Dissecting modern (3G/4G) cellular modems

Harald Welte, Holger Hans Peter Freyther

This talk

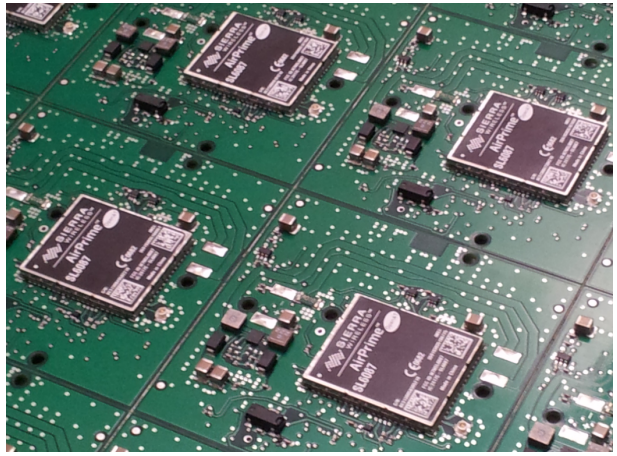
- Our motivation
- A bit of History
- Selecting a device
- An unexpected surprise
- Firmware upgrade
- Outlook/Recommendations/Wishes

Motivation

- Implementing GSM specifications for the last decade (OpenMoko, Osmocom)
- 8 years since *Anatomy of Smartphone Hardware* at 25C3
- 7 years since OsmocomBB for GSM
- Used and built M2M devices using 2G modems at work
- so we're looking for a modem that can be used for
  - our next-generation M2M/embedded devices
  - testing/logging/tracing Osmocom 3G/4G network-side software
  - building more tools to help understanding cellular technology

Cellular Modems in M2M

- Assume you want to build a M2M device
- Classic approach to M2M/Embedded cellular:
  - Cellular modem with AT commands over Serial/USB
  - Main Processor runs M2M application
- if you run Application in Modem, you can save PCB space, power and BOM cost
  - OpenAT by Sierra Wireless
    - Write C code using OpenAT APIs
    - Dynamically loaded into the RTOS
    - Runs without privilege separation, MMU
    - Protocol to multiplex AT, log, debug
    - Discontinued HW platform ⇒ Locked in
    - Various other limitations



Device requirements

Our requirements for a good modem

- Ability to run application code inside modem
- Avoid modem supplier vendor lock-in (EOL, ...)
- Get textual logging when handling messages
- Get a copy of the radio network messages and export to GSMTAP
  - Like Tobias Engels x-goldmon
  - But for all GPRS, EGPRS, UMTS and LTE messages

Qualcomm DIAG protocol

- Qualcomm DIAG in many products (DVB-H, GSM, ...)
- Presented by Guillaume Delugre at 28C3
- Simple HDLC frame (0x7e), cmd, data, CRC16
- Events, Logging, Command/Response
- Thousands of different message structures
- ModemManager, gsm-parser consume only a small fraction

Selecting a device

- Old Option Icon 225 stick exposes DIAG out of the box
- Quectel UC20 (2G+3G) expose DIAG by default
  - but no LTE support
- Quectel EC20 (2G+3G+4G) expose DIAG by default
  - 2G, 3G and 4G sounds quite nice
  - EC20 not only a LGA solder module but also as mini-PCIe
    - convenient for early testing / prototyping without custom board
- EC20 using a Qualcomm MDM9615 chipset
  - Also used in the iPhone5
  - Almost no documentation on MDM9615 available
  - Still, a good candidate for starting our research…

