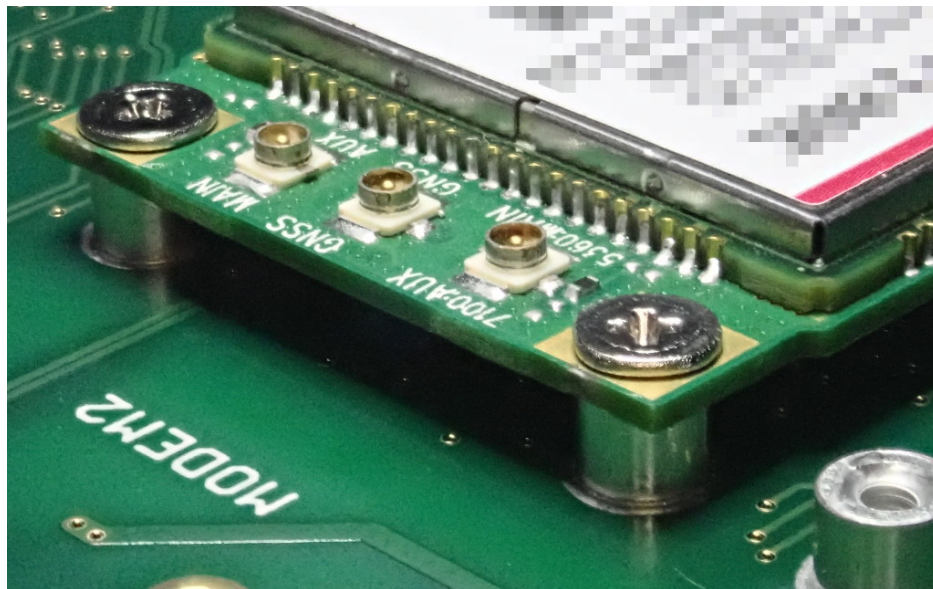


Figure 14: modem screws in place

### 3.3 USB Hub

The sysmoQMOD features an on-board 7-port USB hub with industrial qualification. This 7-port hub interconnects the various components on the board with an external USB host (e.g. PC, embedded SoM, ...)

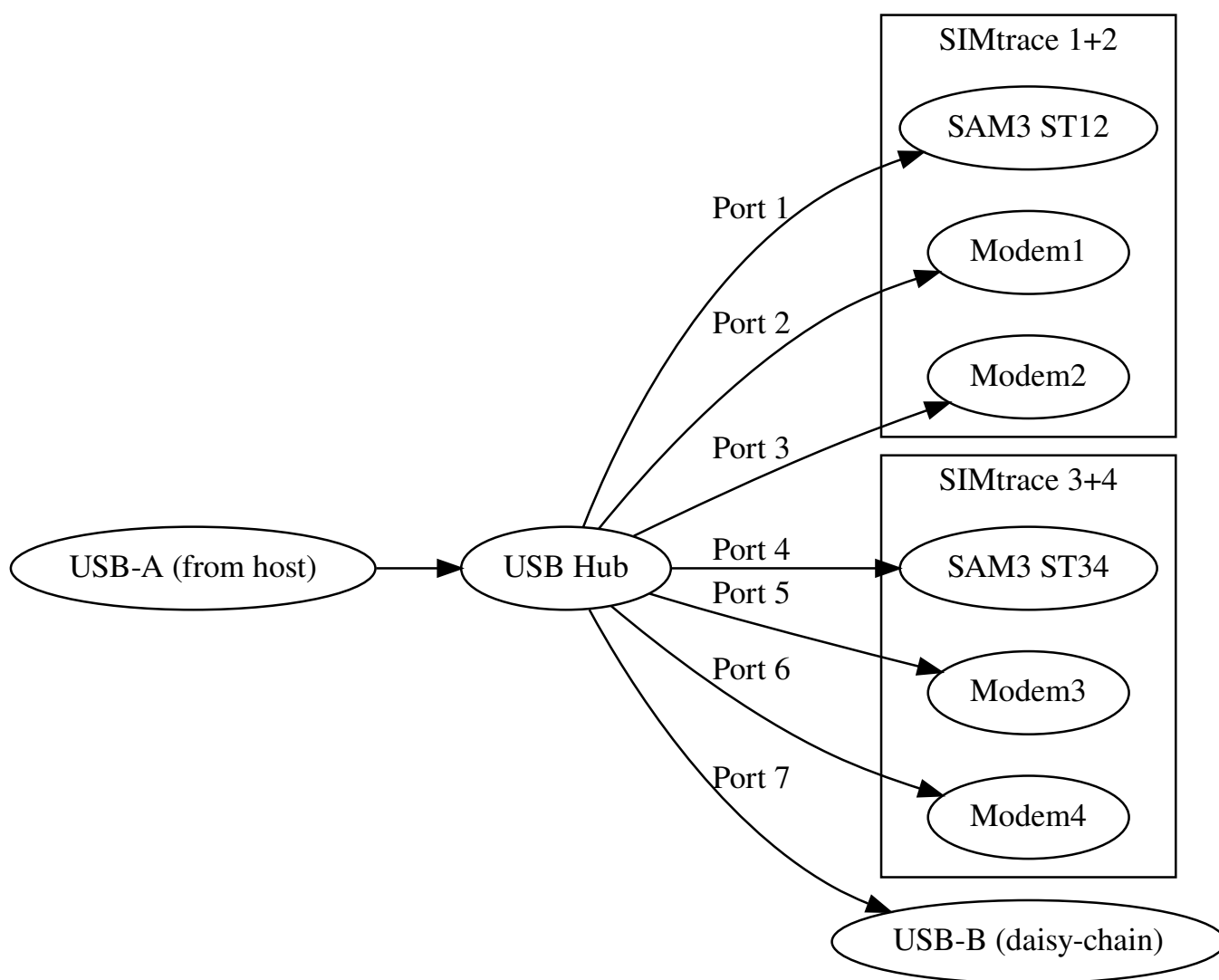


Figure 15: USB Block Diagram of the sysmoQMOD board

Table 10: X101/X201/X301/X401: USB Port Number Allocations

Physical Port	Logical Port	Connects to
UP	UP	USB Host via K501 or J803
DN5	1	SAM3 for Modem 1+2
DN4	2	Modem 1
DN3	3	Modem 2
DN7	4	SAM3 for Modem 3+4
DN2	5	Modem 3
DN1	6	Modem 4
DN6	7	USB device via X503 or J802

On Linux, tools like **lsusb -t** can be used to show the logical tree of USB devices and the status of the ports.

Tools like **uhubctl** can be used to power-up/power-down individual ports at runtime. This can for example be used to disconnect and switchi-off the modems.

### 3.3.1 USB hub configuration data EEPROM

The USB hub configuration is stored in an EEPROM (IC502).

All parameters including USB Product/Vendor IDs are fully configurable by programming. If you require any specific factory programming, please contact sysmocom.

In-the-field reprogramming of the EEPROM is possible, as the EEPROM SDA/SCL signals are routed to the primary SAM3 (U601).

---

**Note**

On common desktop Linux distributions, the package `ModemManager` installs by default, it then handles all installed modems. `ModemManager` up to now contains a blacklist entry for all devices presenting an OpenMoko USB VID 0x1d50 (default for QuadModem). This once was implemented intentionally, but unfortunately also excludes USB hubs, subsequently leading into `ModemManager`-unmanaged modems in QuadModem (entry can be found in file `77-mm-usb-device-blacklist.rules`). - It depends on customer's use case, whether this behaviour is wanted or not; however, this particular blacklist entry can be considered a bug in current `ModemManager`. Sysmocom brought an accepted patch upstream to fix this (see <https://cgit.freedesktop.org/ModemManager/ModemManager/commit/?id=62334b60a24c06a9e699c7b1dde56f52c30729f8>), but usually it may take a while until the package gets published in an updated version by the maintainer, and the respective desktop Linux distribution updates this package too.

