GPL Workshop
How to (not?) use Free Software

by

Harald Welte <laforge@gpl-violations.org>

Contents

Introduction

Who is speaking to you?

- an independent Free Software developer
- who earns his living off Free Software since 1997
- who is one of the authors of the Linux kernel firewall system called netfilter/iptables
- who has started gpl-violations.org to enforce license com pliance
- who IS NOT A LAWYER

Disclaimer

Legal Disclaimer

- All information presented here is provided on an as-is bas is
- There is no warranty for correctness of legal information
- The author is not a lawyer
- This does not comprise legal advise
- The authors' experience is limited to German copyright l aw

Ideas and Goals of the GNU GPL

- Free Software
  - Software that has fundamental freedoms:
    - to use it for any purpose
    - to "help your neighbour" (i.e. make copies)
    - to study it's functionality (reading source code)
    - to fix it myself (make modifications and run them)
- Copyleft
  - Is the legal idea to
    - exercising copyright to grant the above freedoms
    - assure that nobody can take away the freedom
- The GNU General Public License
  - Is a legal instrument to apply they copyleft idea on softwa re

The GNU GPL revisited

Revisiting the GNU General Public License

- Regulates distribution of copyrighted code, not usage
- Allows distribution of source code and modified source code
  - The license itself is mentioned
  - A copy of the license accompanies every copy
- Allows distribution of binaries or modified binaries, if
  - The license itself is mentioned
  - A copy of the license accompanies every copy
  - The complete source code is either included with the copy (alternatively a written offer to send the source code on request to any 3rd party)

Complete Source Code

"... complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable."

- For standard C-language programs, this means:
  - Source Code
  - Makefiles
  - compile-time Configuration (such as kernel .config)
- General Rule:
  - Intent of License is to enable user to run modified versions of the program. They need to be enabled to do so.

Derivative Works

- What is a derivative work?
  - Not dependent on any particular kind of technology (static/dynamic linking, dlopen, whatever)
  - Even while the modification can itself be a copyrightable work, the combination with GPL-licensed code is subject to GPL.
  - As soon as code is written for a specific non-standard API (such as the iptables plugin API), there is significant indication for a derivative work
  - This position has been successfully enforced out-of-court with two Vendors so far (iptables modules/plugins).

Derivative Works

- Binary-only kernel modules
  - In-kernel proprietary code (binary kernel modules) are hard to claim GPL compliant
  - Case-by-case analysis required, as the level of integration into the GPL licensed kernel code depends on particular c ase
  - IBM is in the process of getting rid of all binary-only kernel modules. There are exceptions, but they are very clear ones (such as a filesystem port to linux, where the filesystem code already existed under another OS)
  - There is no general acceptance or tolerance to binary-only kernel modules in the Linux (development) community. Not even Linus himself has ever granted an exception for such modules!

Derivative Works

- Glue Code
  - Acts as glue layer between GPL licensed code and proprietary code
  - Some Vendors think they can avoid the GPL by doing so
  - Is definitely not a bullet-proof legal solution, especially when it is clearly visible that the only purpose of this glue code is to "get rid" of the GPL.

Derivative Works

- Moral Issues
  - Apart from what is legally possible, there are moral issues
  - Even if in a particular case there is no legal way to claim a binary-only kernel module is a derivative work, you might still be acting against the authors' wishes
  - By shipping binary-only kernel modules, you violate the "moral code of conduct" of the Free Software community
  - But it is the work of this very community that enables you to build your product based on Free Software
  - Such action might have long-term detrimental effects on the motivation of FOSS developers (dissatisfaction, demotivation, ...)

Collective Works

"... it is not the intent .. to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works ..."

- GPL controls "collective works"

"... mere aggregation of another work ... with the program on a volume of a storage or distribution medium does not bring the other work und the scope of this license"

- GPL allows "mere aggregation"
  - like a general-porpose GNU/Linux distribution (SuSE, Red Hat, ...)

GPL And Embedded Systems

- Historical background:
  - The GPL was written for userspace programs running on existing operating systems
  - Covering a whole OS (and even userspace programs) is not an ideal match, but if you read it carefully it still makes sense

- Toolchain:

"... the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable."

  - Practical case:
    - You've modified gcc for a specific embedded platform
    - Therefore, this gcc is not "normally distributed with the operating system" and you have to distribute it together with the source code
    - gcc itself is covered under GPL, so you need to provide binaries and source co de(!)

GPL And Embedded Systems

- The "Scripts"
  - (scripts to control compilation and installation, see earlier slide)
  - In case of embedded hardware, the "scripts" include:
    - Tools for generating the firmware binary from the source (even if they are technically no 'scripts')
- Embedded DRM
  - Intent of License is to enable user to run modified versions of the program. They need to be enabled to do s o.
  - Result: Signing binaries and only accepting signed versions from the bootloader (without providing the signature key or a possibility to set a new key in the bootloader) is not acceptable!

Practical Source Code Offer

- Some Rules
  - The "complete corresponding source code" has to be made available
  - It has to be made available for each and every object-code version that was distributed
  - If you strip down the source code offer (e.g. remove proprietary source code), try to see whether the result actually compiles
  - If the product is mixed free / proprietary software, consider including the proprietary parts (as object code) in the "source code package", so the full firmware image can be rebuilt without having to tear apart an existing image and ripping out those proprietary programs from t here.

