Free Software on GSM Phones

December 29, 2005
22C3

by

Harald Welte <laforge@gnumonks.org>

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 can claim to be the first to have enforced the GNU GPL in court
- who is doing way too many projects simultaneously, one of them OpenEZX

Contents

Disclaimer

Disclaimer

- I have no affiliation with Motorola
- OpenEZX project has no affiliation with Motorola
- All Information is based on observation, and may be wrong
- Lots of the work has been done by a large community, I’m a newbie ;)

What is OpenEZX

- OpenEZX project
  - to document EZX phone hardware and software
  - to provide 100% free software stack for frontend CPU
  - might at some future point in time also look into GSM/RF related hacks
  - Homepage: http://openezx.org/ (http://open-ezx.org)
  - Wiki: http://wiki.openezx.org/

History

History of Motorola Linux based gsm phones

- A760, A768
  - Released in Asia in 2003
- EZX (A780, E680, E680i)
  - E680 sold only in asian market
  - A780 sold in China since August 2004
  - A780 first Motorola Linux phone available in EU/US

OpenEZX

A780

- The A780 phone
  - Quad-band GSM
  - AGPS
  - GPRS, EDGE, HSCSD
  - Intel Xscale based
  - Monta Vista CE Linux
  - Bluetooth
  - USB device port (modem / mass storage)
  - Transflash slot (SD-card in smaller form factor)

E680/E680i

- The E680 phone
  - Like A780
  - No GPS
  - full-size SD/MMC slot
  - FM Radio
  - minor differences in Audio system, GPIO assignment, ...
- The E680i phone
  - seems to only differ in software

Techniques for re-engineering

- learn about the device
  - take the device apart
  - take high-res PCB photographs
  - FCC database sometimes quite helpful
  - remove all the shielding covers
  - write down types of all integrated circuits
  - google for those circuits, try locating data sheets
  - sometimes service manuals can be obtained for small fees

Techniques for re-engineering

- try to find a serial console port
  - successful in many embedded devices
  - all you need is a 3.3v<->RS232 level shifter
  - A780: checking all 100+ test points with an oscilloscope :(
  - unfortunately not successful in the case of A780
- try to find a JTAG port
  - cheap JTAG / parallel port adaptors available or DYI
  - only helps if you also have a BSDL file or similar
  - hard to figure out which of the five pins is which
  - be aware: there might be multiple JTAG ports for multiple IC’s

Techniques for re-engineering

- access to the OS instead of the UI
  - serial console helps in many cases, not in this one
  - networked devices sometimes have telnet/ssh available
  - exploits of known-to-be-installed software (zlib-1.1.3)
  - try "weird button combinations" at startup
- access to flash memory
  - read out via JTAG
  - if you have shell access, dd if=/dev/mtd* of=...
  - via vendor-supplied flash programming tool
  - copy / unpack / mount flash image to PC workstation

Techniques for re-engineering

- simulation
  - running ARM binaries from device in QEMU emulation
  - commercial ARM emulators
- disassembling
  - WARNING: may be illegal in most jurisdictions
  - use gnu binutils (objdump, ...)
  - use special-purpose proprietary tools (IDA Pro)

A780 Hardware

- In short
  - A Motorola Neptune LTE based mobile phone plus
  - A PXA270 Xscale based PDA in one case
- Application Processor (PXA270)
  - runs heavily modified linux-2.4.20 kernel
  - 48MB RAM
  - 48MB "wireless" flash
  - software-configurable clock speed up to 400MHz
  - JTAG port on test pads, BSDL file and JFlash available
  - SPI/SSP interface to PCAP and BP
  - directly attached to 320x200 LCD display
  - directly attached to touch screen, buttons
  - directly attached to 1.3Mpixel camera module

A780 Hardware

- Baseband Processor (Neptune LTE)
  - contains ARM7TDMI for GSM stack
  - contains 566xx DSP for digital baseband
  - JTAG port on test pads, but no BSDL file
  - Connected to Application processor via USB
  - SPI/SSP interface to PCAP and AP
  - UART connected to AGPS processor
  - Connects to GSM SIM module
  - 8MB external flash
  - 2MB external RAM

A780 Hardware

- AGPS Processor (Motorola Telematics MG4100)
  - Attached to UART of BP
  - Has it’s own Flash and RAM (2MB?)
- PCAP2 (power management, clock and audio peripheral)
  - produces a 16 different voltages
  - handles all mono/stereo audio
  - connected to 2 speakers, microphone, vibrator
  - clock generation
  - SPI/SSP interface to AP and BP
  - Backlight control

A780 Hardware

RF Part (not very much information known)

- RF6003
  - fractional-n RF synthesizer
- RF2722
  - GPRS/EDGE capable receiver (RX)
- RF3144
  - quad-band power amplifier (TX))))