---

### 3.3.2 USB peripheral power switching

In terms of power switching capabilities, the external circuitry is wired as follows:

- if any of the Modem-facing USB ports are enabled, supply power is provided to all modems. Thus, in order to power-cycle the modems, you must instruct the USB hub to disable power on all modem ports!
- the primary SAM3 for Modem1+Modem2 is powered at all times and intentionally cannot be switched off
- the secondary SAM3 for Modem3+Modem4 can be powered off by switching off the related USB port.
- the downstream USB port (X503/J802) can be powered off by switching off the related USB port.

Please see <https://github.com/mvp/uhubctl> for a Linux command line program that can be used to control the power of USB ports.

**example: show status (QMOD on host, all ports powered, no modem; port 1 is ST12, port 4 is ST34)**

```
$ uhubctl
Current status for hub 1-1.1, vendor 1d50:4002, 7 ports
  Port 1: 0107 power suspend enable connect
  Port 2: 0100 power
  Port 3: 0100 power
  Port 4: 0107 power suspend enable connect
  Port 5: 0100 power
  Port 6: 0100 power
  Port 7: 0100 power
$
```

**example: power-cycle ST34**

```
$ uhubctl -p4 -a2
Current status for hub 1-1.1, vendor 1d50:4002, 7 ports
  Port 4: 0107 power suspend enable connect
Sent power off request
New status for hub 1-1.1, vendor 1d50:4002, 7 ports
  Port 4: 0000 off
Sent power on request
New status for hub 1-1.1, vendor 1d50:4002, 7 ports
  Port 4: 0100 power
$
```



**example: power-cycle the four modem:**

```
$ uhubctl -p2356 -a2
...
```

**Note**

The default cycle action by `uhubctl` is 2 seconds, which might be sometimes too short for all modem to reset properly. In such cases increase the delay with the `-d` option - see `uhubctl -h`.

**3.4 Modem Monitoring / Control via SAM3**

The on-board SAM3 microcontrollers have some capability to monitor and control the on-board modems. This ensures the option of a full remote recovery from any kind of stuck states, which is important in long-term remote deployments without the need for "remote hands" to power cycle equipment.

Specifically, the following features are implemented:

Table 11: Modem Monitoring/Control Capabilities

Signal Name	Description
PERST[1-4]	Assert/De-Assert the !PERST signal towards the mPCIe modem. This can be used to individually reset any of the mPCIe modems under software control of the SAM3.
WWAN[1-4]	Read the Modem LED status as per the WWAN_LED of the mPCIe modem. This can be used to remotely replicate the LED status and blinking pattern by software in the SAM3.
_HUB_RESET	Reset the USB hub chip. This can e.g. be used by some watchdog on a protocol level between the USB host and the sysmoQMOD. Only available on primary SAM3 (U601).

**3.5 SIM Card Emulation**

The SIM Card emulation feature is implemented by two main components:

- a SPDT switch between each mPCIe modem and its SIM slot, allowing the SIM card related signals to be routed to a SAM3 microcontroller.
- a SAM3 microcontroller with specific firmware to emulate the card-side ISO7816-1/-2/-3 interface

The following signals exist on the SAM3 for each of the four modems:

Table 12: SIM Card Emulation Signals

Signal Name	Description
!CONNECT_ST_USIM[1-4]	Switch between remote/emulated SIM (low) and local/physical SIM (high)
!ST_USIM[1-4]_RST	Simulated SIM Reset input (driven by mPCIe Modem)
ST_USIM[1-4]_IO	Simulated SIM I/O
ST_USIM[1-4]_CLK	Simulated SIM clock input (driven by mPCIe Modem)
MDM[1-4]_USIM_VDD	Simulated SIM supply power (driven by mPCIe Modem)

**Note**

The remote SIM functionality depends on a proper implementation of the `PERST#` signal by the modem manufacturer. Many modems have this, but there is also a number of known modems which don't. Systems integrators should run own tests on specific modems to ensure usability for remote SIM function in terms of proper `PERST#` signal implementation for asserting a warm reset on the modem card.

## 3.6 Hardware Options

The hardware can be built with some options, among them are

- USB connector type can be either
  - regular USB-A and USB-B connectors, or
  - horizontal daisy-chaining connectors for adjacent board chaining (one female, one male), or
  - horizontal daisy-chaining connectors for ribbon cable interconnect between boards (both male headers), or
  - vertical daisy-chaining connectors for ribbon cable interconnect between boards (both male headers)
- PCM headers can be present/soldered or not
- Debug UART headers can be present/soldered or not

Please state the exact required options in your use case when placing an order with sysmocom.

### 3.6.1 Enclosure option for sysmoQMOD

sysmoQMOD is also available in an enclosure. The product then consists of:

- sysmoQMOD PCBA
- 6x SIM card drawers
- USB A-B cable (1.8m)
- desktop 12V / 3A power supply (with EU cable included; different plugs on request)
- 8x Pigtail IPX/U.FL(m) / SMA(f,bulkhead) 13cm RG178
- 4x LED light guide Mentor 1270.1001
- 4x adapters for micro/nano SIM cards + ejector pins
- 10x M2x3 screws

## 3.7 LEDs

There are a series of LEDs on the board. They are separate in user-visible and non-user-visible LEDs:

The LEDs are grouped into four groups of two LEDs. Each group contains the LEDs for one slot:

- The WWAN LED as driven by the respective Modem
- A LED controlled by the SAM3 responsible for switching/emulating the SIM card associated with that modem.

Each group of two LEDs can be exposed to the front panel using a Mentor light guide 1270.1001.





Figure 16: Mentor 1270.1001 Light Guide

### 3.7.1 User-visible LEDs

Table 13: User-visible LEDs

LED Number	Connected to	Color	Description
LED101	Modem 1	Yellow	WWAN LED of Modem 1, under Modem control
LED601	ST12	Red	SIM1 LED of SAM3_ST12, under SAM3 firmware control
LED201	Modem 2	Yellow	WWAN LED of Modem 2, under Modem control
LED602	ST12	Red	SIM2 LED of SAM3_ST12, under SAM3 firmware control
LED301	Modem 3	Yellow	WWAN LED of Modem 3, under Modem control
LED701	ST34	Red	SIM3 LED of SAM3_ST34, under SAM3 firmware control
LED401	Modem 4	Yellow	WWAN LED of Modem 4, under Modem control
LED702	ST34	Red	SIM4 LED of SAM3_ST34, under SAM3 firmware control

#### Note

User-visible LEDs can be mounted on either the top side or bottom side of the PCBA, default is mounted on bottom side and determined before a production batch. If you require the user LEDs placement on top side or different colors, this is subject of MOQ and must be stated during inquiry/ordering.

### 3.7.2 Non-User-visible LEDs

These LEDs are primarily for diagnostic purposes and are not normally exposed to the end user.

Table 14: Non-User-visible LEDs

LED Number	Color	Description
LED801	Green	Main 3.3V DC voltage present
LED802	Green	Auxiliary 3.3V DC voltage present

## 4 SAM3 Firmware

The firmware for the SAM3 on the sysmoQMOD is currently undergoing some redesign to support more different board types and all the features of the sysmoQMOD.

In general, the firmware consists of two parts:

- the DFU bootloader, used for in-the-field updates of the SAM3 microcontrollers over USB
- the application program, installed and updated by the DFU bootloader, containing actual code for SIM card forwarding or emulation.

### NOTE

When communicating to the firmware via the serial ports (connectors X601 and X701), the speed on these ports is **115200** bps for firmware versions up to **0.5.1**, and it is **921600** bps for versions from **0.5.1.1** onwards. - Usage of these ports is not needed for normal operation of sysmoQMOD.

### 4.1 Updating the firmware via DFU

At sysmocom factory programming time, the SAM3 controllers are loaded with a DFU bootloader. It occupies the first 16kBytes of flash in the SAM3.

DFU is the official standard of the USB Implementers Forum for **Device Firmware Upgrades**. It is used by many USB devices of various manufacturers.

For more information about DFU, please see the USB DFU specification, which can be obtained from [http://www.usb.org/sites/default/files/DFU\\_1.1.pdf](http://www.usb.org/sites/default/files/DFU_1.1.pdf)

#### 4.1.1 DFU Basics

A DFU-capable device advertises its DFU capabilities via a DFU runtime descriptor which is exported during normal operation of the device (SAM3 application code is running). By sending special requests via USB, the device can be switched from application (runtime) mode into DFU (bootloader) mode.