The biggest myths about the GPL

The biggest myths about the GPL

- The GPL is not enforcible
- Software licensed under GPL has no copyright
- Unmodified distribution does not require source code availability
- The vendor can wait for a source code request (without offering it)

The most common mistakes

The most common mistakes

- not even once reading the GPL text and/or the FAQ from the FSF
- not including the GPL license text with the product
- not including a written offer with the product
- not considering that the GPL also applies to software upd ates
- only providing original source code (e.g. vanilla kernel.org kernel)
- not including the "scripts to control installation"
- only providing off-site hyperlinks to license and/ore source code
- not responding to support requests for source code
- charging rediculously high fees for physical shipping of source code

License Compatibility

- There's lots of Free Software available
  - Different Software uses different Licenses:
    - Linux: GPL
    - glibc: LGPL
    - apache: Apache Software License
    - Perl: Artistic
    - ucd-snmp: BSD
  - If you combine (i.e. link) differently-licensed software,
    - check license compatibility
    - in case of doubt, ask legal person and/or contact software authors
    - authors might give you an exception or consider making licenses compatible

Dual Licensing

- The copyright holder (often the original author) can provide alternative licensing
- Some projects do this as a business model (reiserfs, MySQL)
- In some projects it's impossible due to the extremely distributed copyright (e.g. Linux kern el)
- However, in smaller projects it never hurts to ask whether there would be interest in providing an alternative (non-copyleft) licensing

GPL Violations

- When do I violate the license
  - when one ore more of the obligations are not fulfilled
- What risk do I take if I violate the license?
  - the GPL automatically revokes any usage right
  - any copyright holder can obtain a preliminary injunction banning distribution of the infringing product

Past GPL enforcement

Past GPL enforcement

- GPL violations are nothing new, as GPL licensed software is nothing new.
- However, the recent GNU/Linux hype made GPL licensed software used more often
- The FSF enforces GPL violations of code on which they hold the copyright
  - silently, without public notice
  - in lengthy negotiations

The Linksys case

- During 2003 the "Linksys" case drew a lot of att ention
  - Linksys was selling 802.11 WLAN Acces Ponts / Routers
  - Lots of GPL licensed software embedded in the device (included Linux, uClibc, busybox, iptables, ...)
  - FSF led alliance took the usual "quiet" approach
  - Linksys bought itself a lot of time
  - Some source code was released two months later
  - About four months later, full GPL compliance was achiev ed

The Linksys case

- Some developers didn't agree with this approa ch
  - not enough publicity
  - violators don't loose anything by first not complying and wait for the FSF
  - four months delay is too much for low product lifecycles in WLAN world
- The netfilter/iptables project started to do their own enforcement in more cases that were coming up

Enforcement case timeline

- In chronological order
  - some user sends us a note he found our code somewhere
  - reverse engineering of firmware images
  - test purchase to verify device ships gpl-incompliant
  - sending the infringing organization a warning notice
  - wait for them to sign a statement to cease and desist
  - if no statement is signed
    - contract technical expert to do a study
    - apply for a preliminary injunction
  - if statement was signed
    - try to work out the details
    - grace period for boxes in stock possible
    - try to indicate that a donation would be good PR

Sucess so far

- Success so far
  - amicable agreements with a number (35+) of companies
    - some of which made significant donations to charitable organizations of the free software community
  - preliminary injunction against Sitecom, Sitecom also lost appeals case
  - court decision of munich district court in Sitecom appealscase

  - three more preliminary injunctions
  - more settled cases (not public yet)
  - negotiating in more cases
  - public awareness

Cases so far (1/3)

- Allnet GmbH
- Siemens AG
- Fujitsu-Siemens Computers GmbH
- Axis A.B.
- Securepoint GmbH
- U.S.Robotics Germany GmbH
- Netgear GmbH
- Belkin Compnents GmbH
- Asus GmbH
- Gateprotect GmbH
- Sitecom GmbH / B.V.
- TomTom B.V.
- Gigabyte Technologies GmbH

Cases so far (2/3)

- Sun Deutschland GmbH
- Open-E GmbH
- Siemens AG (second case)
- Deutsche Telekom AG
- Hitachi Inc.
- Tecom Inc.
- ARP Datacon GmbH
- Conceptronic B.V.
- D-Link GmbH
- Adaptec Deutschland GmbH
- Belkin Compnents GmbH (second case)

Cases so far (3/3)

- Siemens AG (third case)
- TARGA GmbH
- Medion AG
- naviflash GmbH
- Maxtor Inc.
- Cisco Deutschland GmbH
- Fortinet
- naviflash GmbH
- iRiver Europe GmbH
- Cisco Deutschland GmbH (second case)
- Acer Deutschland GmbH
- SMC Networks GmbH

- >= 100 more not public

What we've learned

- Copyleft-style licenses can be enforced!
- A lot of companies don't take Free Software licenses seriously
  - Even corporations with large legal departments who should know
  - Reasons unclear, probably the financial risk of infringement was considered less than the expected gains
- The FUD spread about "GPL not holding up in court" has disappeared

Future GPL Enforcement

- GPL Enforcement
  - remains an important issue for Free Software
  - will start to happen within the court more often
  - has to be made public in order to raise awareness
  - will probably happen within some form of organization
  - talks have started with the FSF Europe
- What about Copylefted Content (Creative Commons)
  - probably just a matter of time until CC-licensed works of art are infringed