An unexpected surprise

Firmware update, hints of Linux

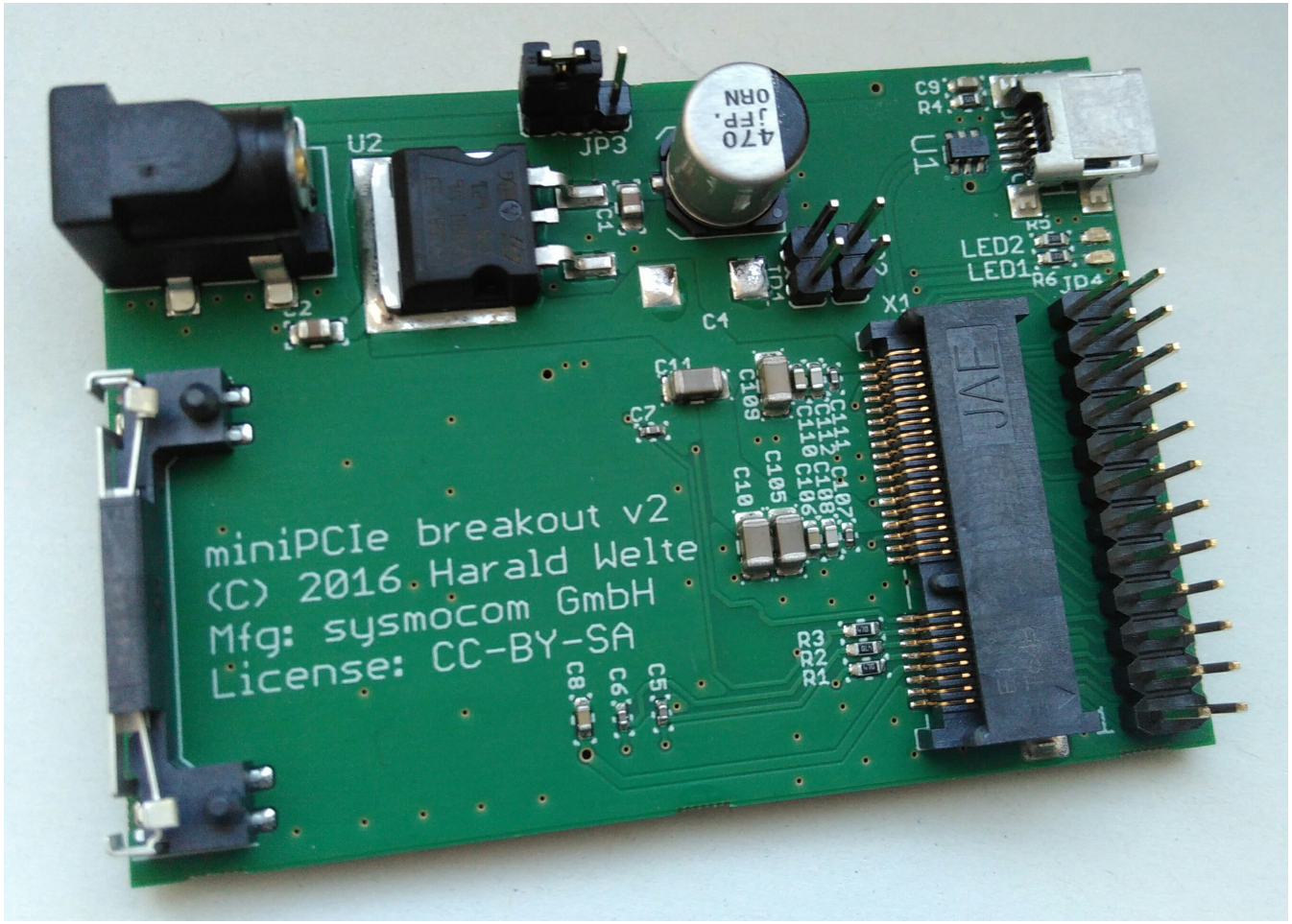
- Got a firmware upgrade to fix stability / bugs
- Looks like it contains traces of Linux?
- Looks like it uses fastboot for the update
- Other people have already found Linux in MDM9615 based products (e.g Mickey Shkatov at DEFCON 23)
- But why would there be Linux inside a Modem?
  - Qualcomm is known for their REX/AMSS on Hexagon baseband ?!?
- And if it contains Linux, GPL requires them to mention that, include License text and provide source code ?!?

GPL compliance

- No written offer, let's see if it runs Linux
- Armijn Hemels **`gpltool.git`** has **`unyaffs`** to unpack yaffs
- **`strings`**, etc. clearly reveal Linux, glibc, busybox
  - other interesting strings like **`AT+QLINUXCMD=?`** show up
- The fun and exploration begins…
  - technical analysis (serial console, firmware reversing, …)
  - legal enforcement to get source code of GPL/LGPL components (Harald is founder of gpl-violations.org)