Once the device enumerates in DFU mode, other DFU-specific requests on the control endpoint can be used to read (upload) and write (download) firmware from/to the device.

A device can export multiple **alternate settings**, and thereby export different memory types, flash partitions or the like. The DFU bootloader of the sysmoQMOD supports

- downloading of application code into volatile RAM (alternate setting 0)
- downloading of application code into the flash (alternate setting 1)

### NOTE

In almost all cases, you will want to install firmware images into the flash. Loading code into RAM is only used during development. You cannot use images linked for the flash in RAM (no relocation) or vice versa!

#### 4.1.2 Entering DFU mode from the runtime

The DFU mode can be entered in the following ways:

- by sending a DFU\_DETACH request from the host PC, or
- by placing a jumper across the RxD and TxD lines of the debug UART (X601/X701) during boot of the controller, or
- by erased/implausible flash content at the start of the application partition (offset 16k = 0x4000)

If either of those three conditions is fulfilled, the SAM3 will boot the DFU bootloader and thus expose DFU-mode USB interfaces, rather than the regular application program.

### 4.1.3 Updating the application using dfu-util

dfu-util is an open source implementation of the USB DFU protocol using libusb, running on Linux and other operating systems.

It can interact with DFU-capable devices and perform operations such as

- listing DFU-capable devices attached to the system
- switching devices from Runtime to DFU mode
- uploading firmware from the device to a file on the Linux host
- downloading firmware from a file on the Linux host to the device
- resetting a device to return from DFU mode to runtime mode

Firmware files for the sysmoQMOD have the naming convention `qmod-<appname>-dfu.bin`, for example `qmod-cardem-dfu.bin`.

You can use a command like

```
$ dfu-util -d 1d50:4004 -p 1-4.4.1.1 -a 1 -D bin/qmod-cardem-dfu.bin -R
```

from the USB host to install the firmware. 1-4.4.1.1 is the path at which the respective processor is attached, which is seen in `dmesg` output

```
[79086.696291] usb 1-4.4.1.1: New USB device found, idVendor=1d50, idProduct=4004
```

The output of `dfu-util` while flashing looks like this:

```
$ dfu-util -d 1d50:4004 -p 1-4.4.1.1 -a 1 -D bin/qmod-cardem-dfu.bin -R
dfu-util 0.9

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2016 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

dfu-util: Invalid DFU suffix signature
dfu-util: A valid DFU suffix will be required in a future dfu-util release!!!
Opening DFU capable USB device...
ID 1d50:4004
Run-time device DFU version 0100
Claiming USB DFU Runtime Interface...
Determining device status: state = appIDLE, status = 0
Device really in Runtime Mode, send DFU detach request...
Resetting USB...
Opening DFU USB Device...
Claiming USB DFU Interface...
Setting Alternate Setting #1 ...
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 0100
Device returned transfer size 512
Copying data from PC to DFU device
Download      [=====] 100%          28320 bytes
Download done.
state(7) = dfuMANIFEST, status(0) = No error condition is present
state(2) = dfuIDLE, status(0) = No error condition is present
Done!
dfu-util: can't detach
Resetting USB to switch back to runtime mode
```

## 4.2 Emergency recovery using SAM-BA

The SAM3 contains a mask-ROM boot loader called SAM-BA (SAM Boot Assist). When erasing the complete flash, this SAM-BA can be activated to perform an emergency recovery and/or to update the DFU bootloader.

---

### Note

It is recommended to only use this SAM-BA method as a last resort, if DFU based flashing as described above has failed, or if you need to update the DFU bootloader itself.

---

The complete flash can be erased the following ways:

- by activating the ERASE signal of the SAM3 while it starts (release from RESET), or
- by JTAG

On the sysmoQMOD, the ERASE signal of the SAM3 can be controlled from the respective other SAM3, i.e. U601 can control the ERASE (and RESET) signal of U701, and vice-versa. In case of severe malfunction or the requirement to update the DFU bootloader, this feature can be used to erase one SAM3 from the other SAM3, and then recover it using the SAM-BA ROM-loader.

### 4.2.1 Using bossac for flashing

`bossac` is an open source command-line utility program that can be used to talk to the SAM-BA loader inside Atmel SAM controllers such as the SAM3 on the sysmoQMOD.

Once a SAM3 has been placed into SAM-BA mode, you can use `bossac` over the Debug UART or over USB.

To program the SAM3 over the UART, you need to connect a 3.3V USB-UART cable (sysmocom CP2102) and an associated 2.5mm stereo-jack to 3-pin 2.54mm cable between the respective SAM3 UART header (X601/X701).

DFU bootloader firmware files have the naming convention `qmod-dfu-flash.bin`. Do not attempt to flash any application images, only the DFU bootloader image shall be flashed via SAM-BA!

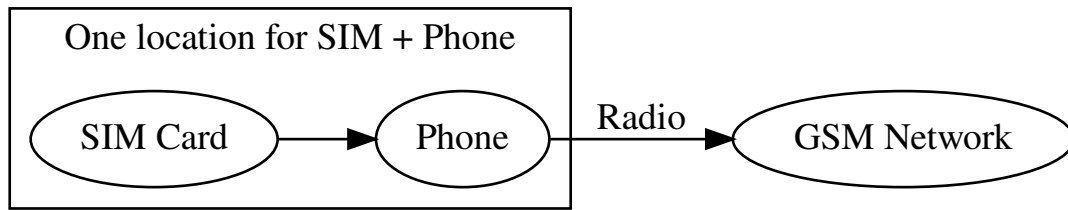
Example of `bossac` usage to flash the DFU bootloader into a chip:

```
$ bossac -e -w -v -b qmod-dfu-flash.bin -p ttyUSB0
Erase flash
Write 14748 bytes to flash
[=====] 100% (58/58 pages)
Verify 14748 bytes of flash
[=====] 100% (58/58 pages)
Verify successful
Set boot flash true
$
```

## 5 Remote SIM functionality

Normally, the SIM/USIM/RUIM card is physically co-located with the phone or cellular modem. The interconnection between SIM card and the cellular baseband processor is just a few centimeters of printed circuit trace. Replacing the SIM card requires physical access to the Phone, and swapping the cards frequently requires lots of manual interaction and associated system downtime.

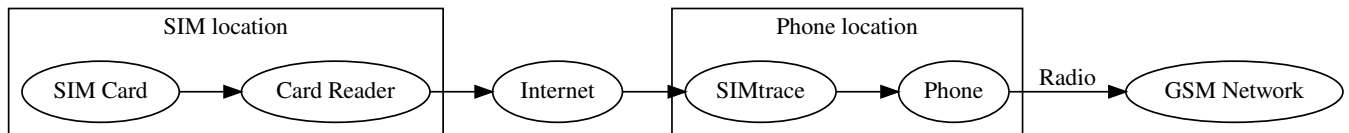
This system is fine for the normal use case where a human is carrying a cellphone, and the cellphone contains the SIM card of a cellular subscription which is changed only infrequently (every few weeks to months to year).



In some applications, it is useful to have a different set-up, where the SIM card is located remotely from the modem/phone.

This is the case in many remotely deployed systems, where a physical SIM card replacement would be very costly due to the logistics of sending the card there and having a human operator replace the card.

The **remote SIM** functionality solves this problem by forwarding the communication between phone and SIM virtually anywhere using TCP/IP transport protocol.



The **SIM side** and the **Radio side** can subsequently be at completely different physical/geographical locations, just as long as there is a transport channel between them.

## 5.1 Remote SIM Transport Channel

This transport channel can be high-latency and low-speed. It must only be capable of forwarding either all APDU data (simplistic implementation), or only the GSM/UMTS Authentication messages (optimized implementation with partial SIM card emulation on **Radio side**).