Problems of GPL Enforcement

- Problems
  - distributed copyright
    - is an important safeguard
    - can make enforcement difficult, since copyright traditionally doesn't know cases with thousands of copyright holders
    - distribution of damages extremely difficult
  - the legal issue of having to do reverse engineering in order to prove copyright infringement(!)
  - only the copyright holder (in most cases the author) can do it
  - users discovering GPL'd software need to communicate those issues to all entitled parties (copyright holders)
  - infringers obfuscating and/or encrypting fres software as disguis

gpl-violations.org

- The http://www.gpl-violations.org/ project was started ~ 3 year ago
  - as a platform wher users can report alleged violations
  - to verify those violations and inform all copyright holders
  - to inform the public about ongoing enforcement efforts
-
- At the moment, project is only backed by the au thor
  - more volunteers needed to investigate all cases
  - something like 270 reported (alleged) violations backlog

Make later enforcement easy

- Practical rules for proof by reverse engineering
  - Don't fix typos in error messages and symbol names
  - Leave obscure error messages like 'Rusty needs more caf feine'
  - Make binary contain string of copyright message, not only source
- Practical rules for potential damages claims
  - Use revision control system
  - Document source of each copyrightable contribution
    - Name+Email address in CVS commit message
  - Consider something like FSFE FLA (Fiduciary License Ag reement)
  - Make sure that employers are fine with contributions of their employees
- If you find out about violation
  - Don't make it public (has to be new/urgent for injunctive relief)

Finding GPL violations

Finding GPL violations in embedded systems

- Software based reverse engineering techniques
  - Analyze firmware update files
  - Exploit security bugs in firmware
  - Use existing "backdoors"
- Hardware based reverse engineering techniques
  - Take actual board apart, note major components
  - Find + use JTAG testpads
  - Find + use serial console

Finding GPL violations

Software based reverse enigneering

- Analyzing firmware update files
  - Typically very easy, you don't even need the device to start with
  - Firmware updates available on the internet
  - Start with hex-editor, 'strings' and some pattern-matchi ng
  - Find magic numbers of kernel image, cramfs/jffs2 rootfs i mage
  - extract kernel and/or rootfs by very easy means (dd, gzip/bzip2, ...)

Finding GPL violations

- Software based reverse enigneering
  - Exploit security bugs in firmware
    - Typical embedded system vendor has
      - zero clue about Linux
      - zero clue about Security
      - ages old software versions
    - Therefore, very easy to exploit years-old security proble ms
      - buffer overflows, stack smashing
      - HTTP UI scripting problems (quoting problems)
    - Goal
      - to get a [root] shell on the affected device

Finding GPL violations

Software based reverse enigneering

- Use existing "backdoors"
  - Sometimes, devices have existing backdoors from R&D
  - e.g. special undocumented command in telnet or SSH UI for getting a shell
  - very device-specific, no generic rule how to find it

Hardware based reverse engineering

Hardware based reverse engineering

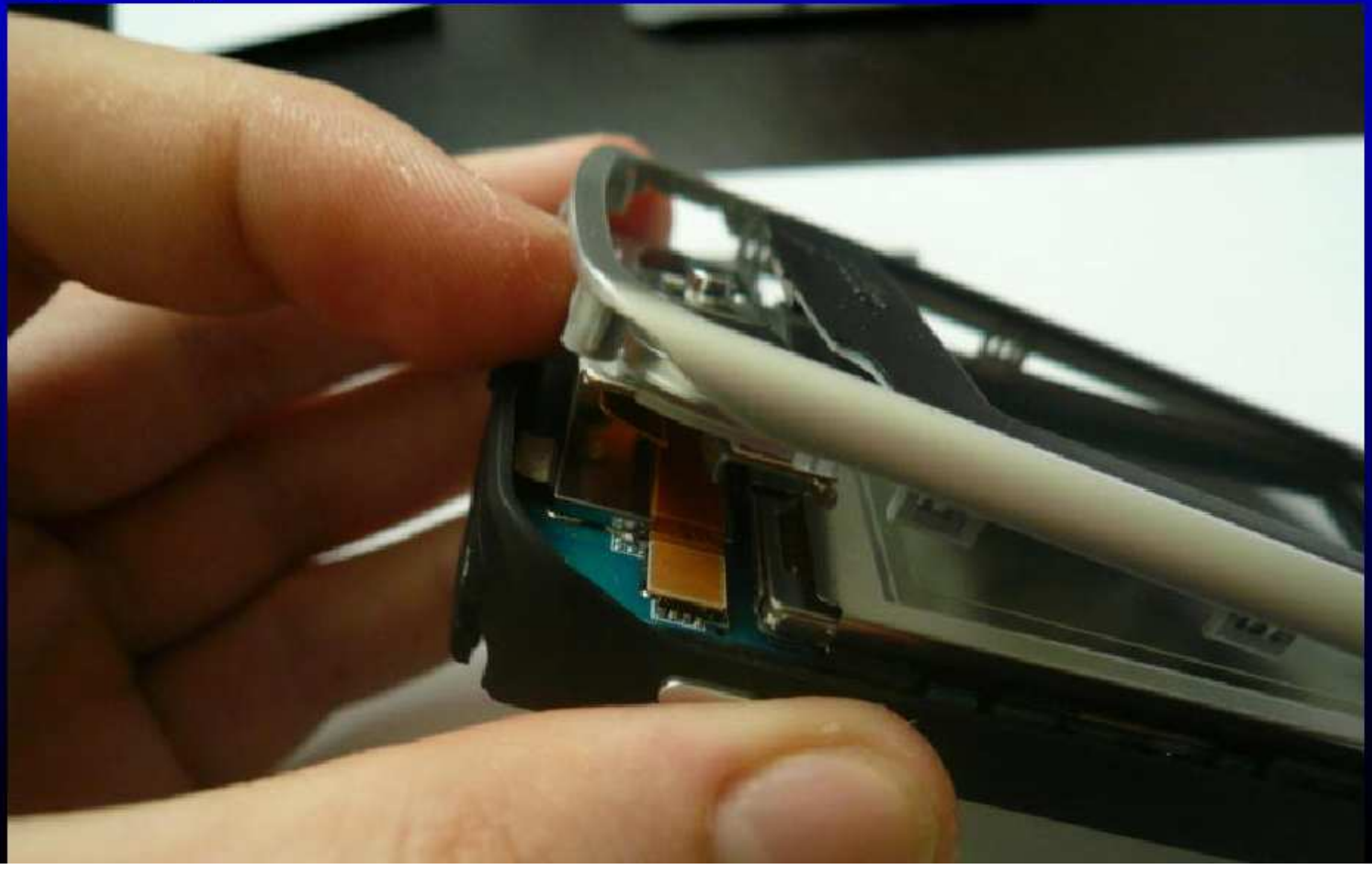
- almost always successful
- amount of work differs from device to device
- see the next slides for an example
- BTW: the device in question is NOT a GPL viola tion

