



Open Source Programování

<http://rtime.felk.cvut.cz/osp/>

Pavel Píša

<pisa@fel.cvut.cz>

<http://cmp.felk.cvut.cz/~pisa>

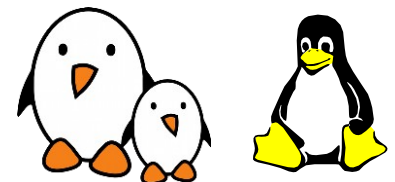
Michal Sojka

František Vacek

DCE FEL ČVUT



© Copyright 2004-2010, Pavel Píša, Michal Sojka, František Vacek,
Free-Electrons.com, GNU.org, kernel.org,
Wikipedia.org, distrowatch.com
Creative Commons BY-SA 3.0 license Latest update: 17. V 2011



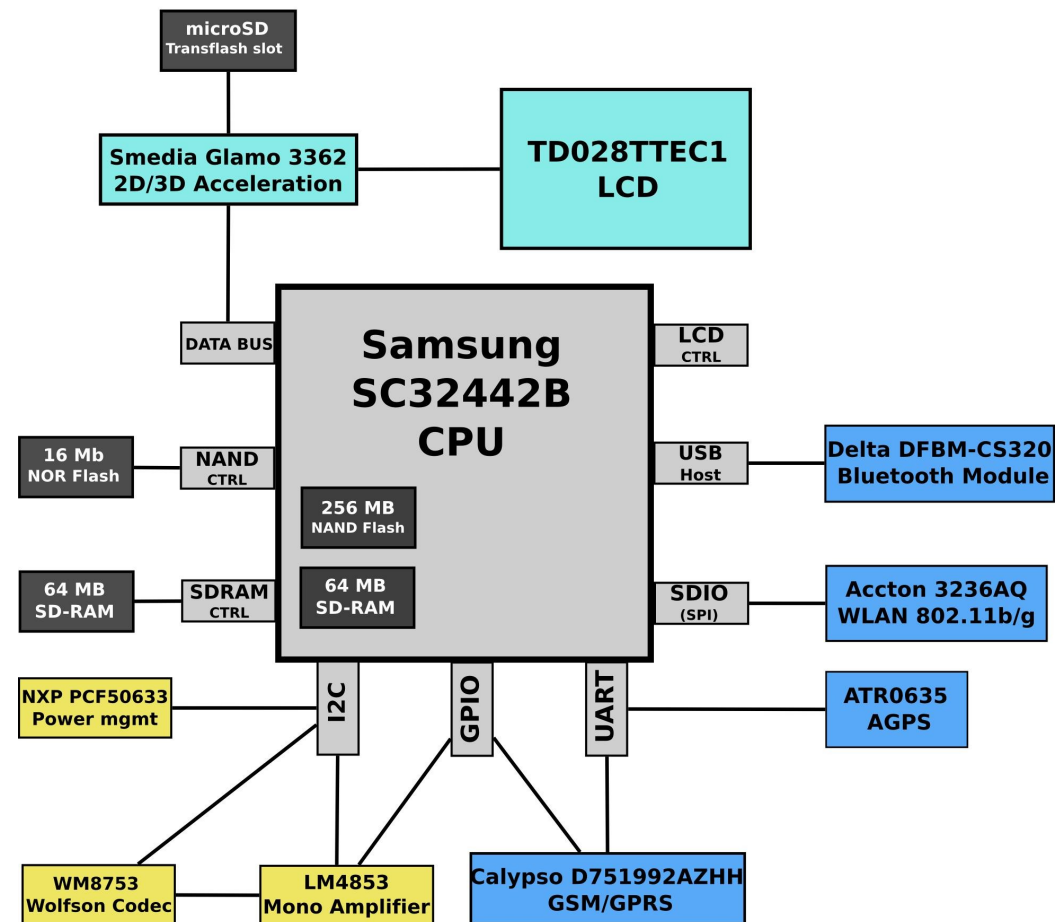
- ▶ High resolution touch screen 2.84" (43mm x 58mm) 480x640 pixels
- ▶ 128MB SDRAM memory
- ▶ 256 MB integrated flash memory (expandable with microSD or microSDHC card)
- ▶ microSD slot supporting up to 16GB SDHC (Secure Digital High Capacity) cards (Supported microSD cards)
- ▶ Internal GPS module
- ▶ Bluetooth
- ▶ 802.11 b/g WiFi
- ▶ 400Mhz ARM processor
- ▶ 2 3D accelerometers
- ▶ 2 LEDs illuminating the two buttons on the rim of the case (one bicolor [blue|orange] behind the power button, 1 unicolor [red] behind the aux button)
- ▶ Tri-band GSM and GPRS
- ▶ USB Host function with 500mA power, allowing you to power USB devices for short periods (will drain the FreeRunner battery faster)



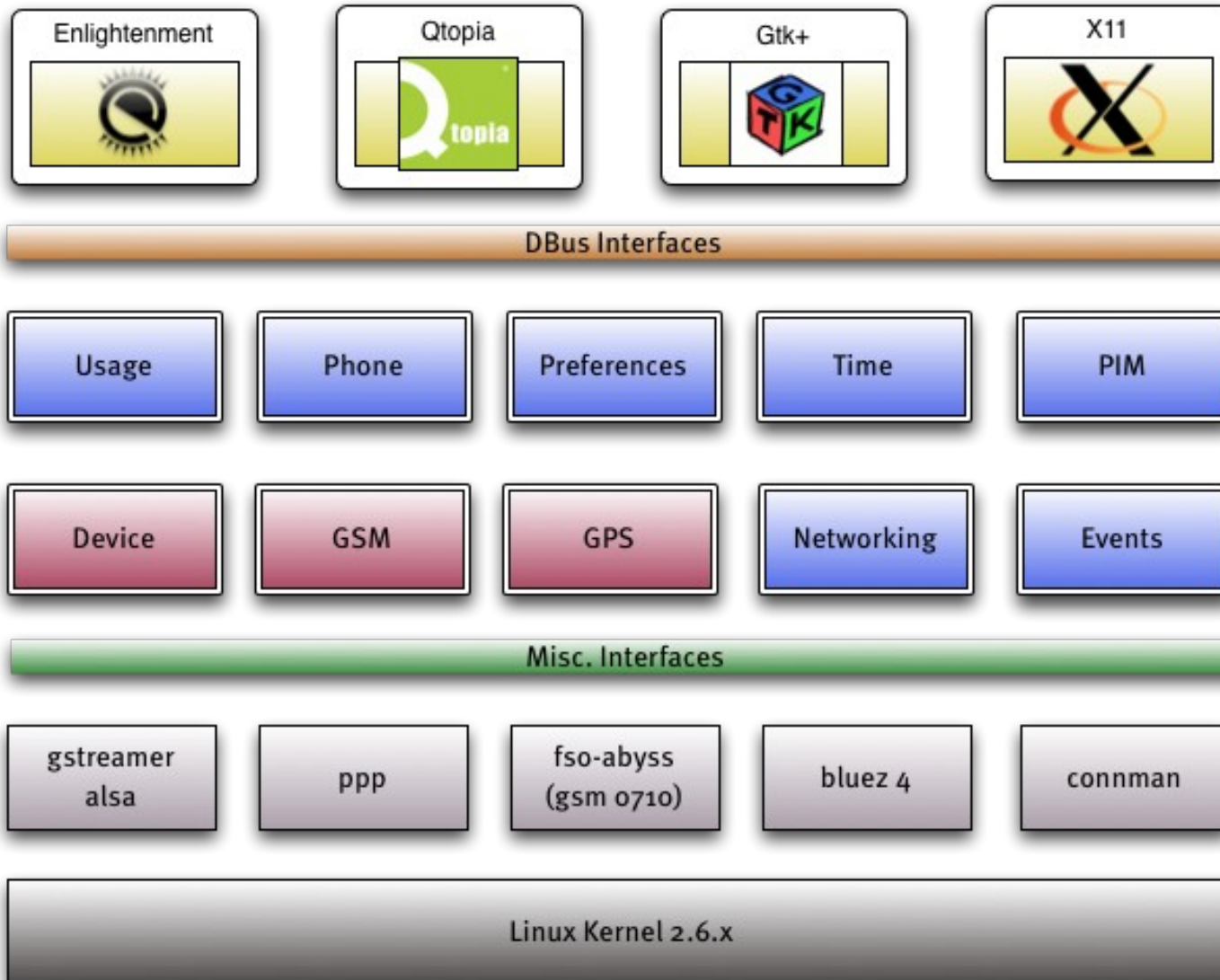
- ▶ Plně otevřený HW
Creative Commons
Attribution Share-Alike
- ▶ Data včetně návrhu
mechaniky
- ▶ Množství distribucí
nad Qt (X11 nebo
FB), Gtk, EFL
(Enlightenment
Foundation Libraries)