In fact, the transport channel in practise has shown to be able to use technologies like

- VSAT satellite links
- another set of physical SIM Card + Modem, so that the expensive roaming SIM is only used to tunnel the authentication data of a virtual other (e.g. non-roaming)SIM which is virtually provisioned in a dynamic way to that remote modem

## 5.2 Remote SIM Use Cases

There are many use case for such a setup, including

- Roaming / IREG testing systems
- Voice Quality testing systems
- Least Cost Routing of cellular data back-haul for vessels close to shore (e.g. using VSAT as transport channel)
- Interoperability testing to test different SIM cards with different modems

## 6 Host Software

There are two different options available in terms of software:

- `simtrace2-cardem-pcsc` as the most simple proof-of-concept. This is useful if you just want to quickly verify the correct operation of the sysmoQMOD product and/or its built-in card-emulation firmware
- `osmo-remsim` as a fully open source software suite to manage populations of remote SIM cards and card emulators. This is more geared towards production operation.

The reference software consists of command-line tools to be used on GNU/Linux based systems. Those programs are provided as C language source code and require the libraries **libosmocore** as well as **libusb** to operate.

The software components are provided as Free / Open Source Software.

`simtrace2-list` and `simtrace2-cardem-pcsc` can be found as part of the <https://git.osmocom.org/simtrace2> repository in the *host* subdirectory.

`osmo-remsim` can be found at <https://git.osmocom.org/osmo-remsim>

All related software is also provided as binary packages for Debian GNU/Linux, Ubuntu as well as Raspbian within the Osmocom binary package feeds, see [https://osmocom.org/projects/cellular-infrastructure/wiki/Binary\\_Packages](https://osmocom.org/projects/cellular-infrastructure/wiki/Binary_Packages)

### 6.1 simtrace2-list

The `simtrace2-list` utility is used to list the physical SIM card interfaces present on the system, to which a program like `simtrace2-remsim` could attach.

There is one line being printed for each interface. In the case of the sysmoQMOD product, you should see four such lines, one for each of the four modems.

```
$ ./simtrace2-list
1d50:4004 Addr=45, Path=1-4.4.1.4, Cfg=1, Intf=0, Alt=0: 255/2/0 (CardEmulator ↔
Modem 3)
1d50:4004 Addr=45, Path=1-4.4.1.4, Cfg=1, Intf=1, Alt=0: 255/2/0 (CardEmulator ↔
Modem 4)
1d50:4004 Addr=51, Path=1-4.4.1.1, Cfg=1, Intf=0, Alt=0: 255/2/0 (CardEmulator ↔
Modem 1)
1d50:4004 Addr=51, Path=1-4.4.1.1, Cfg=1, Intf=1, Alt=0: 255/2/0 (CardEmulator ↔
Modem 2)
```

This shows there are four SIM emulators attached to the system, as well as their associated USB parameters (VID:PID, Address, Path, Configuration, Interface, Altsetting).

### 6.2 simtrace2-cardem-pcsc

The `simtrace2-cardem-pcsc` program is a simple proof-of-concept to demonstrate the remote SIM functionality. In order to use this, you will need

- a `simtrace2-compatible` device attached to your Linux PC and which shows up in `simtrace2-list` (such as the sysmoQMOD)
- a smart card reader attached to your Linux PC, supported and exposed by `pcsc-lite`. We recommend USB-CCID compliant smart card readers, such as the Omnikey CardMan 3121 and 6121 series.

When you have the required hardware set-up and connected, you can start `simtrace2-cardem-pcsc`. After start, it will

- open the first available PC/SC reader on the system
- open the specified USB interface (see `-V`, `-P`, `-C`, `-I`, `-S` command line options)

- instruct the SIMTRACE2 compatible device to switch from local SIM to remote SIM mode
- instruct the SIMTRACE2 compatible device to perform a PERST (power enable reset) of the modem module, which causes implicit re-enumeration on the USB as well as re-reading of the SIM card during boot of the modem
- forward any APDUs between the modem and the SIM card inserted into the PC/SC reader

```
$ ./simtrace2-cardem-pcsc -V 0x1d50 -P 0x4004 -C 1 -I 0 -H 1-4.4.1.1 -n 0
simtrace2-cardem-pcsc - Remote SIM card forwarding
(C) 2010-2017 by Harald Welte <laforge@gnumonks.org>

SCardEstablishContext: OK

SCardListReaders: OK

SCardConnect: OK

<- 01 05 00 00 00 00 09 00 01
<- 02 02 00 00 00 00 09 00 01
<= cardem_request_set_atr(3b 9f 96 80 1f c7 80 31 a0 73 be 21 13 67 43 20 07 18 00 00 01 a5 ←
)
<- 01 02 00 00 00 00 1f 00 16 3b 9f 00 80 1f c7 80 31 a0 73 be 21 13 67 43 20 07 18 00 00 ←
01 a5
<- 02 01 00 00 00 00 0b 00 02 2c 01
Entering main loop
```

The following command-line options are relevant:

- -V to specify the USB vendor ID (0x1d50)
- -P to specify the USB product ID (0x4004 for quad-modem)
- -H to specify the USB path (1-4.4.1.1 here)
- -C to specify the USB configuration number (1)
- -I to specify the USB interface number (0 for first slot; 1 for second)
- -n to specify the PC/SC reader number (0 for first reader)

In a typical sysmoQMOD, you will have:

- -H x-xxx.1 for the controller serving the first two modems
- -H x-xxx.2 for the controller serving the second two modems
- -I 0 for the first modem of a given controller
- -I 1 for the second modem of a given controller.

The program will automatically switch the SPDT switches from local SIM to remote SIM, and will issue a reset pulse to the modem. You should then see follow-up message exchanges like given below:

```
URB:
-> 01 06 00 00 00 00 13 00 01 00 00 00 05 00 00 a4 00 04 02
=> DATA: flags=1, 00 a4 00 04 02 : CLA=00 INS=a4 P1=00 P2=04 P3=02; case=4, lc=2(0), le ←
=0(0)
<= cardem_request_pb_and_rx(a4, 2)
<- 01 01 00 00 00 00 0f 00 08 00 00 00 01 00 a4
URB:
-> 01 06 00 00 00 00 10 00 02 00 00 00 02 00 3f 00
=> DATA: flags=2, 3f 00 : CLA=00 INS=a4 P1=00 P2=04 P3=02; case=4, lc=2(2), le=0(0)
TX: 00 a4 00 04 02 3f 00
SCardEndTransaction: OK
```

Once there are no more messages exchanged for some time, the modem has read all relevant information from the SIM, and the modem should show you the IMSI of the card when issuing "AT+CIMI".

### 6.3 GSMTAP / wireshark APDU tracing

A protocol trace of the APDUs going back and forth between the SIM emulator hardware and the Card can be inspected in wireshark by means of GSMTAP encapsulated SIM Communication on UDP Port 4729 on the loopback device.

FIXME: Describe this in more detail.

### 6.4 osmo-remsim

For more details on the osmo-remsim software, please see

- osmo-remsim project homepage at <https://osmocom.org/projects/osmo-remsim/wiki>
- osmo-remsim user manual at <https://ftp.osmocom.org/docs/latest/osmo-remsim-usermanual.pdf>
- osmo-remsim video presentation at <https://media.ccc.de/v/osmocon2018-93-osmo-remsim-remote-sim-card-software>

## 7 Glossary

### 2FF

2nd Generation Form Factor; the so-called plug-in SIM form factor

### 3FF

3rd Generation Form Factor; the so-called microSIM form factor

### 3GPP

3rd Generation Partnership Project

### 4FF

4th Generation Form Factor; the so-called nanoSIM form factor

### A Interface

Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.008* [[3gpp-ts-48-008](#)])

### A3/A8

Algorithm 3 and 8; Authentication and key generation algorithm in GSM and GPRS, typically COMP128v1/v2/v3 or MILENAGE are typically used

### A5

Algorithm 5; Air-interface encryption of GSM; currently only A5/0 (no encryption), A5/1 and A5/3 are in use

### Abis Interface

Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.058* [[3gpp-ts-48-058](#)] and *3GPP TS 52.021* [[3gpp-ts-52-021](#)])

### ACC

Access Control Class; every BTS broadcasts a bit-mask of permitted ACC, and only subscribers with a SIM of matching ACC are permitted to use that BTS