Take hardware apart

Opening the case and void your warranty

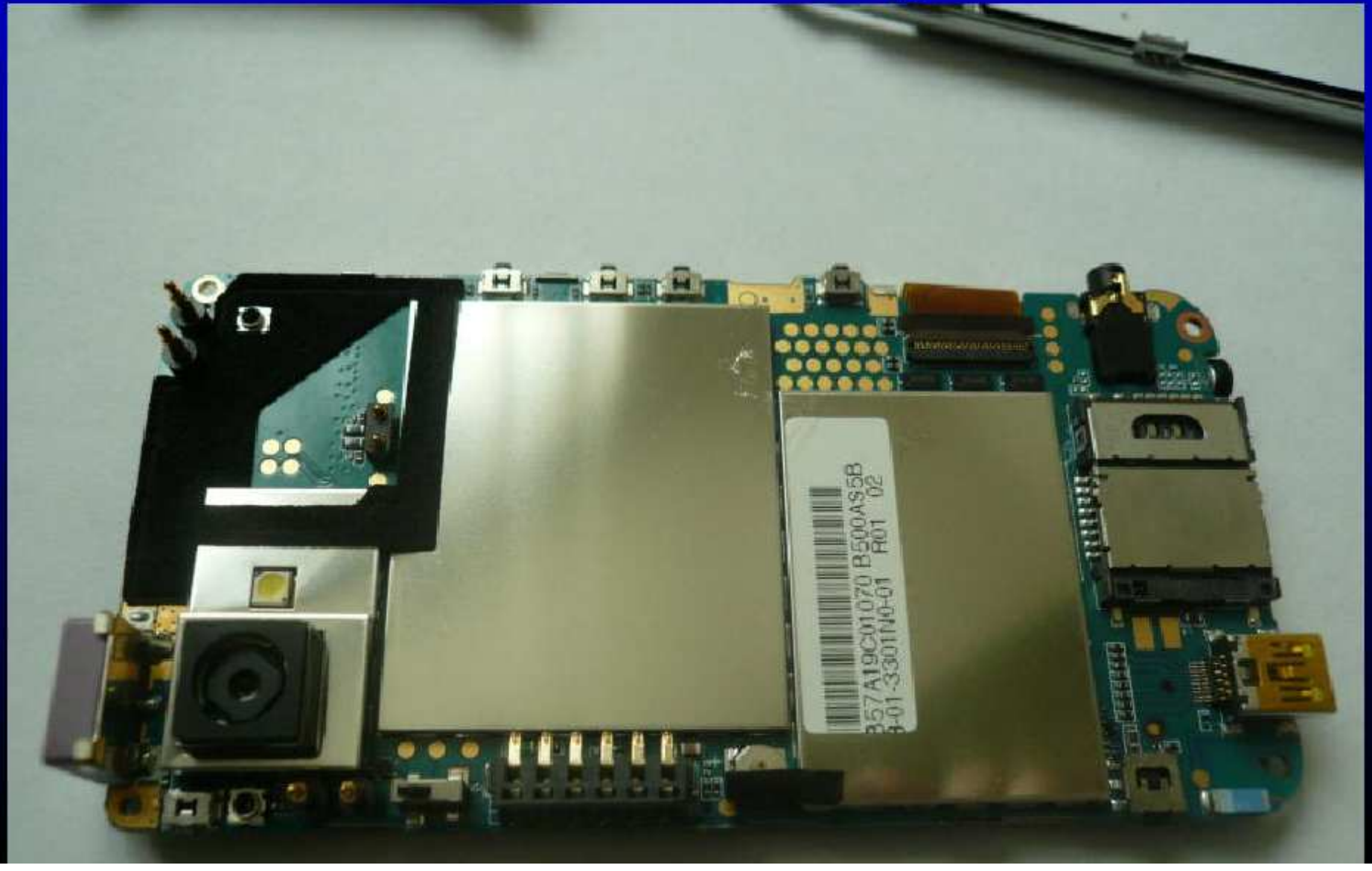
Take hardware apart

Opening the case



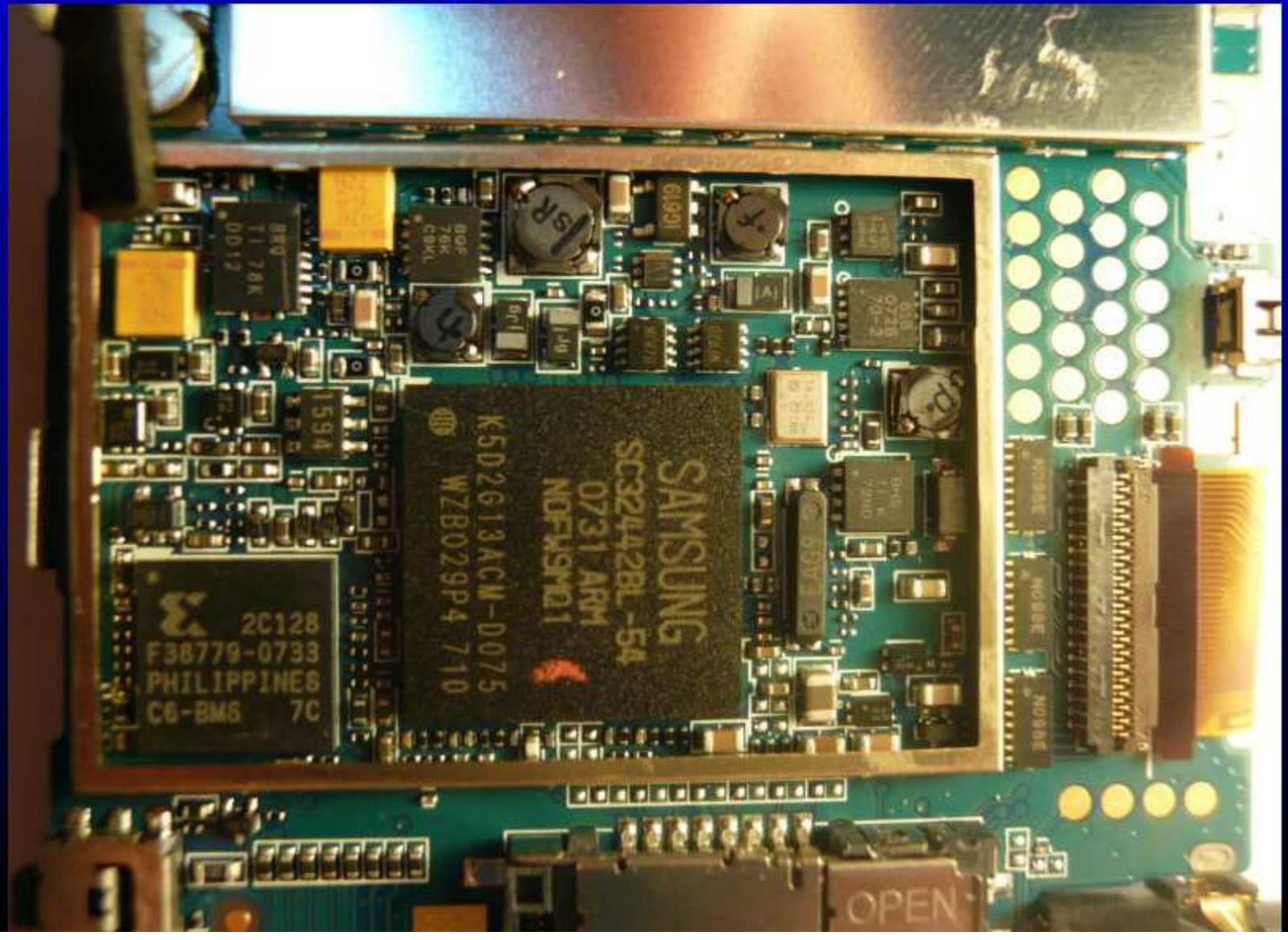
Take hardware apart

The Mainboard with all its shielding covers



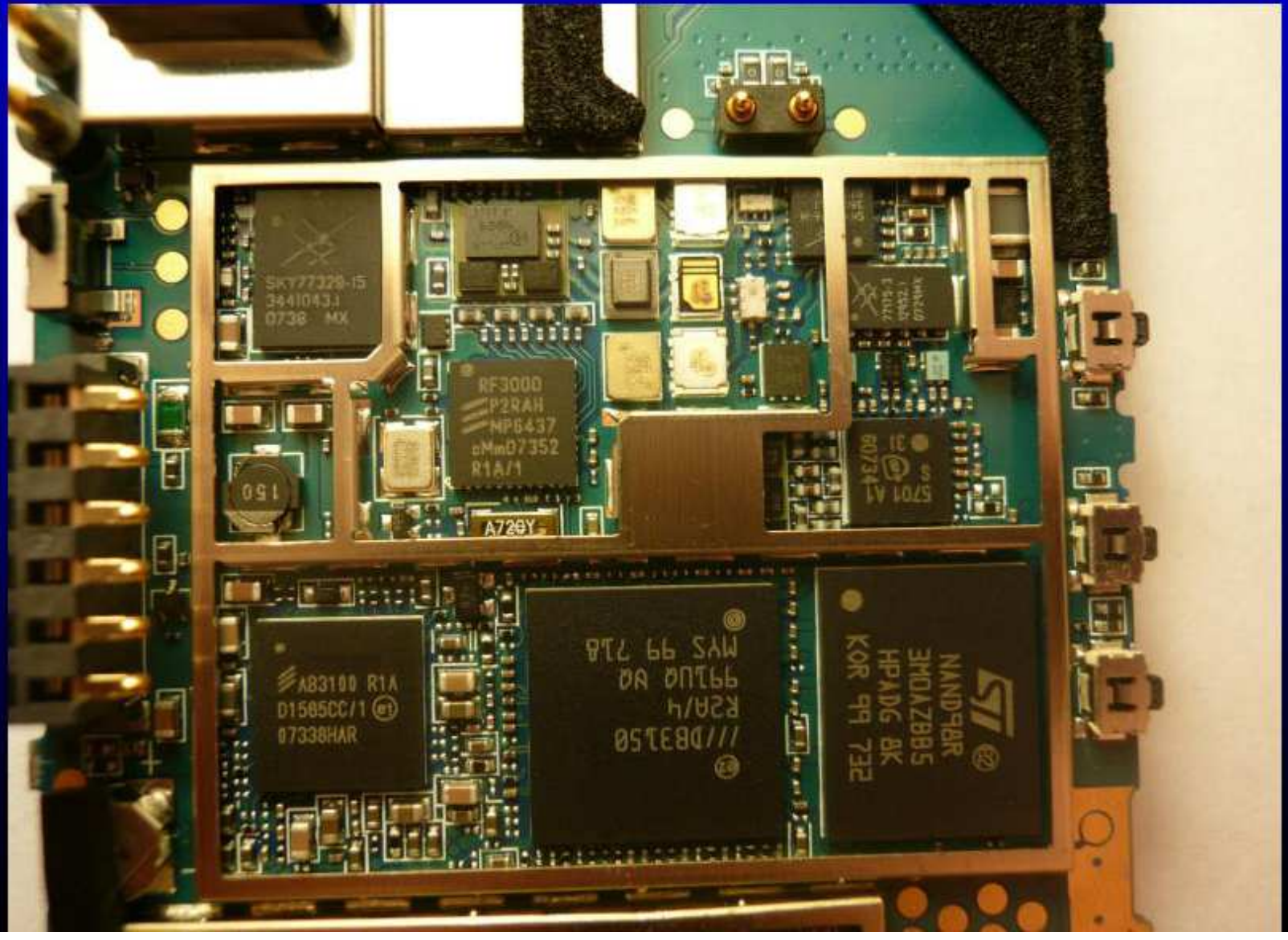