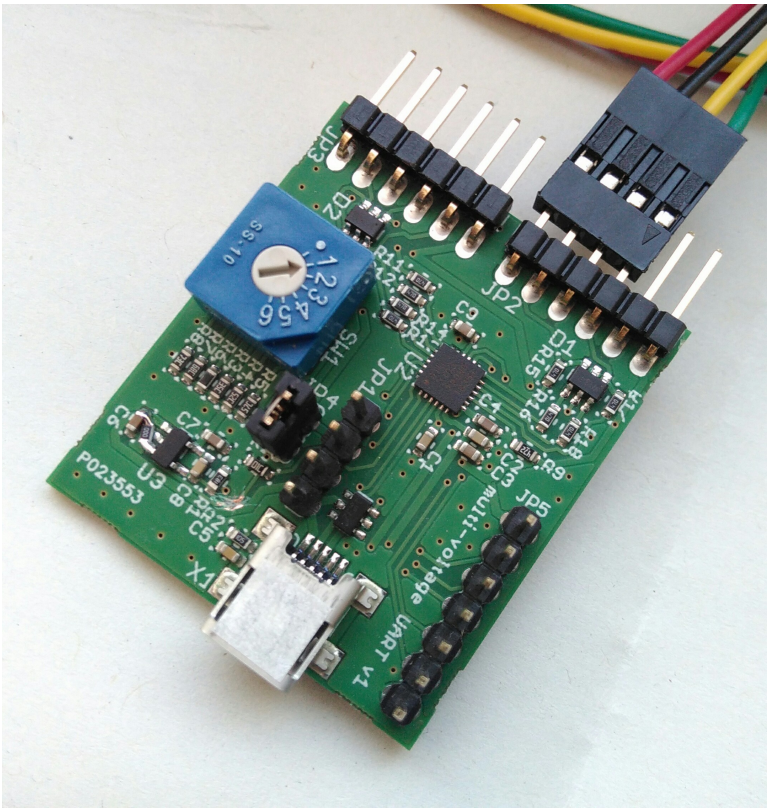
Hardware based analysis

- mPCIe modules often expose additional signals like PCM audio on non-standard pins
- existing PC/embedded mainboards don't use those signals
- create Osmocom mPCIe-breakout board to access those signals
- https://osmocom.org/projects/mpcie-breakout/wiki



Serial Console

- EC20 solder module documents DBG_UART pinout, but not all modules have it enabled?
- serial console is at 1.8V, but the 1.8V supply is not accessible (so not easy to add external level shifter / Vref)
- create Osmocom multi-voltage USB-UART with selectable 1.8, 2.3, 2.5, 2.8, 3.0 and 3.3V logic level
- https://osmocom.org/projects/mv-uart/wiki
- root password (DES hash): **`oelinux123`**



Retro-fitting Serial Console to mPCIe module

- unfortunately the DBG_UART on the LGA module solder pads is not exposed to mPCIE
- some soldering required to retro-fit a 2.54mm header:

GPL compliance

- Linux basis created by Qualcomm and used by Quectel
  - https://wiki.codeaurora.org/xwiki/bin/QLBEP/
  - Many branches, releases, which to use?

> I tried instruction above to build yaffs2 for MDM9615, so I downloaded source **`M9615AAAARNLZA1611161.xml`** but during compilation I faced some libs that are missing such as libQMI and acdb-loader..
>
> — Tonino Perazzi

Releases on release branch

Last modified by Android QAEP Service on 2016/12/23 08:59

Date	Tag / Build ID	Chipset	Manifest
December 23, 2016	LE.UM.1.0.6-01110-8x96.0	apq8096-drone	LE.UM.1.0.6-01110-8x96.0.xml
December 21, 2016	LE.BR.1.2.1.1-12900-9x07	mdm9607	LE.BR.1.2.1.1-12900-9x07.xml

GPL compliance

Asking for the complete and corresponding source

- The source code of Qflash tool in Linux is attached, [...]

Asking again for the complete and corresponding source

> We never been in legal dispute and we always make sure to understand IPR ahead of using technology belonging to third party.
>
> — Quectel

Build a Smarter World

QUECTEL

IPR Declaration Letter

Dated: 14th June, 2016

Quectel Wireless Solutions Co., Ltd.("Quectel") respects the importance of intellectual property rights and respective laws. As such, Quectel has actively engaged with known essential IPR owners and successfully secured licensing to legally use their technologies in **Quectel GSM/GPRS Module, Quectel WCDMA Module, Quectel LTE Module** **("Quectel Products")**. Quectel

GPL compliance

Asking for the complete and corresponding source

> We appreciate the efforts that your client had put into the open source project netfilter/*iptable*. However, [...] **your client does not have the right to empower the copyright**. We think software netfilter/iptable is built on the code operating system *GUN*/Linux, thus subject to GPL terms, where FSF requires that each author of code incorporated in FSF projects either provide copyright assignment to FSF or disclaim copyright. Therefore, It seems that **your client does not have the copyright on netfilter/iptable.**
>
> As one of the leading providers of wireless solution, **Quectel is always respectful IPR**. We would like to compliant with GPL and do some necessary statements，including a disclaimer or appropriate notices. Under the terms of GPL, we would like to dedicate Kernel code of EC25x to free software community.
>
> — Quectel

GPL compliance

Asking for the complete and corresponding source

> Many thanks for your detailed explanations GPL/LGPL license terms and the practical methods. I will carefully study your suggestions again and find a proper way to open GLP/LGPL licensed software. Basically, we will simply provide a tarball of open source for download at this time. And release the git repositories in next step.
>
> — Quectel

Asking for the complete and corresponding source

> We are always willing to achieve GPL compliance.
>
> — Quectel

Asking for the complete and corresponding source

> So we need some time to know of all things and construct the Open Source projects. Within a short time, we cannot construct a perfect web site to present Open Source things now. However, we will continue to do like that.
>
> — Quectel