### AGCH

Access Grant Channel on Um interface; used to assign a dedicated channel in response to RACH request

### AGPL

GNU Affero General Public License, a copyleft-style Free Software License



**AQPSK**

Adaptive QPSK, a modulation scheme used by VAMOS channels on Downlink

**ARFCN**

Absolute Radio Frequency Channel Number; specifies a tuple of uplink and downlink frequencies

**AUC**

Authentication Center; central database of authentication key material for each subscriber

**BCCH**

Broadcast Control Channel on Um interface; used to broadcast information about Cell and its neighbors

**BCC**

Base Station Color Code; short identifier of BTS, lower part of BSIC

**BTS**

Base Transceiver Station

**BSC**

Base Station Controller

**BSIC**

Base Station Identity Code; 16bit identifier of BTS within location area

**BSSGP**

Base Station Subsystem Gateway Protocol (*3GPP TS 48.018* [[3gpp-ts-48-018](#)])

**BVCI**

BSSGP Virtual Circuit Identifier

**CBC**

Cell Broadcast Centre; central entity of Cell Broadcast service

**CBCH**

Cell Broadcast Channel; used to transmit Cell Broadcast SMS (SMS-CB)

**CBS**

Cell Broadcast Service

**CBSP**

Cell Broadcast Service Protocol (*3GPP TS 48.049* [[3gpp-ts-48-049](#)])

**CC**

Call Control; Part of the GSM Layer 3 Protocol

**CCCH**

Common Control Channel on Um interface; consists of RACH (uplink), BCCH, PCH, AGCH (all downlink)

**Cell**

A cell in a cellular network, served by a BTS

**CEPT**

Conférence européenne des administrations des postes et des télécommunications; European Conference of Postal and Telecommunications Administrations.

**CGI**

Cell Global Identifier comprised of MCC, MNC, LAC and BSIC

**CSFB**

Circuit-Switched Fall Back; Mechanism for switching from LTE/EUTRAN to UTRAN/GERAN when circuit-switched services such as voice telephony are required.

**dB**

deci-Bel; relative logarithmic unit

**dBm**

deci-Bel (milliwatt); unit of measurement for signal strength of radio signals

**DHCP**

Dynamic Host Configuration Protocol (*IETF RFC 2131* [[ietf-rfc2131](#)])

**downlink**

Direction of messages / signals from the network core towards the mobile phone

**DSCP**

Differentiated Services Code Point (*IETF RFC 2474* [[ietf-rfc2474](#)])

**DSP**

Digital Signal Processor

**dnxload**

Tool to program UBL and the Bootloader on a sysmoBTS

**EDGE**

Enhanced Data rates for GPRS Evolution; Higher-speed improvement of GPRS; introduces 8PSK

**EGPRS**

Enhanced GPRS; the part of EDGE relating to GPRS services

**EIR**

Equipment Identity Register; core network element that stores and manages IMEI numbers

**ESME**

External SMS Entity; an external application interfacing with a SMSC over SMPP

**ETSI**

European Telecommunications Standardization Institute

**FPGA**

Field Programmable Gate Array; programmable digital logic hardware

**Gb**

Interface between PCU and SGSN in GPRS/EDGE network; uses NS, BSSGP, LLC

**GERAN**

GPRS/EDGE Radio Access Network

**GGSN**

GPRS Gateway Support Node; gateway between GPRS and external (IP) network

**GMSK**

Gaussian Minimum Shift Keying; modulation used for GSM and GPRS

**GPL**

GNU General Public License, a copyleft-style Free Software License

**Gp**

Gp interface between SGSN and GGSN; uses GTP protocol

**GPRS**

General Packet Radio Service; the packet switched 2G technology

**GPS**

Global Positioning System; provides a highly accurate clock reference besides the global position

**GSM**

Global System for Mobile Communications. ETSI/3GPP Standard of a 2G digital cellular network

**GSMTAP**

GSM tap; pseudo standard for encapsulating GSM protocol layers over UDP/IP for analysis

**GSUP**

Generic Subscriber Update Protocol. Osmocom-specific alternative to TCAP/MAP

**GT**

Global Title; an address in SCCP

**GTP**

GPRS Tunnel Protocol; used between SGSN and GGSN

**HLR**

Home Location Register; central subscriber database of a GSM network

**HNB-GW**

Home NodeB Gateway. Entity between femtocells (Home NodeB) and CN in 3G/UMTS.

**HPLMN**

Home PLMN; the network that has issued the subscriber SIM and has his record in HLR

**IE**

Information Element

**IMEI**

International Mobile Equipment Identity; unique 14-digit decimal number to globally identify a mobile device, optionally with a 15th checksum digit

**IMEISV**

IMEI software version; unique 14-digit decimal number to globally identify a mobile device (same as IMEI) plus two software version digits (total digits: 16)

**IMSI**

International Mobile Subscriber Identity; 15-digit unique identifier for the subscriber/SIM; starts with MCC/MNC of issuing operator

**IP**

Internet Protocol (*IETF RFC 791* [[ietf-rfc791](#)])

**IPA**

*ip.access GSM over IP* protocol; used to multiplex a single TCP connection

**Iu**

Interface in 3G/UMTS between RAN and CN

**IuCS**

Iu interface for circuit-switched domain. Used in 3G/UMTS between RAN and MSC

**IuPS**

Iu interface for packet-switched domain. Used in 3G/UMTS between RAN and SGSN

**LAC**

Location Area Code; 16bit identifier of Location Area within network

**LAPD**

Link Access Protocol, D-Channel (*ITU-T Q.921* [[itu-t-q921](#)])

**LAPDm**

Link Access Protocol Mobile (*3GPP TS 44.006* [[3gpp-ts-44-006](#)])

**LLC**

Logical Link Control; GPRS protocol between MS and SGSN (*3GPP TS 44.064* [[3gpp-ts-44-064](#)])

**Location Area**

Location Area; a geographic area containing multiple BTS

**LU**

Location Updating; can be of type IMSI-Attach or Periodic. Procedure that indicates a subscriber's physical presence in a given radio cell.

**M2PA**

MTP2 Peer-to-Peer Adaptation; a SIGTRAN Variant (*RFC 4165* [[ietf-rfc4165](#)])

**M2UA**

MTP2 User Adaptation; a SIGTRAN Variant (*RFC 3331* [[ietf-rfc3331](#)])

**M3UA**

MTP3 User Adaptation; a SIGTRAN Variant (*RFC 4666* [[ietf-rfc4666](#)])

**MCC**

Mobile Country Code; unique identifier of a country, e.g. 262 for Germany

**MTF**

Machine-to-Machine Form Factor; a SIM chip package that is soldered permanently onto M2M device circuit boards.

**MGW**

Media Gateway

**MM**

Mobility Management; part of the GSM Layer 3 Protocol

**MNC**

Mobile Network Code; identifies network within a country; assigned by national regulator

**MNCC**

Mobile Network Call Control; Unix domain socket based Interface between MSC and external call control entity like osmo-sip-connector

**MNO**

Mobile Network Operator; operator with physical radio network under his MCC/MNC

**MO**

Mobile Originated. Direction from Mobile (MS/UE) to Network

**MS**

Mobile Station; a mobile phone / GSM Modem

**MSC**

Mobile Switching Center; network element in the circuit-switched core network

**MSC pool**

A number of redundant MSCs serving the same core network, which a BSC / RNC distributes load across; see also the "MSC Pooling" chapter in OsmoBSC's user manual [[userman-osmobsc](#)] and 3GPP TS 23.236 [[3gpp-ts-23-236](#)]

**MSISDN**

Mobile Subscriber ISDN Number; telephone number of the subscriber

**MT**

Mobile Terminated. Direction from Network to Mobile (MS/UE)

**MTP**

Message Transfer Part; SS7 signaling protocol (*ITU-T Q.701* [[itu-t-q701](#)])

**MVNO**

Mobile Virtual Network Operator; Operator without physical radio network

**NCC**

Network Color Code; assigned by national regulator

**NITB**

Network In The Box; combines functionality traditionally provided by BSC, MSC, VLR, HLR, SMSC functions; see OsmoNITB