Take hardware apart

The application processor section



Take hardware apart

The HSDPA modem section



Take hardware apart

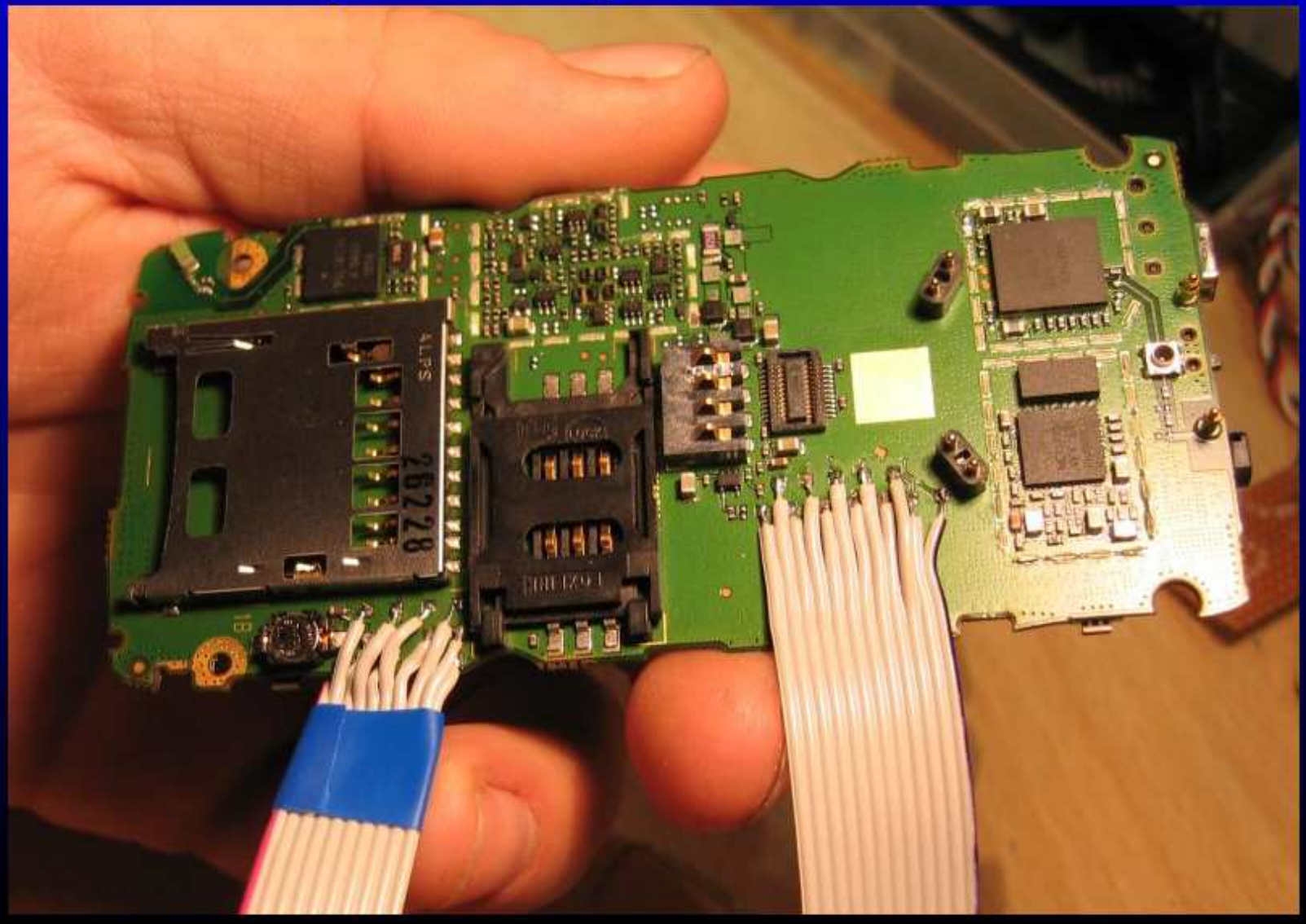
The backside

JTAG pins

- JTAG is a very useful interface
  - boundary scan (EXTEST + INTEST)