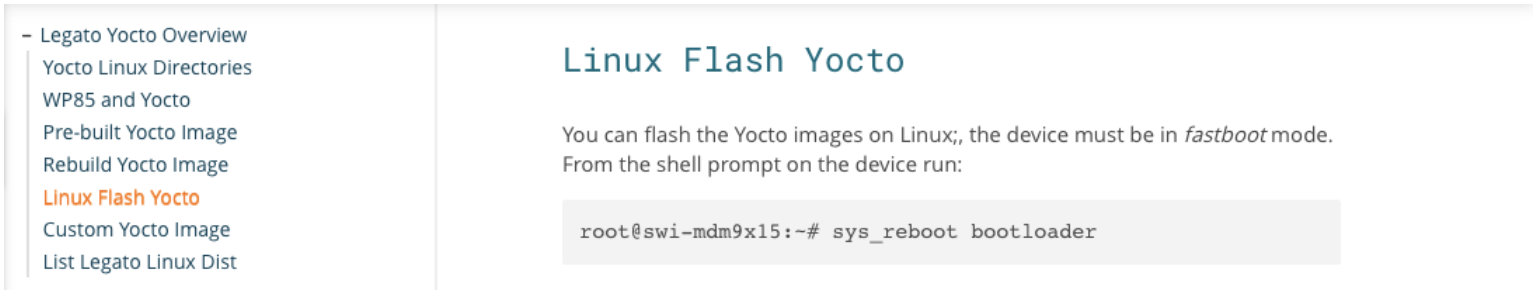
GPL compliance

Your tarball is missing some files

> We have issued all GPL licensed source code. **We have no the xt_dscp file in the project, and nor Qulacomm**. It must be caused by your compilation environment. If you have more question or problem during the development with Quectel module, please add my Skype ID (XXXXX), I will continue to support you on Skype.
> **The email will not discuss the compiling issue any more.**
>
> — Quectel

GPL compliance

- … many months later
  - we have received various source tarballs
  - they contain not only GPL/LGPL code but other FOSS code (thanks!)
  - full license compliance still not achieved, but improving…
- Sierra Wireless Legato is a positive example of a competitor
  - they not only provide the OE/Linux source but extensive documentation!
  - but they try to lure customers into a proprietary Legato framework, and thus again vendor-lock-in :(



MDM 9615 HW and SW

Qualcomm Hardware

- Qualcomm MDM9615 chipset
- Used in the iPhone 5 and automotive
- Modems like Quectel EC20, Sierra Wireless MC7355
- No public HW documentation?!
- Either not many people study it or are not allowed to share?

MDM 9615 HW Overview

- ????

How to access the system?

- serial console requires soldering re-work and is slow
- easy mechanism to get shell and transfer files from/to target
- Android **`adbd`** present on the modem but not exposed via USB
- it's possible to re-configure the Linux kernel Android USB Gadget:
  - **`AT+QLINUXCMD="/usr/bin/usb_uartdiag"`**
  - device re-enumerates with different composite USB interfaces
- Linux kernel driver on host needs patching (static interface mapping assumption)
  - patches available in **`quectel-experiments.git`**, documented in wiki

MDM 9615 AP SW Overview

The software stack seems to be called **Qualcomm LE**

- Android Bootloader
- Android Linux kernel
- Android Debug Bridge (adb)
- but: GNU libc, busybox userland
- Using OpenEmbedded to build images
- Developed and maintained by Qualcomm

Qualcomm Linux kernel overview

- Qualcomm Android Linux kernel
- Huge changes compared to mainline **`git diff -w | wc -l`**
  - **`v3.0.21`** in EC20: 1.5 million lines
  - **`v3.18.20`** in EC25: 1.9 million lines
- Expected: CPU + peripheral drivers
- Less expected:
  - smem_log (shared memory logging)
  - ipc_log (inter-processOR communication)
  - remote spinlocks