**NRI**

Network Resource Indicator, typically 10 bits of a TMSI indicating which MSC of an MSC pool attached the subscriber; see also the "MSC Pooling" chapter in OsmoBSC's user manual [\[userman-osmobsc\]](#) and *3GPP TS 23.236* [\[3gpp-ts-23-236\]](#)

**NSEI**

NS Entity Identifier

**NVCI**

NS Virtual Circuit Identifier

**NWL**

Network Listen; ability of some BTS to receive downlink from other BTSs

**NS**

Network Service; protocol on Gb interface (*3GPP TS 48.016* [\[3gpp-ts-48-016\]](#))

**OCXO**

Oven Controlled Crystal Oscillator; very high precision oscillator, superior to a VCTCXO

**OML**

Operation & Maintenance Link (ETSI/*3GPP TS 52.021* [\[3gpp-ts-52-021\]](#))

**OpenBSC**

Open Source implementation of GSM network elements, specifically OsmoBSC, OsmoNITB, OsmoSGSN

**OpenGGSN**

Open Source implementation of a GPRS Packet Control Unit

**OpenVPN**

Open-Source Virtual Private Network; software employed to establish encrypted private networks over untrusted public networks

**Osmocom**

Open Source MOBILE COMmunications; collaborative community for implementing communications protocols and systems, including GSM, GPRS, TETRA, DECT, GMR and others

**OsmoBSC**

Open Source implementation of a GSM Base Station Controller

**OsmoNITB**

Open Source implementation of a GSM Network In The Box, combines functionality traditionally provided by BSC, MSC, VLR, HLR, AUC, SMSC

**OsmoSGSN**

Open Source implementation of a Serving GPRS Support Node

**OsmoPCU**

Open Source implementation of a GPRS Packet Control Unit

**OTA**

Over-The-Air; Capability of operators to remotely reconfigure/reprogram ISM/USIM cards

**PC**

Point Code; an address in MTP

**PCH**

Paging Channel on downlink Um interface; used by network to page an MS

**PCP**

Priority Code Point (*IEEE 802.1Q* [?])

**PCU**

Packet Control Unit; used to manage Layer 2 of the GPRS radio interface

**PDCH**

Packet Data Channel on Um interface; used for GPRS/EDGE signalling + user data

**PIN**

Personal Identification Number; a number by which the user authenticates to a SIM/USIM or other smart card

**PLMN**

Public Land Mobile Network; specification language for a single GSM network

**PUK**

PIN Unblocking Code; used to unblock a blocked PIN (after too many wrong PIN attempts)

**RAC**

Routing Area Code; 16bit identifier for a Routing Area within a Location Area

**RACH**

Random Access Channel on uplink Um interface; used by MS to request establishment of a dedicated channel

**RAM**

Remote Application Management; Ability to remotely manage (install, remove) Java Applications on SIM/USIM Card

**RF**

Radio Frequency

**RFM**

Remote File Management; Ability to remotely manage (write, read) files on a SIM/USIM card

**Roaming**

Procedure in which a subscriber of one network is using the radio network of another network, often in different countries; in some countries national roaming exists

**Routing Area**

Routing Area; GPRS specific sub-division of Location Area

**RR**

Radio Resources; Part of the GSM Layer 3 Protocol

**RSL**

Radio Signalling Link (*3GPP TS 48.058* [[3gpp-ts-48-058](#)])

**RTP**

Real-Time Transport Protocol (*IETF RFC 3550* [[ietf-rfc3550](#)]); Used to transport audio/video streams over UDP/IP

**SACCH**

Slow Associate Control Channel on Um interface; bundled to a TCH or SDCCH, used for signalling in parallel to active dedicated channel

**SCCP**

Signaling Connection Control Part; SS7 signaling protocol (*ITU-T Q.711* [[itu-t-q711](#)])

**SDCCH**

Slow Dedicated Control Channel on Um interface; used for signalling and SMS transport in GSM

**SDK**

Software Development Kit

**SGs**

Interface between MSC (GSM/UMTS) and MME (LTE/EPC) to facilitate CSFB and SMS.

**SGSN**

Serving GPRS Support Node; Core network element for packet-switched services in GSM and UMTS.

**SIGTRAN**

Signaling Transport over IP (*IETF RFC 2719* [\[ietf-rfc2719\]](#))

**SIM**

Subscriber Identity Module; small chip card storing subscriber identity

**Site**

A site is a location where one or more BTSs are installed, typically three BTSs for three sectors

**SMPP**

Short Message Peer-to-Peer; TCP based protocol to interface external entities with an SMSC

**SMSC**

Short Message Service Center; store-and-forward relay for short messages

**SS7**

Signaling System No. 7; Classic digital telephony signaling system

**SS**

Supplementary Services; query and set various service parameters between subscriber and core network (e.g. USSD, 3rd-party calls, hold/retrieve, advice-of-charge, call deflection)

**SSH**

Secure Shell; *IETF RFC 4250* [\[ietf-rfc4251\]](#) to 4254

**SSN**

Sub-System Number; identifies a given SCCP Service such as MSC, HLR

**STP**

Signaling Transfer Point; A Router in SS7 Networks

**SUA**

SCCP User Adaptation; a SIGTRAN Variant (*RFC 3868* [\[ietf-rfc3868\]](#))

**syslog**

System logging service of UNIX-like operating systems

**System Information**

A set of downlink messages on the BCCH and SACCH of the Um interface describing properties of the cell and network

**TCH**

Traffic Channel; used for circuit-switched user traffic (mostly voice) in GSM

**TCP**

Transmission Control Protocol; (*IETF RFC 793* [\[ietf-rfc793\]](#))

**TFTP**

Trivial File Transfer Protocol; (*IETF RFC 1350* [\[ietf-rfc1350\]](#))

**TOS**

Type Of Service; bit-field in IPv4 header, now re-used as DSCP (*IETF RFC 791* [\[ietf-rfc791\]](#))

**TRX**

Transceiver; element of a BTS serving a single carrier

**TS**

Technical Specification

**u-Boot**

Boot loader used in various embedded systems

**UBI**

An MTD wear leveling system to deal with NAND flash in Linux

**UBL**

Initial bootloader loaded by the TI Davinci SoC

**UDP**

User Datagram Protocol (*IETF RFC 768* [\[ietf-rfc768\]](#))

**UICC**

Universal Integrated Chip Card; A smart card according to *ETSI TR 102 216* [\[etsi-tr102216\]](#)

**Um interface**

U mobile; Radio interface between MS and BTS

**uplink**

Direction of messages: Signals from the mobile phone towards the network

**USIM**

Universal Subscriber Identity Module; application running on a UICC to provide subscriber identity for UMTS and GSM networks

**USSD**

Unstructured Supplementary Service Data; textual dialog between subscriber and core network, e.g. *\*100 → Your extension is 1234*

**VAMOS**

Voice services over Adaptive Multi-user channels on One Slot; an optional extension for GSM specified in Release 9 of 3GPP GERAN specifications (*3GPP TS 48.018* [\[3gpp-ts-48-018\]](#)) allowing two independent UEs to transmit and receive simultaneously on traffic channels

**VCTCXO**

Voltage Controlled, Temperature Compensated Crystal Oscillator; a precision oscillator, superior to a classic crystal oscillator, but inferior to an OCXO

**VLAN**

Virtual LAN in the context of Ethernet (*IEEE 802.1Q* [\[ieee-802.1q\]](#))

**VLR**

Visitor Location Register; volatile storage of attached subscribers in the MSC

**VPLMN**

Visited PLMN; the network in which the subscriber is currently registered; may differ from HPLMN when on roaming

**VTY**

Virtual Teletype; a textual command-line interface for configuration and introspection, e.g. the OsmoBSC configuration file as well as its telnet link on port 4242

