FreeCalypso

A fully liberated GSM baseband

Mychaela Falconia

REcon Montreal 2017

The problem of the baseband

- Proprietary baseband/modem/radio processors are an insult to personal computing freedom
- The problem is even worse for those who prefer non-smart cellphones, as the baseband is the entirety of the phone

Prior art: OsmocomBB

- A toy-only from-scratch reimplementation of the upper layers of the mobile-side GSM protocol stack, using knowledge from leaked sources for driving Calypso hardware and DSP
- It’s a dead project: the capabilities and quality of OsmocomBB solution today are the same as they were at the 27C3 presentation in 2010

Alternatives to OsmocomBB:
using leaked TI sources directly

In 2011 I started collecting leaked sources and docs, creating a GSM mini-Wikileaks central repository:

ftp://ftp.freecalypso.org/pub/GSM/

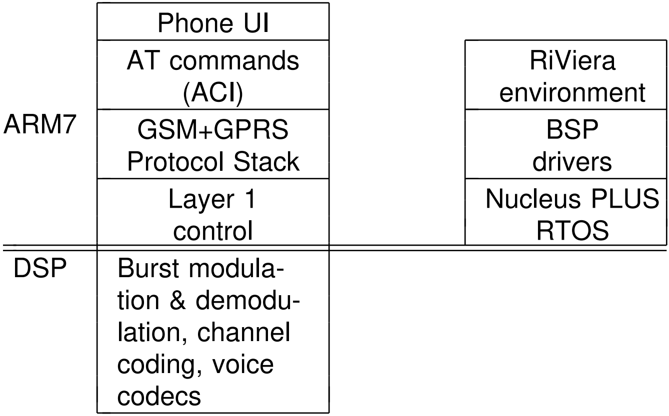
Three major TI source leaks:

Source leak	HW platform
TSM30	Oddball
TCS211 semi-src	Calypso+Iota+Rita, DSP 36
Peek	LoCosto

TCS211 modem firmware semi-source

- The most essential basis for FreeCalypso
- Wrongfully withheld from Humanity for years
- OsmocomBB used it despite their denials — damning evidence will be shown
- Liberated in the fall of 2013 thanks to a valiant Russian Comrade

TCS211 firmware architecture



First attempt at blob-free GSM firmware

Throughout 2014 and 2015, I attempted to put together a totally blob-free fw for the Calypso, compiling with gcc instead of TI's proprietary compiler, by pulling bits from different source leaks and reintegrating from the bottom up.

After arduous effort we got this fw to kinda-sorta-work, but it was plagued by bugs resulting from mismatches between TCS211 and LoCosto and from having too many variables changed all at once.

Then I realized that a different approach is called for...

Deblobbing of the Calypso firmware

- The two major components of TCS211 fw that came in binary-only form are Layer 1 and the G23M protocol stack
- The G23M PS is chipset-independent, thus wholesale replacement of this component with the LoCosto version (full C source) is feasible — and has been done successfully — but it was still non-trivial!
- L1 is very chipset-dependent; for the chipset of interest to us we got all original *.h files for L1, but only *.obj instead of *.c
- The C source for L1 for the right chipset has been painstakingly reconstructed in a labour of love

Reconstruction of the Calypso L1 source

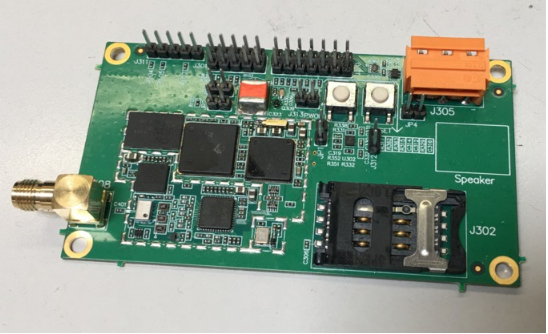
- Our starting point: *.obj and *.h files for the right chipset, plus full source for the wrong chipset
- Objective: L1 for the right chipset in recompilable C source form
- I took each individual L1 C module from LoCosto, dropped it into the TCS211 environment (compiler, *.h files, everything else), and massaged it until it compiled into an exact match to the original TCS211 binary object.
- I wrote my own disassembler highly customized to the COFF ABI and symbolic info produced by TI's TMS470 compiler: I refuse to use IDA

The firmware itself is not all that’s needed!

All supporting tools and accessories had to be developed anew by yours truly:

- Tools for loading our own code into RAM and flash on Calypso devices
- Tools for communicating with running firmwares: decoding and displaying debug trace, sending commands to the fw, extensions of our own invention
- Tools for working with TI’s flash file system

Our own hardware



TI Calypso+Iota+Rita chipset

What about an equivalent for 3G+?

- Supposedly there exist leaked LTE modem sources from Qualcomm, but I haven't seen them myself
- MTK source leaks are mostly binary objects
- Someone else would need to lead the 3G+ project — I am too invested into Calypso
- Enormous amount of work even with source and documentation leaks as rich as the ones from TI

Hard-to-beat strengths of FreeCalypso GSM/2G solution

- Building our own hardware means that we can package it in any desired form factor

- The liberated modem functionality includes all protocol stack processing and the control aspects of Layer 1 — in contrast, the known ‘‘source’’ leaks from MTK and Qualcomm have these parts in binary objects

Morally superior alternative: Form a GSM village

- Do not bow down to forcible imposition of unwanted 3G/4G technologies
- Call your carrier and tell them: we don’t want 4G LTE, we want 2G forever
- OpenBTS and Osmocom Cellular Infrastructure projects: set up our own GSM/2G networks
- If all else fails, move to a third world village where we can set up our own GSM network without anyone coming to shut it down

What the Mother uses in the interim

- My ultimate goal is to build my own dumbphone hardware based on FreeCalypso, but I have to use *something* in the meantime
- I use a Pirelli DP-L10: same Calypso core chipset as in the dream phone I wish to build, but with a bunch of undocumented peripherals
- Running Pirelli's original proprietary firmware: practically usable libre fw on this hw model is unlikely
- This phone's proprietary fw is close enough to TI's baseline that most FreeCalypso tools work with it: debug trace and file system access

The ultimate goal of FreeCalypso

- Build a “dumbphone” just like the Pirelli DP-L10, but without all those extra chips, so it can run FreeCalypso firmware.
- Build a physical GSM user community in some remote corner of the world (a real-life version of Themyscira) where we can have GSM/2G service **forever**, without ever being forced into 3G or 4G or 8G or whatever.
- What about those who do want 3G/4G of their own free will and desire? My answer: OK *as long as* at least 1 MHz of spectrum (5 GSM channels) remains reserved for the GSM/2G minority forever.

How YOU can help FreeCalypso **now**

- Support the GSM/2G minority by increasing our numbers: get an old 2G phone (ebay etc) and use it **instead of** 3G/4G.
- Our FreeCalypso phone has not been built yet, so all currently available phones are proprietary — but a proprietary phone that runs on GSM/2G only is still morally superior to 3G/4G ones.
- Your carrier will see that you are using their legacy 2G network instead of their marketed 3G/4G one; the more people use 2G, the more incentive for them to keep those legacy networks.

For more information:

www.freecalypso.org