  - ARM Integrated Debug Macrocell
- Find + use JTAG testpads
  - look for suspicious testpads on PCB
  - tracing PCB traces impossible at 8-layer PCB
  - trial + error
  - sometimes you might find schematics ;)

JTAG pins

Find + use JTAG testpads

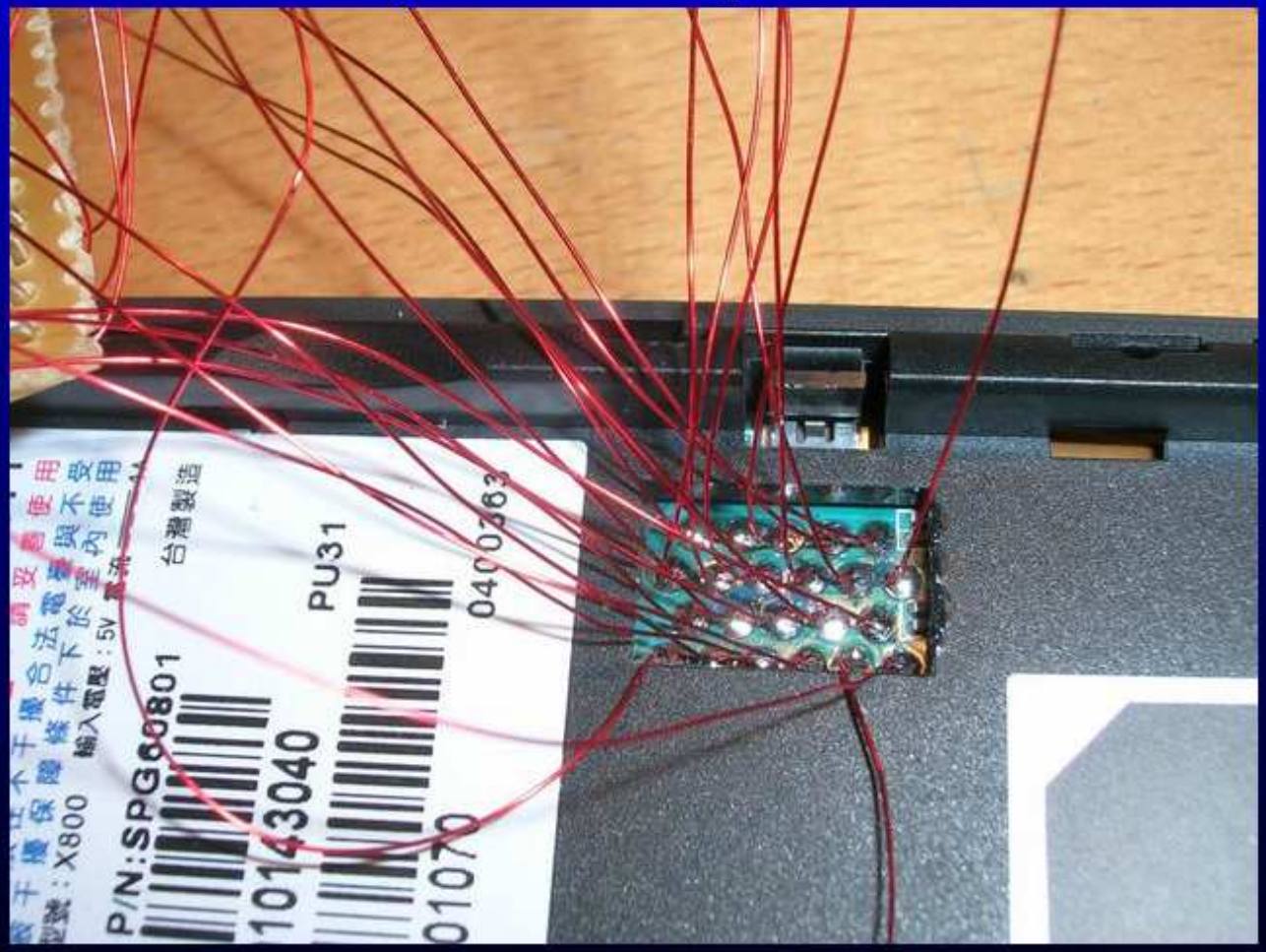


JTAG pins

- Find + use JTAG testpads
  - JTAG is basically a long shift register
  - Input, Output, Clock (TDI, TDO, TCK)
  - Therefore, you can try to shift data in and check if/where it comes out
  - Automatized JTAG search by project "jtagfinder" by Hunz (German CCC member)