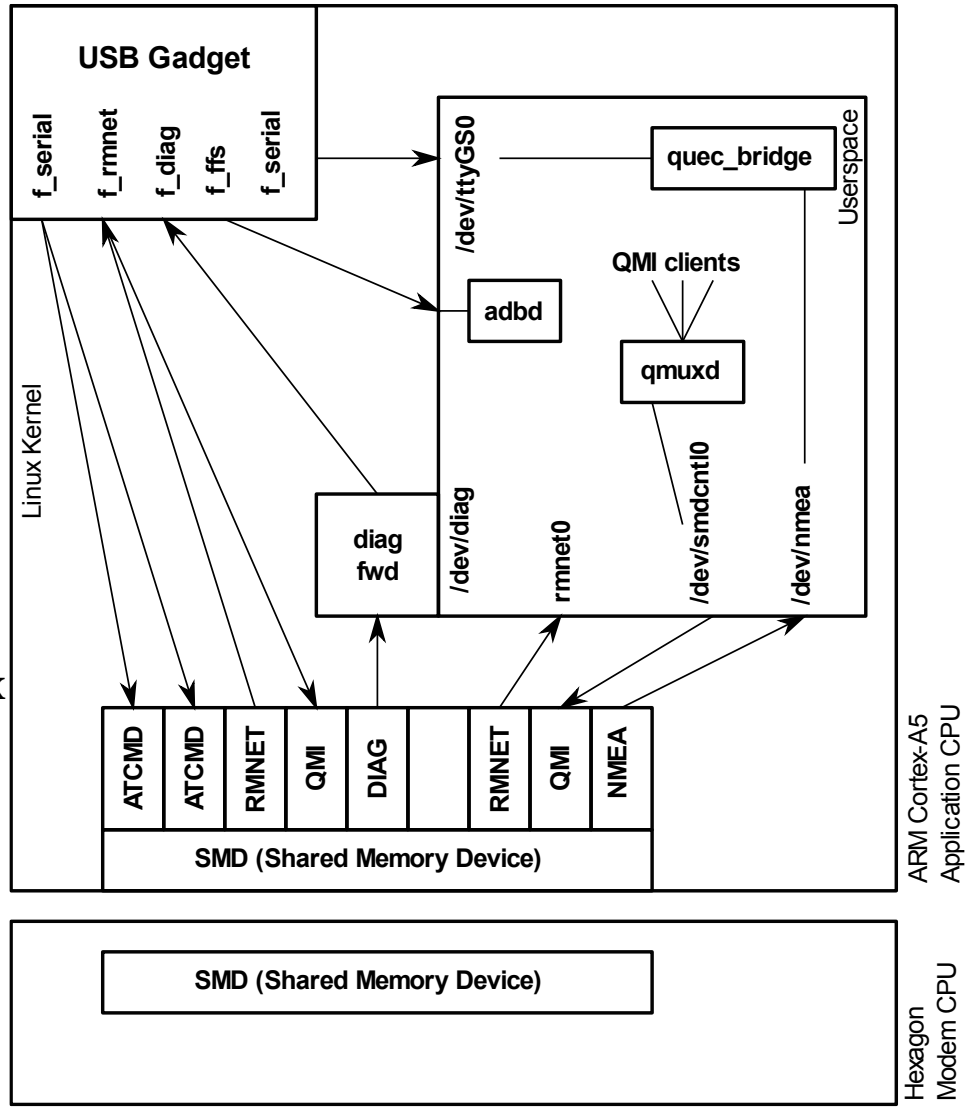
Qualcomm Linux kernel subsystems

Some of the Qualcomm-specific kernel sub-systems

SMD	Shared Memory Device
IPC	Inter Processor Communications
RMNET	Remote Network
BAM	Bus Access Manager
IPA	Internet Packet Accelerator
DIAGFWD	DIAG Forwarding
AF_MSM_IPC	Socket family for Qualcomm IPC