## A Bibliography / References

### References

- [1] [\[userman-ice1usb\]](#) Osmocom Project: icE1usb User Manual.
- [2] [\[userman-ogt\]](#) Pau Espin: osmo-gsm-tester User Manual.
- [3] [\[userman-remsim\]](#) Harald Welte: osmo-remsim User Manual.
- [4] [\[osmobts-abis-spec\]](#) Neels Hofmeyr & Harald Welte. OsmoBTS Abis Protocol Specification. <https://ftp.osmocom.org/docs/latest/osmobts-abis.pdf>



- [5] [userman-osmobsc] Osmocom Project: OsmoBSC User Manual. <https://ftp.osmocom.org/docs/latest/osmobsc-usermanual.pdf>
- [6] [vty-ref-osmobsc] Osmocom Project: OsmoBSC VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmobsc-vty-reference.pdf>
- [7] [userman-osmobts] Osmocom Project: OsmoBTS User Manual. <https://ftp.osmocom.org/docs/latest/osmobts-usermanual.pdf>
- [8] [vty-ref-osmobts] Osmocom Project: OsmoBTS VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmobts-trx-vty-reference.pdf> <https://ftp.osmocom.org/docs/latest/osmobts-sysmo-vty-reference.pdf> <https://ftp.osmocom.org/docs/latest/osmobts-lc15-vty-reference.pdf> <https://ftp.osmocom.org/docs/latest/osmobts-oc2g-vty-reference.pdf> <https://ftp.osmocom.org/docs/latest/osmobts-octphy-vty-reference.pdf> <https://ftp.osmocom.org/docs/latest/osmobts-virtual-vty-reference.pdf>
- [9] [userman-osmocbc] Osmocom Project: OsmoCBC User Manual. <https://ftp.osmocom.org/docs/latest/osmocbc-usermanual.pdf>
- [10] [vty-ref-osmocbc] Osmocom Project: OsmoCBC VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmocbc-vty-reference.pdf>
- [11] [userman-osmogbproxy] Osmocom Project: OsmoGBProxy User Manual. <https://ftp.osmocom.org/docs/latest/osmogbproxy-usermanual.pdf>
- [12] [vty-ref-osmogbproxy] Osmocom Project: OsmoGBProxY VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmogbproxy-vty-reference.pdf>
- [13] [userman-osmoggsn] Osmocom Project: OpenGGSN User Manual. <https://ftp.osmocom.org/docs/latest/osmoggsn-usermanual.pdf>
- [14] [vty-ref-osmoggsn] Osmocom Project: OsmoGGSN VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmoggsn-vty-reference.pdf>
- [15] [userman-osmohlr] Osmocom Project: OsmoHLR User Manual. <https://ftp.osmocom.org/docs/latest/osmohlr-usermanual.pdf>
- [16] [vty-ref-osmohlr] Osmocom Project: OsmoHLR VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmohlr-vty-reference.pdf>
- [17] [userman-osmohnbgw] Osmocom Project: OsmoHNBGW User Manual. <https://ftp.osmocom.org/docs/latest/osmohnbgw-usermanual.pdf>
- [18] [vty-ref-osmohnbgw] Osmocom Project: OsmoHNBGW VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmohnbgw-vty-reference.pdf>
- [19] [userman-osmomgw] Osmocom Project: OsmoMGW User Manual. <https://ftp.osmocom.org/docs/latest/osmomgw-usermanual.pdf>
- [20] [vty-ref-osmomgw] Osmocom Project: OsmoMGW VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmomgw-vty-reference.pdf>
- [21] [userman-osmomsc] Osmocom Project: OsmoMSC User Manual. <https://ftp.osmocom.org/docs/latest/osmomsc-usermanual.pdf>
- [22] [vty-ref-osmomsc] Osmocom Project: OsmoMSC VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmomsc-vty-reference.pdf>
- [23] [userman-osmonitb] Osmocom Project: OsmoNITB User Manual. <https://ftp.osmocom.org/docs/latest/osmonitb-usermanual.pdf>
- [24] [vty-ref-osmonitb] Osmocom Project: OsmoNITB VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmonitb-vty-reference.pdf>

- [25] [userman-osmopcu] Osmocom Project: OsmoPCU User Manual. <https://ftp.osmocom.org/docs/latest/-osmopcu-usermanual.pdf>
- [26] [vty-ref-osmopcu] Osmocom Project: OsmoPCU VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmopcu-vty-reference.pdf>
- [27] [userman-osmosgsn] Osmocom Project: OsmoSGSN User Manual. <https://ftp.osmocom.org/docs/latest/-osmosgsn-usermanual.pdf>
- [28] [vty-ref-osmosgsn] Osmocom Project: OsmoSGSN VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmosgsn-vty-reference.pdf>
- [29] [userman-osmosipconnector] Osmocom Project: OsmoSIPconnector User Manual. <https://ftp.osmocom.org/docs/latest/osmosipconnector-usermanual.pdf>
- [30] [vty-ref-osmosipconnector] Osmocom Project: OsmoSIPconnector VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmosipconnector-vty-reference.pdf>
- [31] [userman-osmosmlc] Osmocom Project: OsmoSMLC User Manual. <https://ftp.osmocom.org/docs/latest/-osmosmlc-usermanual.pdf>
- [32] [vty-ref-osmosmlc] Osmocom Project: OsmoSMLC VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmosmlc-vty-reference.pdf>
- [33] [userman-osmostp] Osmocom Project: OsmoSTP User Manual. <https://ftp.osmocom.org/docs/latest/osmostp-usermanual.pdf>
- [34] [vty-ref-osmostp] Osmocom Project: OsmoSTP VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/-osmostp-vty-reference.pdf>
- [35] [userman-osmotrx] Osmocom Project: OsmoTRX User Manual. <https://ftp.osmocom.org/docs/latest/osmotrx-usermanual.pdf>
- [36] [vty-ref-osmotrx] Osmocom Project: OsmoTRX VTY Reference Manual. <https://ftp.osmocom.org/docs/latest/osmotrx-uhd-vty-reference.pdf> <https://ftp.osmocom.org/docs/latest/osmotrx-lms-vty-reference.pdf> <https://ftp.osmocom.org/docs/latest/osmotrx-ipc-vty-reference.pdf> <https://ftp.osmocom.org/docs/latest/-osmotrx-usrp1-vty-reference.pdf>
- [37] [3gpp-ts-23-041] 3GPP TS 23.041: Technical realization of Cell Broadcast Service (CBS)
- [38] [3gpp-ts-23-048] 3GPP TS 23.048: Security mechanisms for the (U)SIM application toolkit; Stage 2 <https://www.3gpp.org/DynaReport/23048.htm>
- [39] [3gpp-ts-23-236] 3GPP TS 23.236: Intra-domain connection of Radio Access Network (RAN) nodes to multiple Core Network (CN) nodes <https://www.3gpp.org/DynaReport/23236.htm>
- [40] [3gpp-ts-24-007] 3GPP TS 24.007: Mobile radio interface signalling layer 3; General Aspects <https://www.3gpp.org/DynaReport/24007.htm>
- [41] [3gpp-ts-24-008] 3GPP TS 24.008: Mobile radio interface Layer 3 specification; Core network protocols; Stage 3. <https://www.3gpp.org/dynareport/24008.htm>
- [42] [3gpp-ts-31-101] 3GPP TS 31.101: UICC-terminal interface; Physical and logical characteristics <https://www.3gpp.org/DynaReport/31101.htm>
- [43] [3gpp-ts-31-102] 3GPP TS 31.102: Characteristics of the Universal Subscriber Identity Module (USIM) application <https://www.3gpp.org/DynaReport/31102.htm>
- [44] [3gpp-ts-31-103] 3GPP TS 31.103: Characteristics of the IMS Subscriber Identity Module (ISIM) application <https://www.3gpp.org/DynaReport/31103.htm>
- [45] [3gpp-ts-31-111] 3GPP TS 31.111: Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) <https://www.3gpp.org/DynaReport/31111.htm>