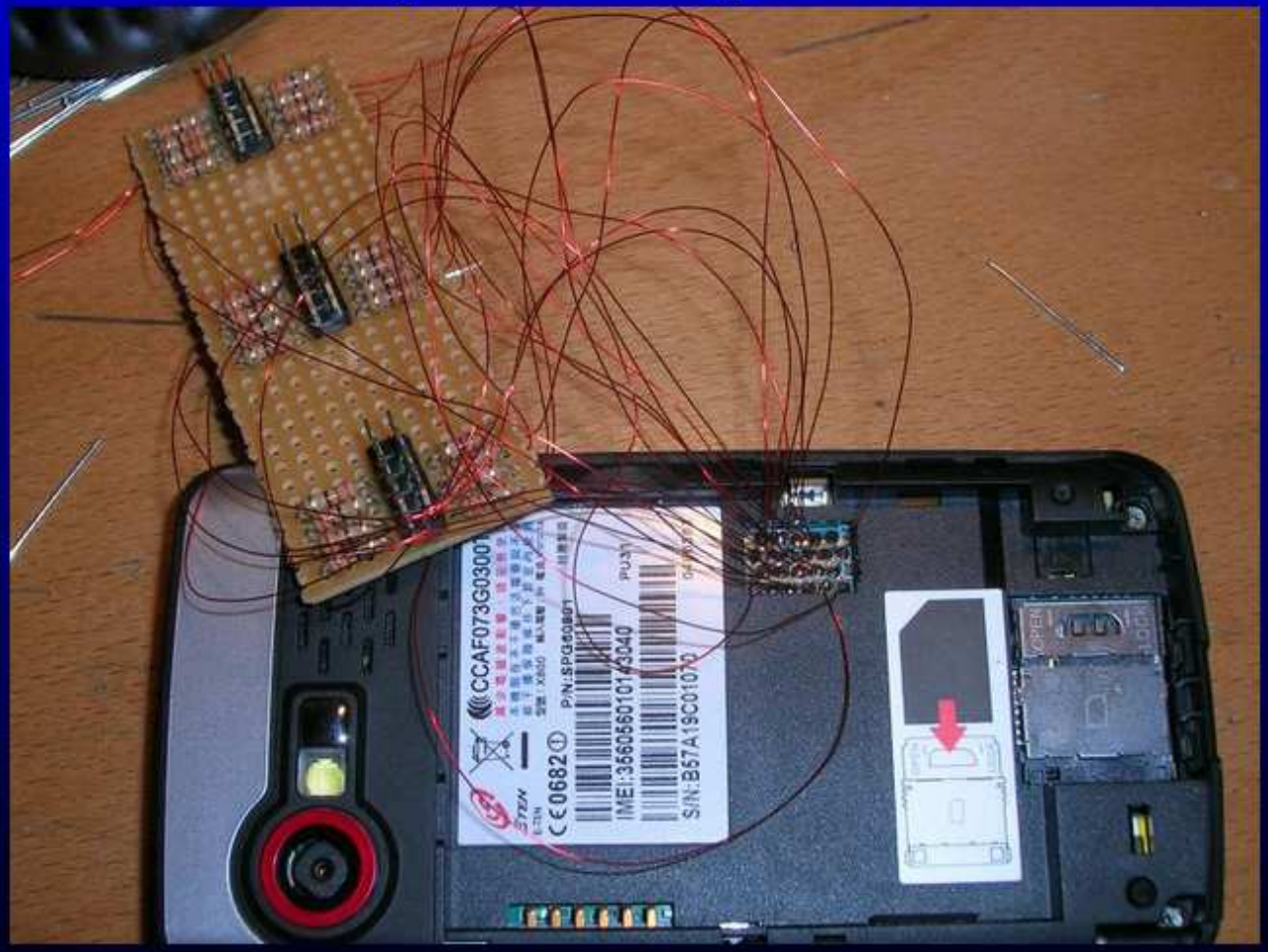
JTAG pins

Find + use JTAG testpads



JTAG pins

Find + use JTAG testpads



JTAG pins

Find + use JTAG testpads

JTAG pins

Find + use JTAG testpads

JTAG pins

Found JTAG pins

- Chain 1
  - Samsung S3C2442 Application Processor
  - Has standard ARM JTAG ICE
- Chain 2
  - CPLD programming interface
- Remaining work
  - find the nTRST and nSRST pins

Serial console

How to find the serial console

- Just run some code that you think writes to it
- Use a Scope to find typical patterns of a serial p ort
- I haven't actually done (or needed) this on the glofiish yet, but on many other devices
- RxD pin is harder to find, just trial+error usually works as soon as you have some interactive prompt that echo's the characters you write
- Don't forget to add level shifter from 3.3/5V to RS232 levels

The End

- Further reading:
- The http://gpl-violations.org/ project
- The Free Software Foundation http://www.fsf.org/, http://www.fsf-europe.org/
- The GNU Project http://www.gnu.org/
- The netfilter homepage http://www.netfilter.org/