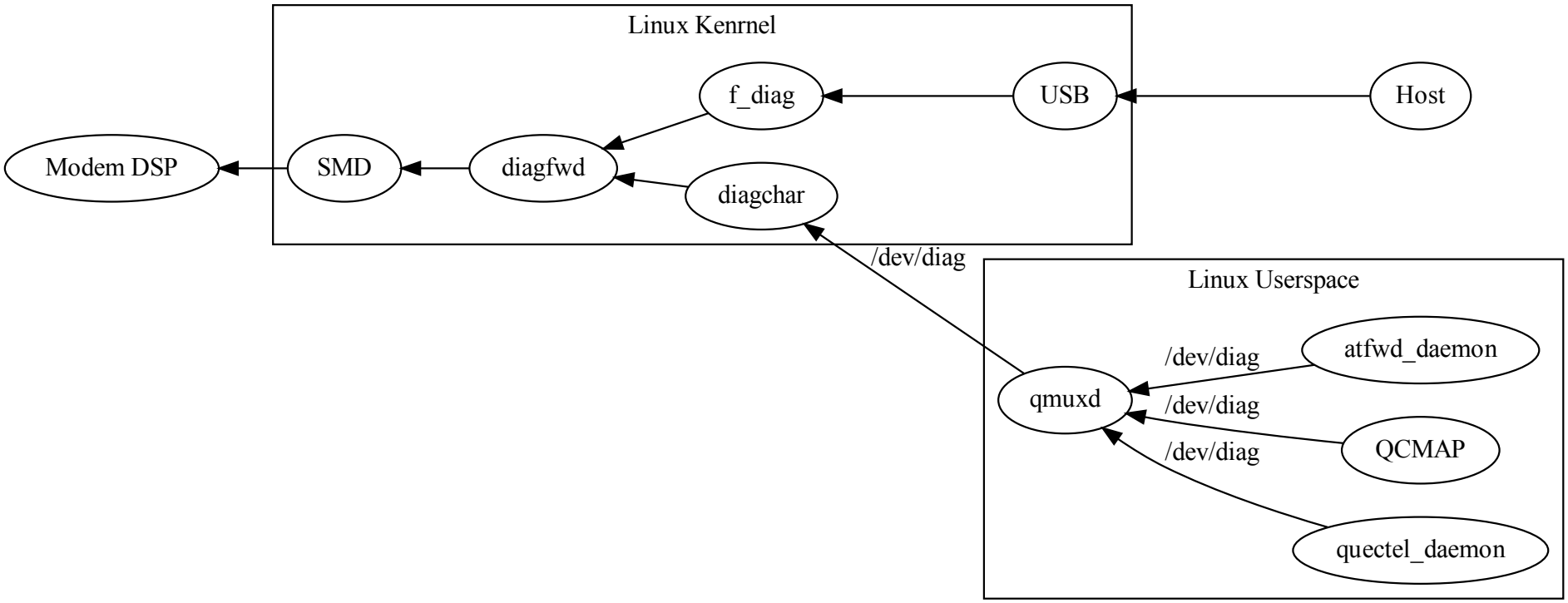
Qualcomm LE System Architecture

- simplified block diagram
- USB interface fully controlled by Linux AP
  - very complex Qualcomm Android USB Gadget
  - some endpoints mapped to SMD queues
  - other endpoints handled by *regular* Linux
  - GPS NMEA takes completely different path than AT commands, despite both being serial ports?
  - DIAG and QMI handled in more complex ways



DIAG in Qualcomm LE

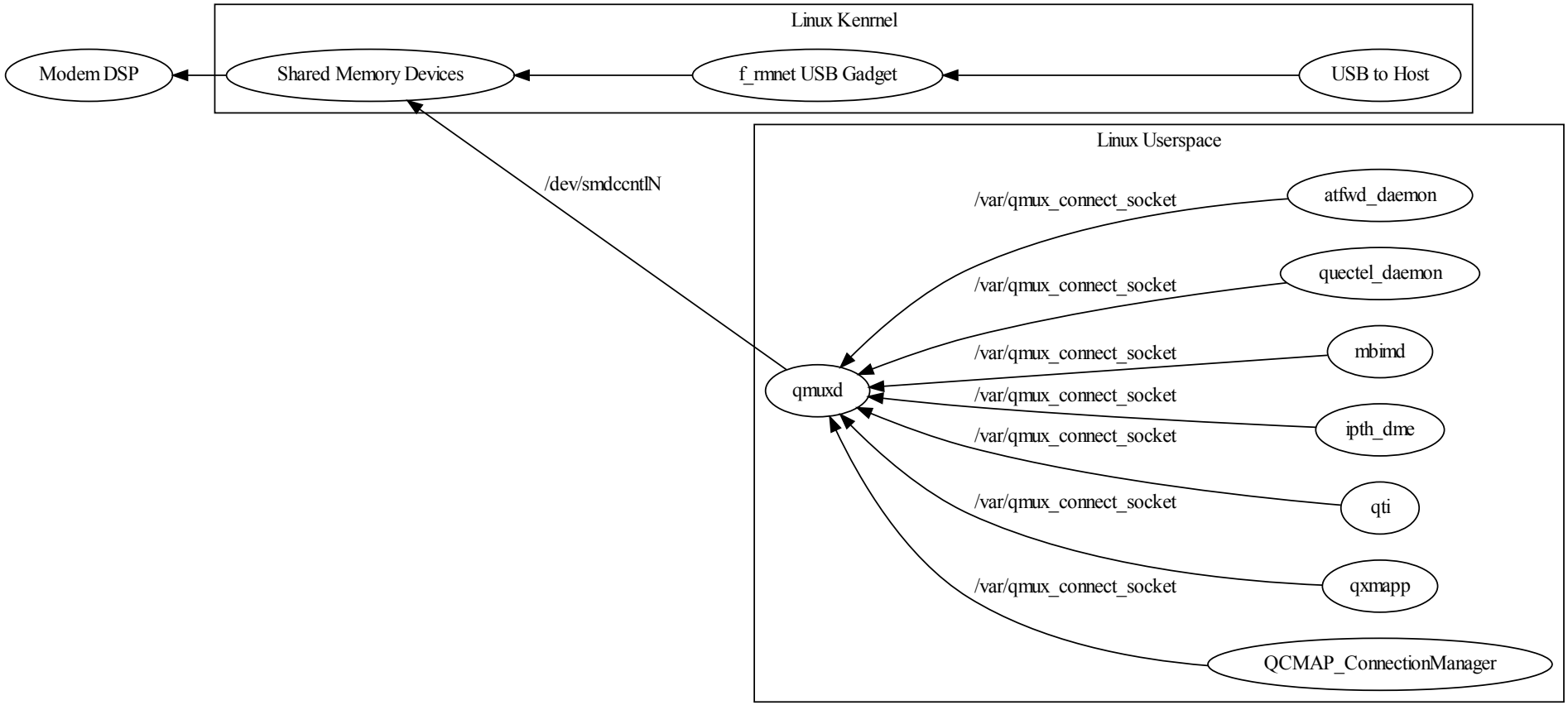
- DIAG interface of Modem exposed on SMD
- diagfwd distributes messages between USB, SMD and **`/dev/diagchar`**
- Linux userspace processes don't use syslog, but diag msg for logging via **`libdiag.so`**