Neo FreeRunner (GTA02) Simplified hardware component diagram

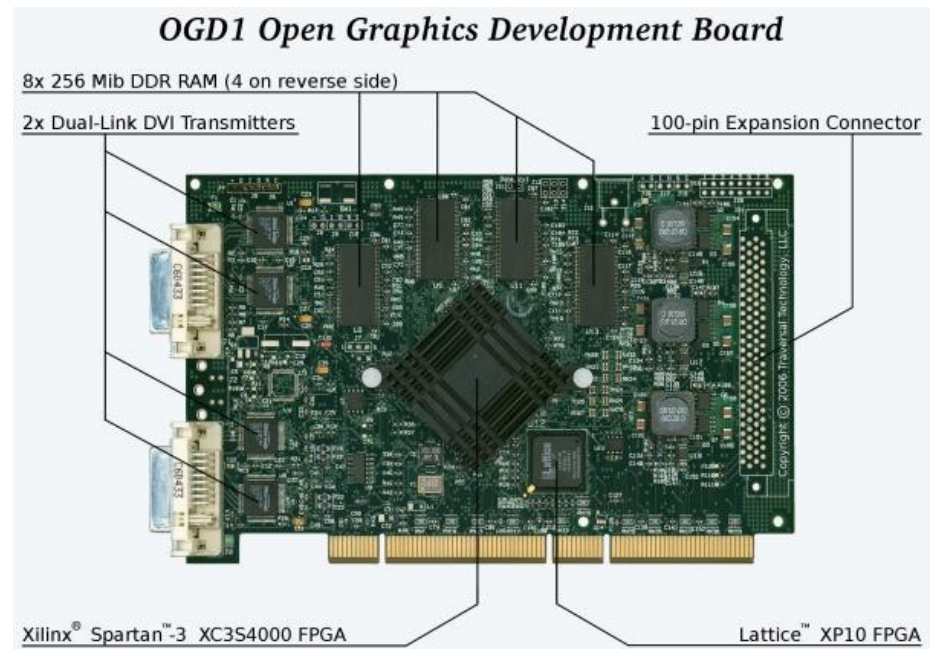
2008 Kim Hauritz, some rights reserved - CC: A-NC-SA