- [46] [3gpp-ts-31-115] 3GPP TS 31.115: Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications <https://www.3gpp.org/DynaReport/31115.htm>
- [47] [3gpp-ts-31-116] 3GPP TS 31.116: Remote APDU Structure for (U)SIM Toolkit applications <https://www.3gpp.org/DynaReport/31116.htm>
- [48] [3gpp-ts-35-205] 3GPP TS 35.205: 3G Security; Specification of the MILENAGE algorithm set: General
- [49] [3gpp-ts-35-206] 3GPP TS 35.206: 3G Security; Specification of the MILENAGE algorithm set: Algorithm specification <https://www.3gpp.org/DynaReport/35206.htm>
- [50] [3gpp-ts-44-006] 3GPP TS 44.006: Mobile Station - Base Station System (MS - BSS) interface; Data Link (DL) layer specification <https://www.3gpp.org/DynaReport/44006.htm>
- [51] [3gpp-ts-44-018] 3GPP TS 44.018: Mobile radio interface layer 3 specification; Radio Resource Control (RRC) protocol <https://www.3gpp.org/DynaReport/44018.htm>
- [52] [3gpp-ts-44-064] 3GPP TS 44.064: Mobile Station - Serving GPRS Support Node (MS-SGSN); Logical Link Control (LLC) Layer Specification <https://www.3gpp.org/DynaReport/44064.htm>
- [53] [3gpp-ts-45-002] 3GPP TS 45.002: Digital cellular telecommunications system (Phase 2+) (GSM); GSM/EDGE Multiplexing and multiple access on the radio path <https://www.3gpp.org/DynaReport/45002.htm>
- [54] [3gpp-ts-48-008] 3GPP TS 48.008: Mobile Switching Centre - Base Station system (MSC-BSS) interface; Layer 3 specification <https://www.3gpp.org/DynaReport/48008.htm>
- [55] [3gpp-ts-48-016] 3GPP TS 48.016: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN) interface; Network service <https://www.3gpp.org/DynaReport/48016.htm>
- [56] [3gpp-ts-48-018] 3GPP TS 48.018: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN); BSS GPRS protocol (BSSGP) <https://www.3gpp.org/DynaReport/48018.htm>
- [57] [3gpp-ts-48-049] 3GPP TS 48.049: Digital cellular communications system; Base Station Controller - Cell Broadcast Centre (BSC-CBC) interface specification; Cell Broadcast Service Protocol (CBSP) <https://www.3gpp.org/DynaReport/48049.htm>
- [58] [3gpp-ts-48-056] 3GPP TS 48.056: Base Station Controller - Base Transceiver Station (BSC - BTS) interface; Layer 2 specification <https://www.3gpp.org/DynaReport/48056.htm>
- [59] [3gpp-ts-48-058] 3GPP TS 48.058: Base Station Controller - Base Transceiver Station (BSC - BTS) Interface; Layer 3 specification <https://www.3gpp.org/DynaReport/48058.htm>
- [60] [3gpp-ts-51-011] 3GPP TS 51.011: Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface
- [61] [3gpp-ts-51-014] 3GPP TS 51.014: Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface <https://www.3gpp.org/DynaReport/51014.htm>
- [62] [3gpp-ts-52-021] 3GPP TS 52.021: Network Management (NM) procedures and messages on the A-bis interface <https://www.3gpp.org/DynaReport/52021.htm>
- [63] [etsi-tr102216] ETSI TR 102 216: Smart cards [https://www.etsi.org/deliver/etsi\\_tr/102200\\_102299/102216/-03.00.00\\_60/tr\\_102216v030000p.pdf](https://www.etsi.org/deliver/etsi_tr/102200_102299/102216/-03.00.00_60/tr_102216v030000p.pdf)
- [64] [etsi-ts102221] ETSI TS 102 221: Smart Cards; UICC-Terminal interface; Physical and logical characteristics [https://www.etsi.org/deliver/etsi\\_ts/102200\\_102299/102221/13.01.00\\_60/ts\\_102221v130100p.pdf](https://www.etsi.org/deliver/etsi_ts/102200_102299/102221/13.01.00_60/ts_102221v130100p.pdf)
- [65] [etsi-ts101220] ETSI TS 101 220: Smart Cards; ETSI numbering system for telecommunication application providers [https://www.etsi.org/deliver/etsi\\_ts/101200\\_101299/101220/12.00.00\\_60/ts\\_101220v120000p.pdf](https://www.etsi.org/deliver/etsi_ts/101200_101299/101220/12.00.00_60/ts_101220v120000p.pdf)
- [66] [ieee-802.1q] IEEE 802.1Q: Bridges and Bridged Networks <https://ieeexplore.ieee.org/document/6991462>
- [67] [ietf-rfc768] IETF RFC 768: User Datagram Protocol <https://tools.ietf.org/html/rfc768>
- [68] [ietf-rfc791] IETF RFC 791: Internet Protocol <https://tools.ietf.org/html/rfc791>

- [69] [ietf-rfc793] IETF RFC 793: Transmission Control Protocol <https://tools.ietf.org/html/rfc793>
- [70] [ietf-rfc1035] IETF RFC 1035: Domain Names - Implementation and Specification <https://tools.ietf.org/html/rfc1035>
- [71] [ietf-rfc1350] IETF RFC 1350: Trivial File Transfer Protocol <https://tools.ietf.org/html/rfc1350>
- [72] [ietf-rfc2131] IETF RFC 2131: Dynamic Host Configuration Protocol <https://tools.ietf.org/html/rfc2131>
- [73] [ietf-rfc2474] IETF RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers <https://tools.ietf.org/html/rfc2474>
- [74] [ietf-rfc2719] IETF RFC 2719: Signal Transport over IP <https://tools.ietf.org/html/rfc2719>
- [75] [ietf-rfc3331] IETF RFC 3331: Message Transfer Part 2 User Adaptation Layer <https://tools.ietf.org/html/rfc3331>
- [76] [ietf-rfc3550] IETF RFC 3550: RTP: A Transport protocol for Real-Time Applications <https://tools.ietf.org/html/rfc3550>
- [77] [ietf-rfc3596] IETF RFC 3596: DNS Extensions to Support IP Version 6 <https://tools.ietf.org/html/rfc3596>
- [78] [ietf-rfc3868] IETF RFC 3868: SCCP User Adaptation Layer <https://tools.ietf.org/html/rfc3868>
- [79] [ietf-rfc4165] IETF RFC 4165: Message Transfer Part 2 Peer-to-Peer Adaptation Layer <https://tools.ietf.org/html/rfc4165>
- [80] [ietf-rfc4251] IETF RFC 4251: The Secure Shell (SSH) Protocol Architecture <https://tools.ietf.org/html/rfc4251>
- [81] [ietf-rfc4666] IETF RFC 4666: Message Transfer Part 3 User Adaptation Layer <https://tools.ietf.org/html/rfc4666>
- [82] [ietf-rfc5771] IETF RFC 5771: IANA Guidelines for IPv4 Multicast Address Assignments <https://tools.ietf.org/html/rfc5771>
- [83] [itu-t-q701] ITU-T Q.701: Functional Description of the Message Transfer Part (MTP) <https://www.itu.int/rec/T-REC-Q.701/en/>
- [84] [itu-t-q711] ITU-T Q.711: Functional Description of the Signalling Connection Control Part <https://www.itu.int/rec/T-REC-Q.711/en/>
- [85] [itu-t-q713] ITU-T Q.713: Signalling connection control part formats and codes <https://www.itu.int/rec/T-REC-Q.713/en/>
- [86] [itu-t-q714] ITU-T Q.714: Signalling connection control part procedures <https://www.itu.int/rec/T-REC-Q.714/en/>
- [87] [itu-t-q921] ITU-T Q.921: ISDN user-network interface - Data link layer specification <https://www.itu.int/rec/T-REC-Q.921/en>
- [88] [smpp-34] SMPP Developers Forum. Short Message Peer-to-Peer Protocol Specification v3.4 [https://docs.nimta.com/SMPP\\_v3\\_4\\_Issue1\\_2.pdf](https://docs.nimta.com/SMPP_v3_4_Issue1_2.pdf)
- [89] [gnu-agplv3] Free Software Foundation. GNU Affero General Public License. <https://www.gnu.org/licenses/agpl-3.0.en.html>
- [90] [freeswitch\_pbx] FreeSWITCH SIP PBX <https://freeswitch.org>