QMI in Qualcomm LE

every **`rmnet`** data device has associated QMI control

- on your Linux PC: **`qmi_wwan`** and **`/dev/cdc-wdm`**
- on Qualcomm LE modem: **`/dev/smdcntlN`**, multiplexed by **`qmuxd`**



Tools for analysis

We created some tools to help our analysis

- used OE to build matching **`opkg`** and OE packages for **`socat`**, **`lsof`**, **`strace`**
- FOSS programs for the Linux AP linked against proprietary **`libqmi-framework.so`**
  - **`qmi_test`**: Simple program to read IMEI via QMI
  - **`atcop_test`**: Test program to implement AT commands in Linux userspace
- 100% FOSS programs
  - **`qmuxd_wrapper`**: LD_PRELOAD wrapper for tracing between **`qmuxd`** and QMI clients
  - **`libqmi-glib`** transport support for **`qmuxd`** (work in progress)
  - **`osmo-qcdiag`**: Host tool for obtaining DIAG based logs from Linux programs + QMI traces, decoded via **`libmi-glib`**

Userspace programs

We found a bunch of proprietary Linux userspace programs

`adbd`	Implements Android Debug Bridge
`atfwd_daemon`	Implement Quectel-Specific AT Commands
`quectel_daemon`	?; various ASoC related bits
`qti`	?
`mbim`	Mobile Broadband IF Model (translates MBIM to QMI)
`QCMAP_ConnectionManager`	runs linux-base WiFi AP/router with LTE backhaul
`quec_bridge`	reads GPS NMEA from **`/dev/nmea`** and writes it to **`/dev/ttyGS0`**

Funny bits + pieces

Funny AT commands

- **`AT+QLINUXCMD`**, e.g. switch usb config to get adb
  - arbitrary shell commands executed as root on r/w rootfs!
- **`AT+QFASTBOOT`**, switch to the bootloader
- **`AT+QPRINT`**, print dmesg
- AT for **`system("echo mem > /sys/power/state")`**

How many processes does it take to reboot a system?

- **`rebootdiagapp`** registers DIAG command (cmd code 0x29)
  - spawns thread that runs **`system("qmi_simple_ril_test input=/tmp/reset")`**
  - **`system("echo 'modem reset' > /tmp/reset")`**
    - makes **`qmi_simple_ril_test`** send a QMI message to modem
  - **`system("rm /tmp/reset")`**
  - writes "REBOOT" to **`/dev/rebooterdev`** this time using **`fwrite()`**!
- **`reboot_daemon`** reads **`/dev/rebooterdev`**

```
read_count = read(pipe_fd,buf,MAX_BUF-1);
/* if read REBOOT_STR, then call reboot */
if(strncmp(buf,REBOOT_STR,strlen(REBOOT_STR)) == 0) {
    debug_printf("going for reboot\n");
    printf("reboot-daemon: initiating reboot\n");
    system("reboot");
}
```

C programs that look like shell scripts

- strings /usr/bin/quectel_daemon

```
echo "nau8814-aif1" > /sys/devices/platform/soc-audio.0/tx_dai_name
cp -f /cache/usb/qcfg_usbcfg /etc/; cp -f /cache/usb/usb /etc/init.d/
echo 90 >/sys/kernel/debug/pm8xxx-pwm-dbg/0/duty-cycle
pkill -f "/bin/sh /usr/bin/nmea_demon.sh"
ps ef | grep "quec_bridge /dev/nmea /dev/ttyGS0" | grep -v grep
cd /cache/ufs;ls
```

Firmware upgrade

recovery and applypatch

- Qualcomm uses recovery.git from Android ~4.0
- Updates are zip files with deltas, SHA1+RSA
- recovery started on boot, drives applypatch

```
// Look for an RSA signature embedded in the .ZIP file comment given
// the path to the zip.  Verify it matches one of the given public
// keys.
```

Qualcomm EC20 firmware upgrade

- Based on the recovery.git code
- But for some reason using RedBend for the update (legacy?)
- RSA still linked into the binary but not used
- RedBend used by many more companies and systems (e.g. Quectel UC20, automotive)