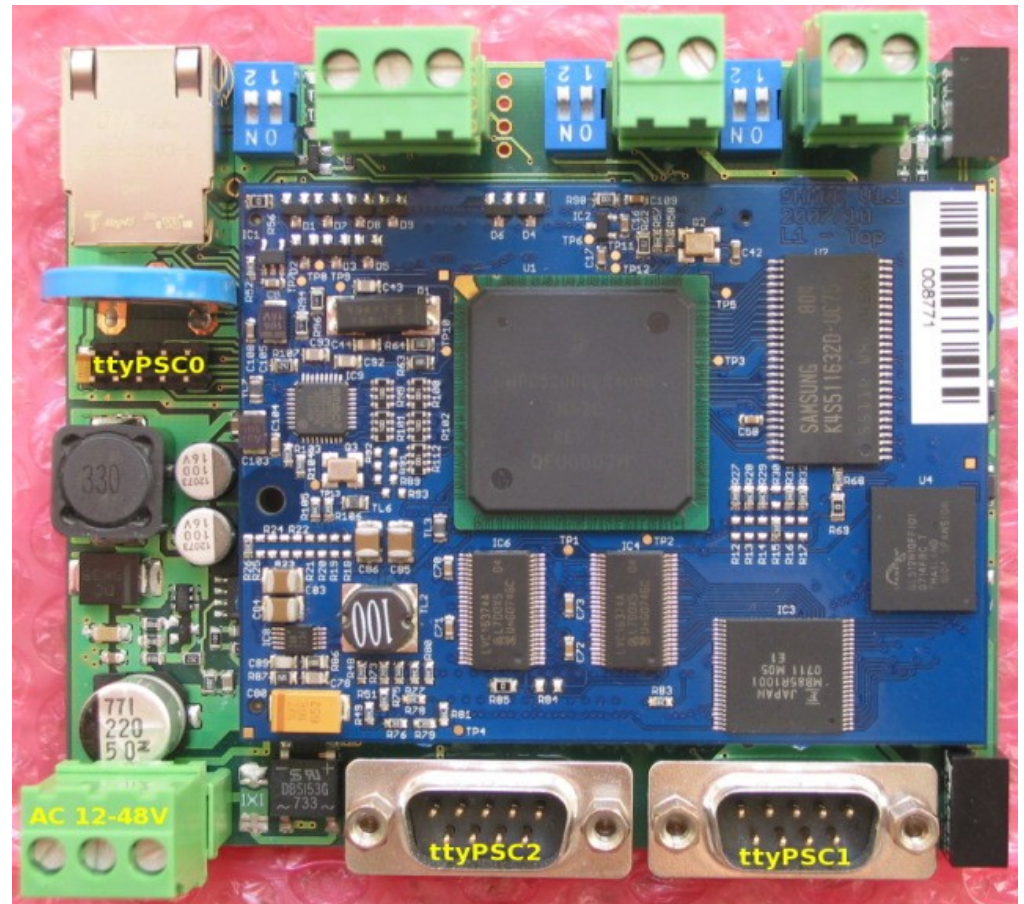
FREESMARTPHONE.ORG ARCHITECTURE

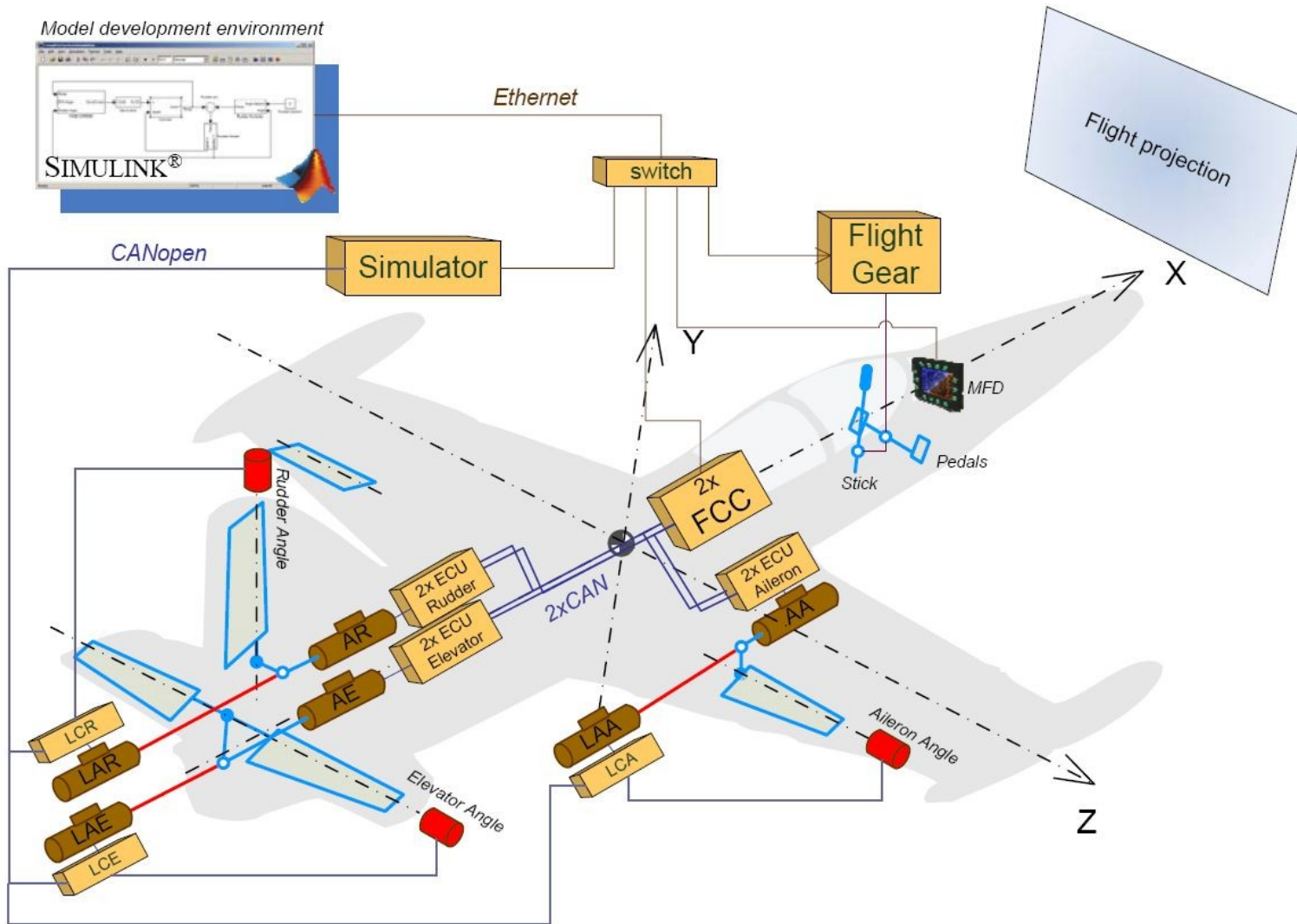


- ▶ První návrh je prototypová deska pro PCI sběrnici. Podporuje dva DVI porty a TV výstup
- ▶ Algoritmy pro zobrazování a akceleraci grafiky jsou realizované pomocí FPGA
- ▶ Pro start OS i podpora VGA kompatibilního režimu - dosaženo již bootování běžné Gentoo distribuce
- ▶ Vlastní návrh HW - GPL licence
- ▶ Device Drivers (MIT/X11 licence)
- ▶ RTL FPGA návrh – Verilog – dvojí licence GPL a proprietární
- ▶ RTL pro PCI bridge ASIC bude uvolněno pod GPL po zaplacení ceny návrhu chipu
- ▶ Stránky projektu: <http://www.opengraphics.org/>
- ▶ Projekt vznikl především jako reakce na problémy s uzavřeností ovladačů a specifikací běžných akcelerovaných grafických karet. Může však být zajímavý i pro projekty akcelerovaného zpracování různých jiných dat.