A780 AP Software

- linux-2.4.20
  - whole bunch of montavista additions
  - dynamic power management
  - EZX arm subarchitecture
  - low-level drivers for
    - SPI/SSP
    - PCAP Audio (mono/stereo/headset/...)
    - Vibrator (/dev/vibrator)
    - USB host port attached to BP
    - USB device port (belcarra usbd, not gadget)
    - Transflash/SD/MMC
  - THREE proprietary flash file systems
    - Intel VFM (hatcreek.o)
    - m-systems DiskOnChip (tffs.o)
    - third unknown

A780 AP Software

- mux_cli.o
  - hooks into special functions of USB host driver
  - provides GSM TS07.10 (de)multiplex
  - userspace has tty devices
- gprsv.o
  - implements GPRS line discipline for mux_cli ttys
  - hooks into netfilter to intercept DNS packets ?!?
  - provides gprs0 / grps1 network devices
- ipsec.o
  - proprietary ipsec stack (don’t we already have two GPL licensed?)
  - Copyright Certicom Corp

A780 Software

- Libraries
  - glibc
- Bluetooth
  - proprietary userspace program directly opens HCI
- GPS
  - no NMEA, no serial device emulation :(
  - proprietary library via mux_cli kernel module
- UI
  - embeddedQt
  - Motorola EZX toolkit
- Java
  - Full J2ME support
    - (but who wants java if there's linux?)

A780 Software

- Apps
  - Opera
  - Helix Player with codecs
    - aac, amr, mp4, realvideo, mid, mp3, mp4, wma
  - movianVPN
    - proprietary IPsec VPN client
  - CoPilot
    - proprietary GPS navigation, map&route program

EZX Firmware Images

- EZX Firmware Images
  - Motorola ships .SHX firmware images to service centres
  - No legal way for users to get FW updates
  - Proprietary Windows apps flash phone via USB
    - Motorola PST
    - Motorola RSD lite
  - SHX files contain ’code groups’
    - AP bootloader (blob based)
    - AP linux kernel
    - AP root filesystem
    - AP /ezxlocal filesystem
    - AP "language pack"
    - Bootup Logo/Animation
    - BP OS
    - DSP code
    - Cryptographic Signature(s)

EZX bootloader

- EZX bootloader
  - based on GPL licensed blob
  - source code not yet released by Motorola
  - low-level initialization code (GPIO config, clock, ...)
  - vendor specific USB device that allows for
    - transfer of executable code from USB host
    - execution of transferred executable
  - serial console code is present in binary, but not used :(
  - PST/RSD firmware updates work by uploading a ’ramloader’

EZX USB (EMU)

- EZX phones seem to have USB device port
  - Actually, it’s "Enhanced Mini USB" (EMU)
  - Depending on pullup/pulldown/... resistors
    - USB device port
    - Serial port (RS232 at 3.3V levels)
    - Stereo audio signal
    - 500mA charger
    - Carkit (easy install, professionally installed)
    - Factory test

EZX USB (EMU)

- USB Configurations
  - Even in USB device EMU mode, there are many configs
  - Official configs
    - cdc_acm (serial modem emulation for host pc)
    - USB mass storage (transflash and VFAT-on-TFFS devices)
  - Undocumented configs
    - usbnet (network device over USB)

Allows telnet into phone

    - PST

Mode used by PST Windows App

    - DSPlog

Apparently a way to dump data from DSP

    - NetMonitor

supposedly for GSM network monitor

Status

- Status of OpenEZX
  - fairly good picture about phone architecture
  - initial 2.6.14 port done, still lots of bugs
  - Updated toolchain (gcc-3.4)
  - EZX / OPIE / embeddedQt integration
  - Linux native BlueZ bluetooth working
  - netfilter/iptables port (you can do NAT between GPRS and usbnet)
  - nmap/tcpdump/af_packet.o
  - lsof, busybox, bash2,
  - gameboy emulator
  - qonsole (qt console app with OSD keyboard)

TODO

- TODO
  - get 2.6.x kernel fully running, including all drivers + power management
  - write free software backend to talk to Neptune LTE (tapisrv)
  - reimplement mux_cli and gprsv kernel modules
  - some reference application that can make voice and/or data calls from the commandline
  - USB On-The-GO support (hardware support present!)
  - discover how DSPlog, PST, other interfaces work
  - write linux-based app for phone flashing via USB
  - dm-crypt for your personal contacts/data
  - native IPsec
  - ScummVM port [320x240 and touchpad, ideal!] :)
  - at some point merge with openembedded.org ?

Thanks

- Thanks to
  - the BBS scene, Z-Netz, FIDO, ...
    - for heavily increasing my computer usage in 1992
  - KNF (http://www.franken.de/)
    - for bringing me in touch with the internet as early as 1994
    - for providing a playground for technical people
    - for telling me about the existance of Linux!
  - Alan Cox, Alexey Kuznetsov, David Miller, Andi Kleen
    - for implementing (one of?) the world's best TCP/IP stacks
  - Astaro AG
    - for sponsoring parts of my free software work
  - Chaos Computer Club (http://www.ccc.de/)
    - for providing an inspiring environment for cool hacks
- The slides and the an according paper of this presentation are available at http://svn.gnumonks.org/projects/presentations