RedBend (delta update) software

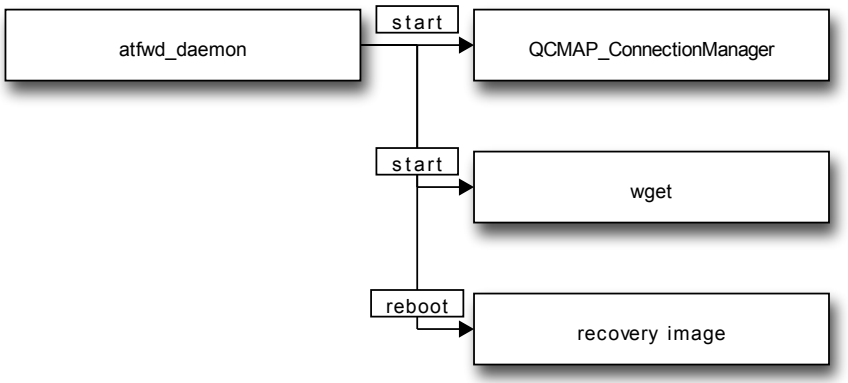
- Used in OMA DeviceManagement as well? (e.g. Mathew Solnik)
- Lots of starring at hexdumps, lots of help from Dieter Spaar
- Created tools to partially extract and create .diff files
- Heavy in pointers/offsets, not robust
- Crashes on crafted files
- Not cryptographically signed!

CRC32	Length	ID header							RAM size		Sector size	
Offset into ToC	Compr size of ToC	Alloc size	copy	upd	ins	del	deld	dir	dell	lnk	crup	crins

Firmware upgrade overview

```
$ strings atfwd_daemon | egrep
"wget|QCMAP|fota|update.z"

... QCMAP_ConnectionManager
/etc/mobileap_cfg.xml n n
fotanet
/usr/bin/wget -T 20 -t 3 %s -O
%s
mv %s %s && mkdir -p /cache/fota
&& echo %s > %s
/cache/fota/ipth_config_dfs.txt
rm -rf /cache/fota /cache/recovery /cache/update.zip
Start download fota for update.zip
```



- atfwd_daemon can be asked to start upgrade
- Configure APN, specify URL, store result to update.zip
- Add status and reboot to recovery
- Apply update.zip and reboot

Recommendation to modem vendors

- It is great to have an open and accessible Qualcomm based modem for further research and developing custom applications/extensions
- Security issues (particularly unverified FOTA) must be fixed
- We need security from attackers *without locking out the user/owner*
  - If vendors introduce verified boot and/or FOTA, allow owner specified keys!
- Please keep it open, good for learning and many applications
- Allow owners to modify the software of their device
- Secure the FOTA upgrading with owner specified keys

Status and Outlook

- Status today
  - Osmocom wiki with all our findings public now!
  - debug tools (**`osmo-qcdiag`**, LD_PRELOAD wrapper, **`qmi_test`**, etc.) released
  - mpcie-breakout + mv-uart released + available
  - **`libqmi-glib`** integration WIP
- Outlook
  - we hope to grow documentation in wiki
  - please help us out: read code, play with devices + update wiki
  - OE/opkg package feed planned
  - aim is to have 100% FOSS userland on Cortex-A5

Unrelated Announcement

- Osmocom project has gained support for 3G/3.5G during 2016
- Osmocom suffers from lack of contributions :(
- We want to motivate more contributions
  - *Accelerate 3.5G* programme provides 50 free 3.5 femtocells to contributors
  - tell us how you would use your free femtocell to improve Osmocom
  - Call for Proposals runs until January 31st, 2017.
  - see http://sysmocom.de/downloads/accelerate_3g5_cfp.pdf

Questions

- Questions?

Links

- Our results / hacks
  - https://osmocom.org/projects/quectel-modems
  - git://git.osmocom.org/quectel-experiments.git
  - git://git.osmocom.org/osmo-qcdiag.git
  - ftp://ftp.osmocom.org/quectel (mirrored)
- Collection of links for further study
  - ftp://ftp2.quectel.com/OpenSrc/
  - https://wiki.codeaurora.org/xwiki/bin/QLBEP/
  - https://events.ccc.de/congress/2011/Fahrplan/attachments/2022_11-ccc-qcombbdbg.pdf
  - https://media.defcon.org/DEF%20CON%2023/DEF%20CON%2023%20presentations/DEFCON-23-Mickey-Shkatov-Jesse-Michael-Scared-poopless-LTE-and-your-laptop-UPDATED.pdf
  - https://github.com/2b-as/xgoldmon
  - https://www.blackhat.com/docs/us-14/materials/us-14-Solnik-Cellular-Exploitation-On-A-Global-Scale-The-Rise-And-Fall-Of-The-Control-Protocol.pdf