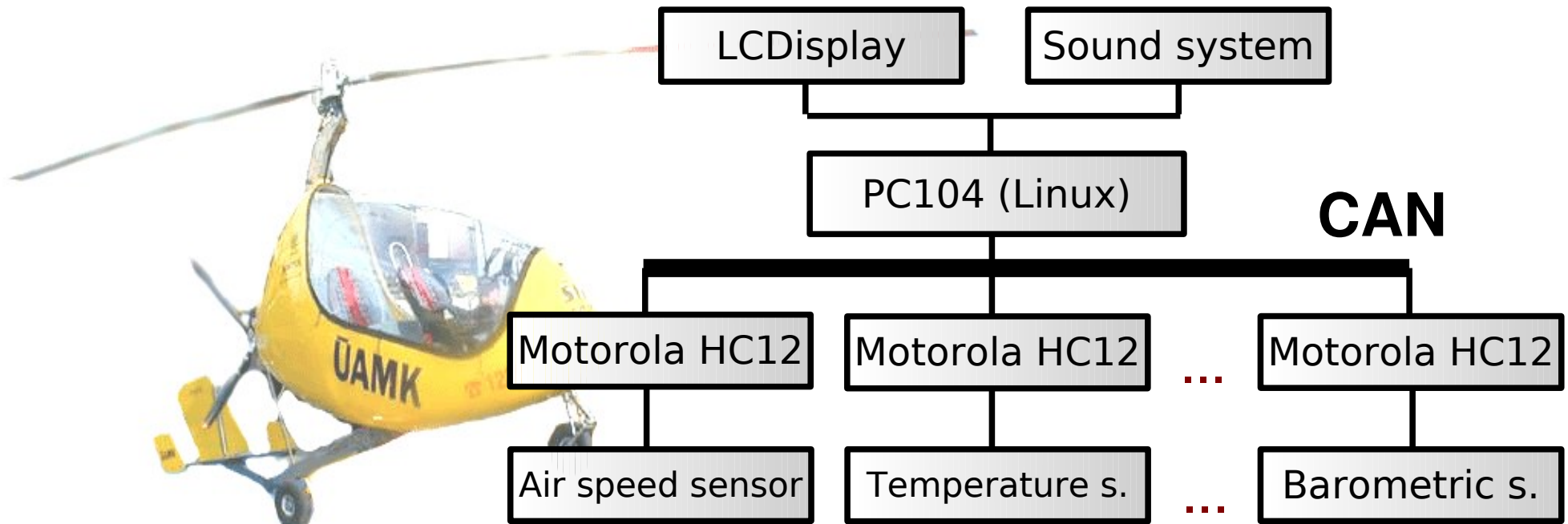
- ▶ Využívá několik projektů na Katedře řídicí techniky (DCE FEL)
 - ▶ Eurobot Flamingos DCE team (Linux + CAN + vlastní sysless HW)
 - ▶ RAMA UAV Control System Homepage – letový SW pro helikoptéru (Linux + CAN + vlastní sysless HW)
- ▶ PowerPC 400 MHz MPC603e FPU, MMU, 16K I/16K D Caches, 10/100 Ethernet MAC, ATA/IDE, USB 1.1 Host, 6x UART, SPI, 2x I2C, 3x I2S, 2x CAN, GPIO Timers, 1.5V core, 3.3V external
- ▶ 128 MB DDR RAM
- ▶ 64 MB NOR Flash
- ▶ Operační systémy
 - ▶ Linux 2.6.x
 - ▶ RTEMS (GPL + linking exception)
 - ▶ VxWorks (uzavřený)





- ▶ BOA 5200 – základ MPC 5200
- ▶ Matlab Simulink Real Time Workshop pro modelování zátěže a řízení
- ▶ Cílová platforma Linux + CANfestival + HW FreeScale DSP 56k

- ▶ OCERA components
LinCAN – CAN driver
- ▶ Software Environment
Linux 2.4.26, SVGAlib, OpenAL, Tiff, ALSA



▶ **Bezpilotní vrtulník RAMA**

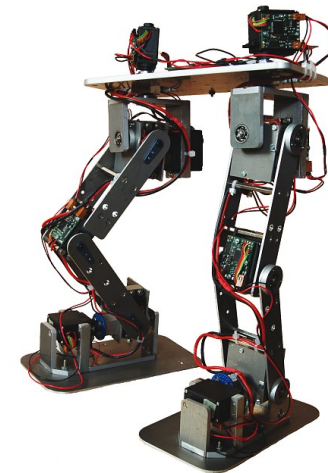
3 palubní počítače
Programováno v C
OS Linux

<http://rtime.felk.cvut.cz/helicopter>



▶ **Kráčející robot SPEJBL**

13 (!) palubních počítačů
Programováno v C
OS Linux



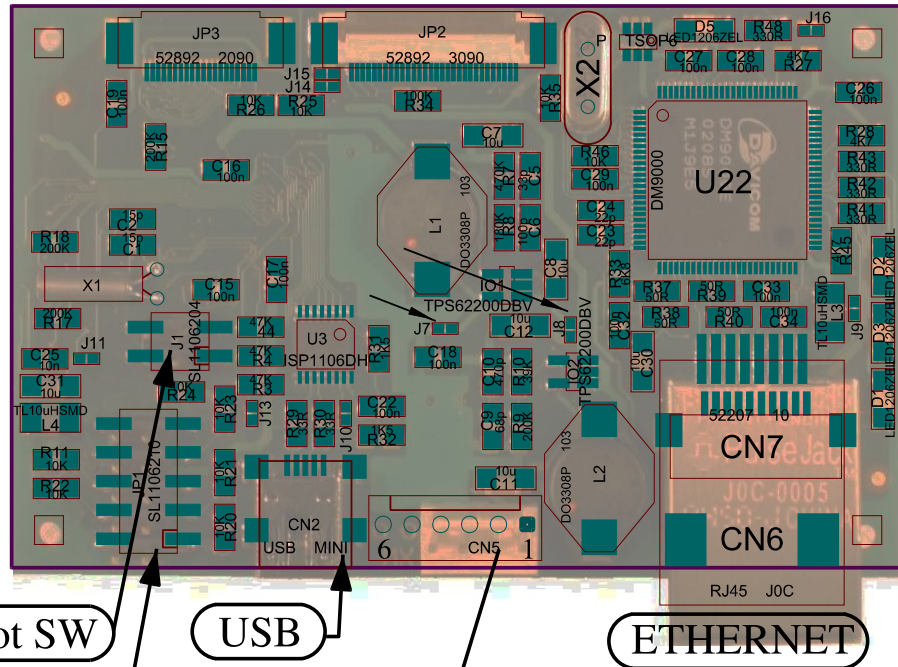
▶ **Mobilní robot ReToBot**

3 palubních počítače
Programováno v C
OS Linux

<http://dce.felk.cvut.cz/retobot>



CMOS Video LCD+KBD

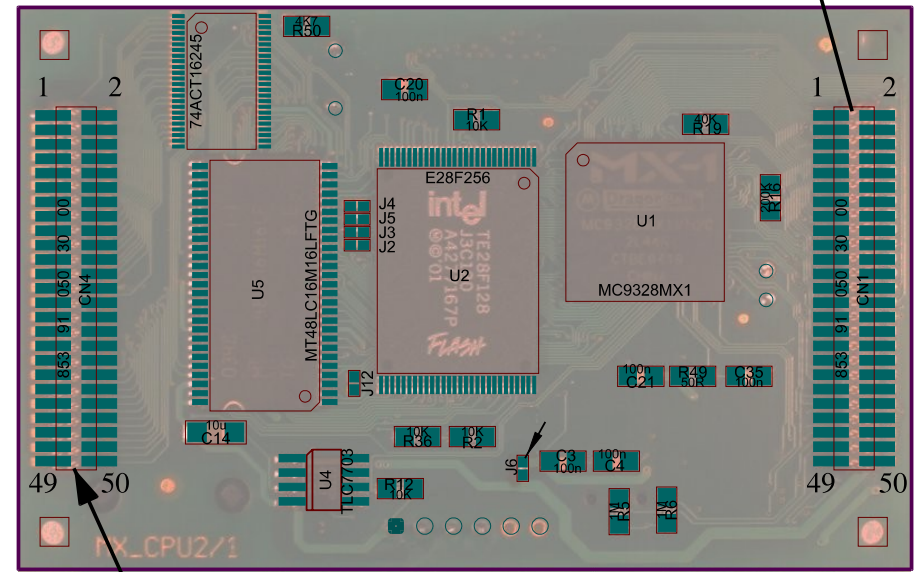


Boot SW JTAG USB UART1 ETHERNET

- ▶ Power supply +5 VDC from USB or +5VDC or 3.1-3.3 VDC external
- ▶ Onboard 1.8 and 3.1 VDC switched regulators
- ▶ Dimensions 50 mm x 79 mm
- ▶ SW RTEMS, U-Boot, Linux, sysess

- ▶ MCU ARM M9328 MX1 ARM920T
- ▶ 32 MB 100 MHz SDRAM (x16 only)
- ▶ 4 MB to 32 MB StrataFlash
- ▶ 3x UART (MX1) 3.1 Volts levels
- ▶ Direct LCD and Keyboard
- ▶ CMOS Video Input
- ▶ 2x 50-pin PRECI-DIP Ultraminiature

IO Extension



BUS Extension

http://www.pikron.com/pages/products/cpu_boards/pimx1.html

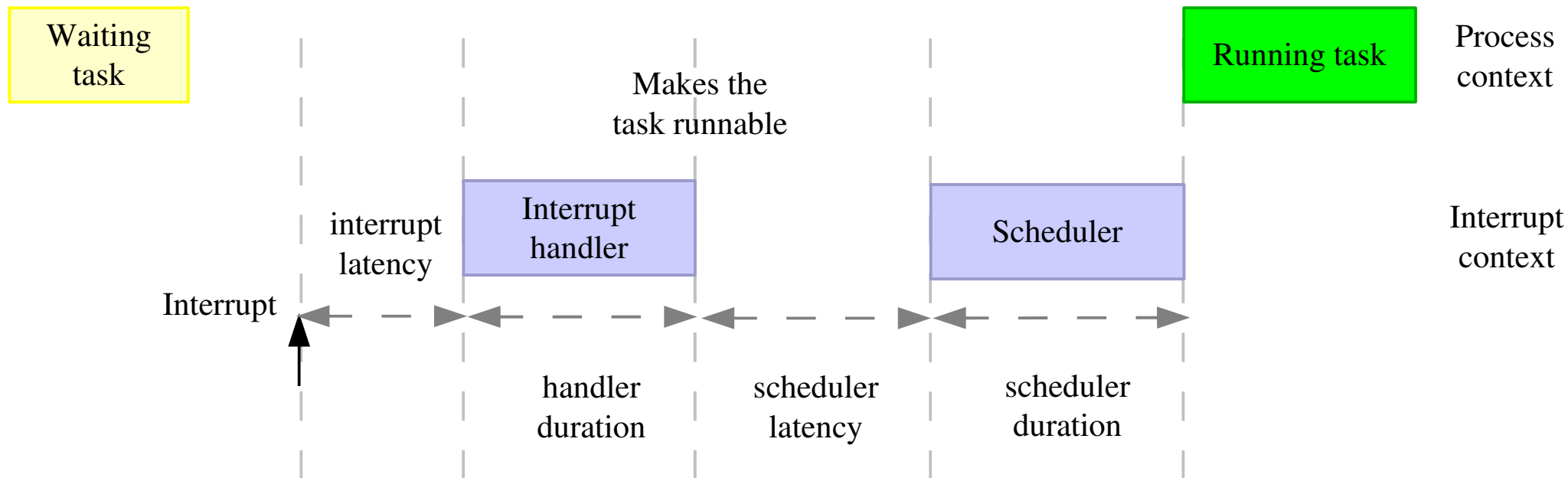
- ▶ AMV infúzní pumpa – RTEMS, uLut, SuiTk
- ▶ Monitoring na PC – Qt + GNU/Linux (případně klient i pro Windows)
- ▶ Infrastruktura pro sběr dat – Wifi AP OpenWRT (MIPS Linux, uLut)



- ▶ 1992 potřeba vybrat ve firmě PiKRON vhodný protokol pro komunikaci mezi MCU (v té době 8051/80552, nyní LPC2xxx, LPC17xx) pro sadu laboratorních přístrojů a zařízení
- ▶ V té době ETHERNET nepřichází pro malá zařízení v úvahu, dnes sice do úvahy přichází, nevýhoda - topologie již není sběrnice
- ▶ CAN pěkný, ale 8 B na zprávu je málo a není integrován všude
- ▶ Profibus – díky patentové politice Profibus International nebezpečná bomba, zdegradoval na řešení bez slovníků a objektové vrstvy, v zásadě master-slave
- ▶ Volba vlastního protokolu nad RS-485, řešení SW nad UART, 9-bit
- ▶ GPL driver Linux, Windows, sysless ARM LPC2xxx, LPC17xx
- ▶ GPL/LGPL/MPL řešení komunikace, objektových slovníků a správy pro zařízení



- ▶ Pro veškeré zde uvedené HW aplikace je kritická spolehlivost časování
- ▶ Nutné použít RT-Priority, RT Exekutivy (RTEMS) nebo plně preemptivní Linux
- ▶ **Pozor:** dnes i tak striktní požadavky na burze, i tam je zájem o RT jádro a odezvy v řádu milisekund



- ▶ Free-Electrons Real-time in embedded Linux systems
<http://free-electrons.com/docs/realtime/>

Michael Opdenacker, Thomas Petazzoni

- ▶ Real-Time Linux Wiki – <http://rt.wiki.kernel.org/>

- ▶ Největší otevřená (LGPL, 3-cl BSD) knihovna FPGA návrhů
- ▶ Založeno 1999 - Damjan Lampret
- ▶ Procesorů (8/53):
 - ▶ OpenRISC, klony Zilog Z80, openMSP430, Plasma (MIPS), PIC16F84, 8051, 8080, Zet 16-bit x86, PIC16F84, ZCPU, TG68 m68k

<http://plasmacpu.no-ip.org:8080/>

▶ Řadiče/periferie:

- ▶ USB 2.0 controller
- ▶ Tri Ethernet 10/100/1000 Mbit
- ▶ Kryptografie DES, AES & RSA
- ▶ HyperTransport Tunnel

Plasma

[Home](#) [CPU](#) [RTOS](#) [TCP/IP](#) [Author](#) [Games](#)

Welcome to the Plasma CPU site.

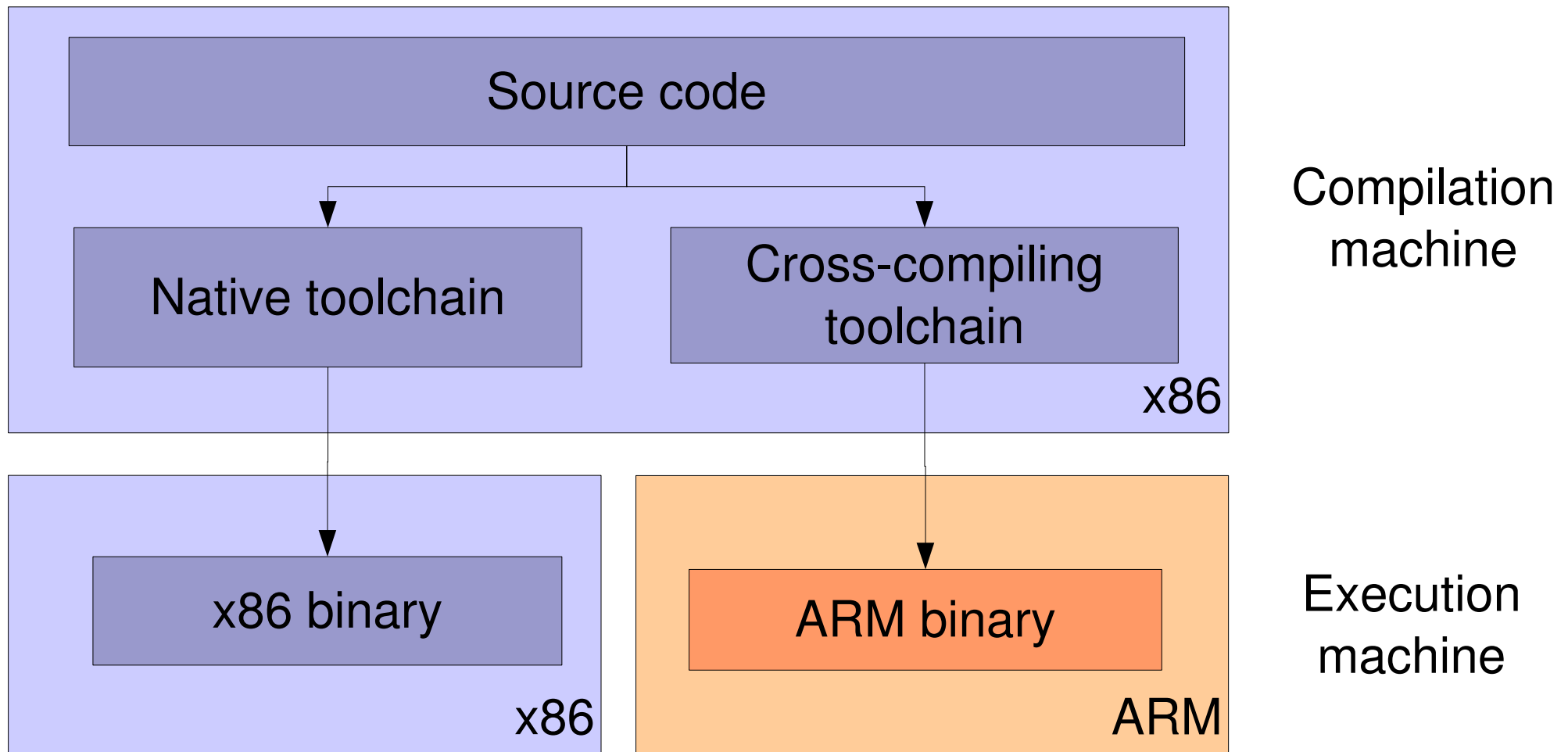
This web server is running on a Plasma CPU. The Plasma CPU is an open source CPU that is written in VHDL and implemented on a Xilinx FPGA.

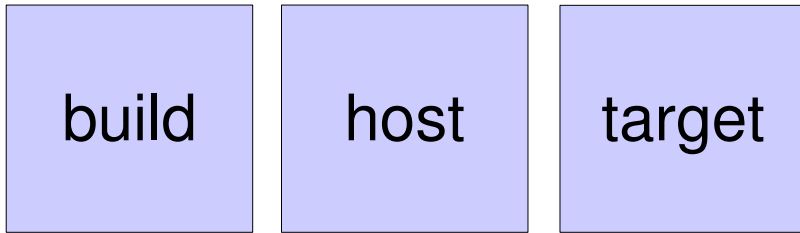


Cross-compiling toolchains

Thomas Petazzoni, Michael Opdenacker, Free Electrons

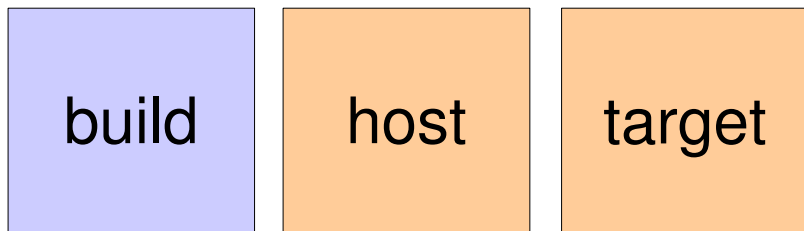
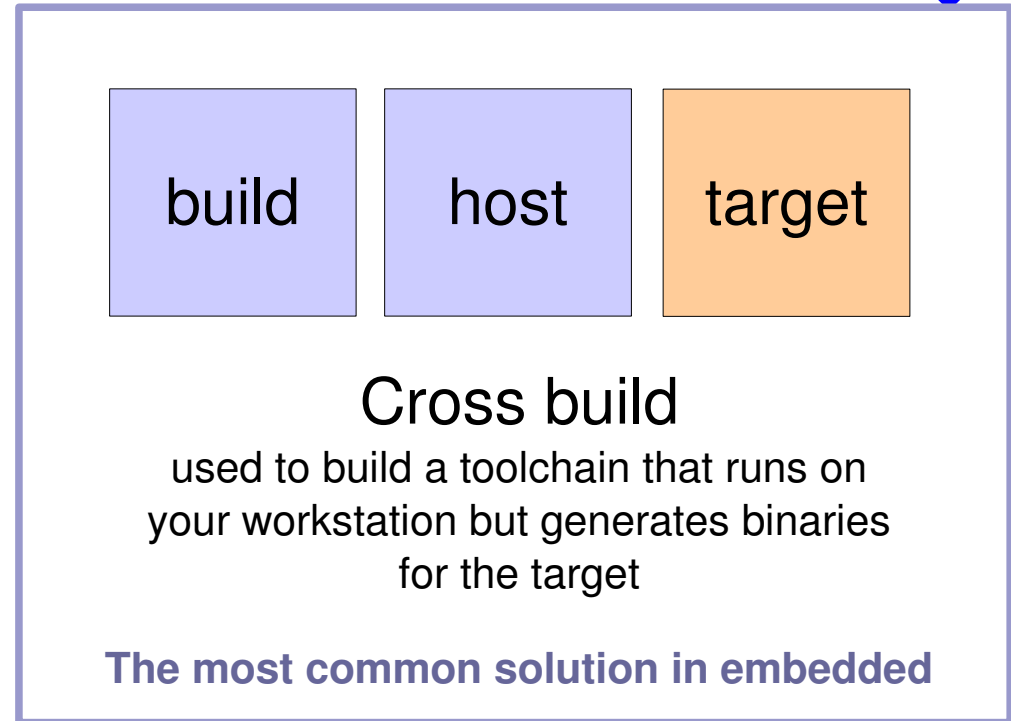
<http://free-electrons.com/docs/toolchains>





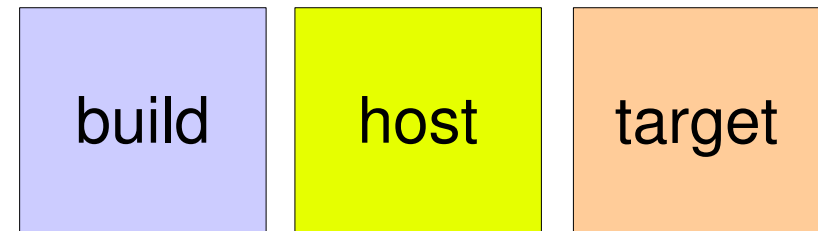
Native build

used to build the normal gcc of a workstation



Cross-native build

used to build a toolchain that runs on your target and generates binaries for the target



Canadian build

used to build on architecture A a toolchain that runs on architecture B and generates binaries for architecture C



```
../../../../gcc-4.3/configure -v --enable-languages=c,c++ \  
--prefix=/usr \  
--with-system-zlib --without-included-gettext --disable-nls \  
--enable-clocale=gnu \  
--enable-threads=posix --enable-tls --enable-secureplt \  
--enable-shared \  
--enable-objc-gc \  
--with-float=soft --enable-cxx-flags=-msoft-float \  
--enable-targets=arm-linux-gnu \  
--enable-target-optspace \  
--enable-symvers=gnu \  
--with-arch=armv4t \  
--enable-checking=release \  
--build=x86_64-linux-gnu \  
--host=x86_64-linux-gnu \  
--target=arm-linux-gnueabi \  
--enable-version-specific-runtime-libs \  
--with-headers=/usr/arm-linux-gnueabi/sys-include
```

Jak skompilovat Binutils, GCC, GLIB-C viz

http://rtime.felk.cvut.cz/hw/index.php/How_to_build_GNU_cross-compilers

- ▶ Při přípravě balíku
 - ▶ aclocal: `configure.ac` → `aclocal.m4`
 - ▶ autoheader: `configure.ac` → `config.h.in`
 - ▶ autoconf: `configure.ac` → `configure`
 - ▶ automake: `Makefile.am` → `Makefile.in`
- ▶ Konfigurace před kompilací na cílovém nebo build systému
 - ▶ `<srcdir>./configure --host=arm-linux-gnueabi --enable-feature --with-package-x=/opt/x`
 - ▶ `Makefile.in` → `Makefile`, `config.h.in` → `config.h`
 - ▶ `configure` je možné ovlivnit i `CFLAGS=x`, `LDFLAGS=x` v prostředí
- ▶ Kompilace
 - ▶ `make all`
- ▶ Instalace
 - ▶ `make DESTDIR=/packaging/root install`

- ▶ Je nutné psát čistě a používat jen to, **co je** jazykem **deklarováno**
- ▶ Používat pokud to je možné standardizovaná **API** (např. **POSIX**, The Open Group Base Specifications, IEEE Std 1003.1)
- ▶ Nepředpokládat pořadí byte/charů ve slově (**little/big-endian**)
endian.h: `__BYTE_ORDER`, `__LITTLE_ENDIAN`, `__BIG_ENDIAN`
byteswap.h: `bswap_16`, `bswap_32`
- ▶ Nepředpokládat **počet bitů** v adresační jednotce (**CHAR_BIT**)
(např DSP, kde char/int/long/float/double = 32-bit)
stdint.h: `int32_t`, `uint32_t`, obecně `intN_t`, `uintN_t`, `(u)int_leastN_t`,
`(u)int_fastN_t`, `(u)intmax_t`, `size_t`, `ssize_t`
- ▶ Nikdy nepřetypovat ukazatel na **int** a zpět, dokonce ani na long
(především, kvůli Win32_64), `intptr_t`, `uintptr_t`
- ▶ `sizeof(struct s)`, `offsetof(struct s, d)`
- ▶ Nezarovnané přístupy nemusí být architekturou podporované,
v každém případě snižují výkonnost
- ▶ Atomické operace `atomic_t` a další jsou-li k dispozici

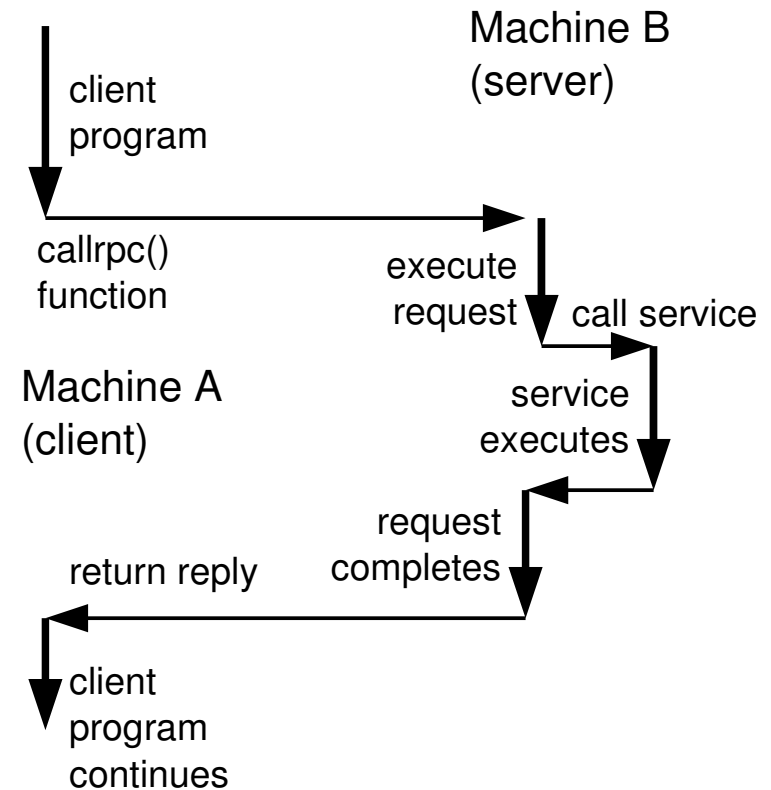
- ▶ Vnější síťové formáty vždy oddělovat oddělovat od vnitřních
 - ▶ Zarovnání struktur se může lišit
 - ▶ Při přenosu binárních dat používat alespoň
 - `uint32_t htonl(uint32_t hostlong);`
 - `uint16_t htons(uint16_t hostshort);`
 - `uint32_t ntohl(uint32_t netlong);`
 - `uint16_t ntohs(uint16_t netshort);`
 - ▶ lépe IDL (Interface description language)/External Data Representation (XDR)
 - ▶ Sun's ONC RPC
 - ▶ The Open Group's Distributed Computing Environment
 - ▶ IBM's System Object Model
 - ▶ Object Management Group's CORBA
 - ▶ nebo textové formáty
 - ▶ XML, XMLRPC, SOAP, HTML, JSON atd.
 - ▶ Pozor: zjednoduší řešení ale parsování je pomalé (FastRPC)



- ▶ Od roku 1995 IETF standard
- ▶ Zajišťuje kódování/zabalení dat/hodnot způsobem, který je nezávislý na architektuře použitého počítačového systému
- ▶ Zakódovaná data mohou být přenášena heterogenním prostředí
- ▶ **Encoding** - kódování lokální reprezentace do přenosového XDR reprezentace
- ▶ **Decoding** – převod XDR reprezentace do lokálního formátu použitému v systému příjemce
- ▶ Kódování čísel, délek atd. je binární a položky jsou zarovnané na 4 byte, datové typy: boolean, int (32 bit), unsigned int (32 bit), hyper (64 bit), unsigned hyper (64 bit), IEEE float, IEEE double, quadruple (new in RFC1832), enumeration, structure, string, fixed length array, variable length array, union, fixed length opaque data, variable length opaque data, void - zero byte quantity, optional (some type + present flag)

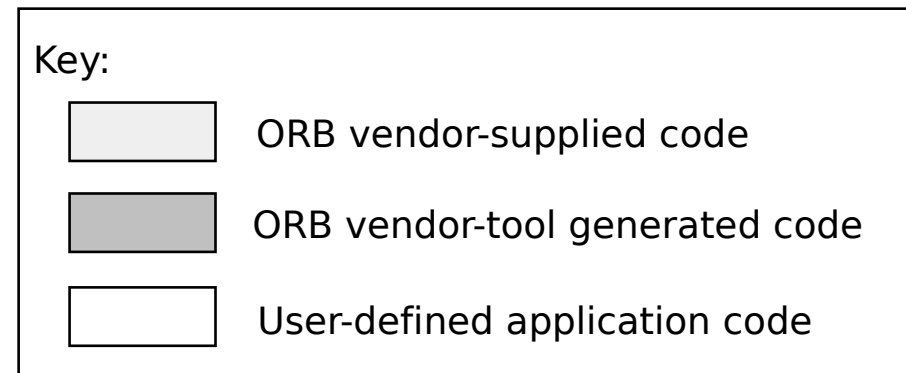
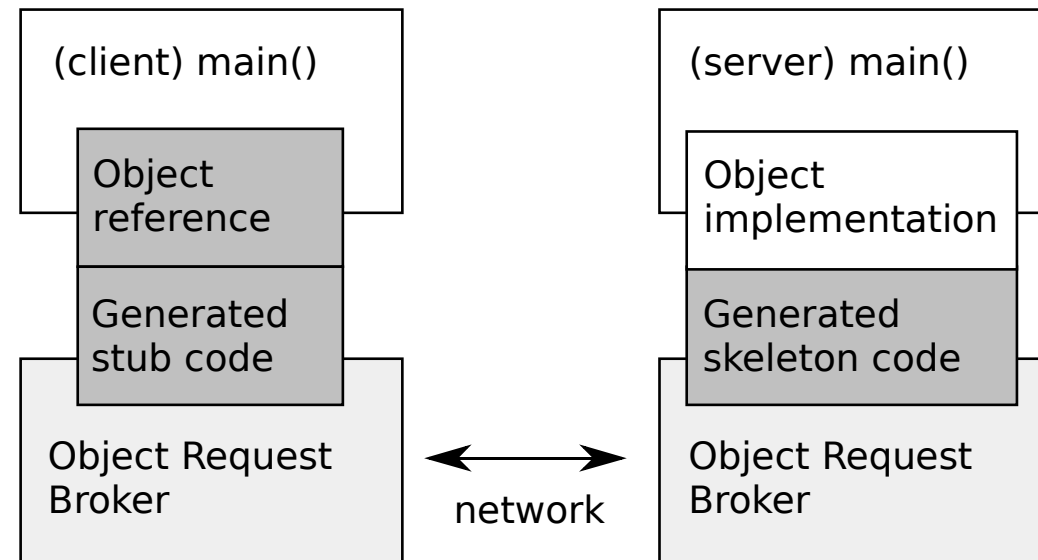
Vzdálené volání – popis API funkcí daný IDL

- ▶ Základní mechanismus pro budování klient-server řešení
- ▶ Na straně klienta volání náhradní funkce
 - ▶ `rpcprog_clnt.c` – klient stub
- ▶ Ta zabalí data do XDR a pošle přes přenosový kanál (TCP/IP)
- ▶ Na straně serveru data přijata vygenerovaným kódem serveru
 - ▶ `rpcprog_svc.c -- server stub`
- ▶ Po dekódování do lokální podoby volaná uživatelská funkce serveru
- ▶ Její návratová data zakódovaná do XDR, předaná klientovi
- ▶ Generovaná funkce v `rpcprog_clnt.c` dekóduje data a předá je volajícímu programu
- ▶ Pomocné společné funkce v `rpcprog_xdr.c` a `rpcprog.h`



<http://www.cs.cf.ac.uk/Dave/C/node33.html>

- ▶ SW vrstva/standard
- ▶ umožňuje budovat objektově orientované aplikace, jejichž komponenty mohou být implementované v různých jazycích a běhových prostředích a mohou běžet na různých počítačích
- ▶ normalizace volání metod a referencí na objekty tak, aby byla síťově stransparentní



- ▶ Free Electrons <http://free-electrons.com/>
- ▶ LinuxDevices.com <http://www.linuxfordevices.com/>
- ▶ Neo FreeRunner – Openmoko <http://wiki.openmoko.org/>
- ▶ Rockbox <http://www.rockbox.org/>
- ▶ RTIME DCE HW Wiki <http://rtime.felk.cvut.cz/hw/>
Vlastní HW vytvořený studenty a členy skupiny, popis přípravy a používání programovacích nástrojů pro různé platformy a odkazy na repository
- ▶ Odkazy na námi řešené projekty zabývajícími se sběrnici CAN
<http://rtime.felk.cvut.cz/can/>
- ▶ Otevřený projekt a návrh protokolu pro laboratorní přístroje, domácí a zemědělskou automatizaci nad linkovou vrstvou RS-485
<http://ulan.sourceforge.net/>
- ▶ Zatím kvůli hrozbám a dezinformacím z Profibus International blokováné GPL implementace protokolu Profibus
<http://profim.sourceforge.net/>
<http://www.pbmaster.org/>
- ▶ Projekt letového SW pro vrtulník <http://rtime.felk.cvut.cz/helicopter/>