

eCos in commercial use - the Sinar eMotion

Outline



- Introduction
- Sinar eMotion
- Overview Operating Systems
- Application Design
- eCos
- Development Environment
- Roundup

*Dipl.Ing. Alexander Neundorf
<alexander.neundorf AT jenoptik.com>
Jenoptik Laser, Optik, Systeme GmbH*

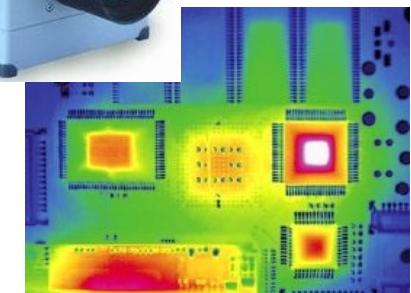
Introduction

eCos in the Sinar eMotion



Jenoptik LOS GmbH / Sensor Systems

- located in Jena, Germany, „Saalecon Valley“
- 500 employees
- highly specialized products
- digital cameras, thermography cameras, optical modulators and more



Sinar eMotion characteristics

eCos in the Sinar eMotion

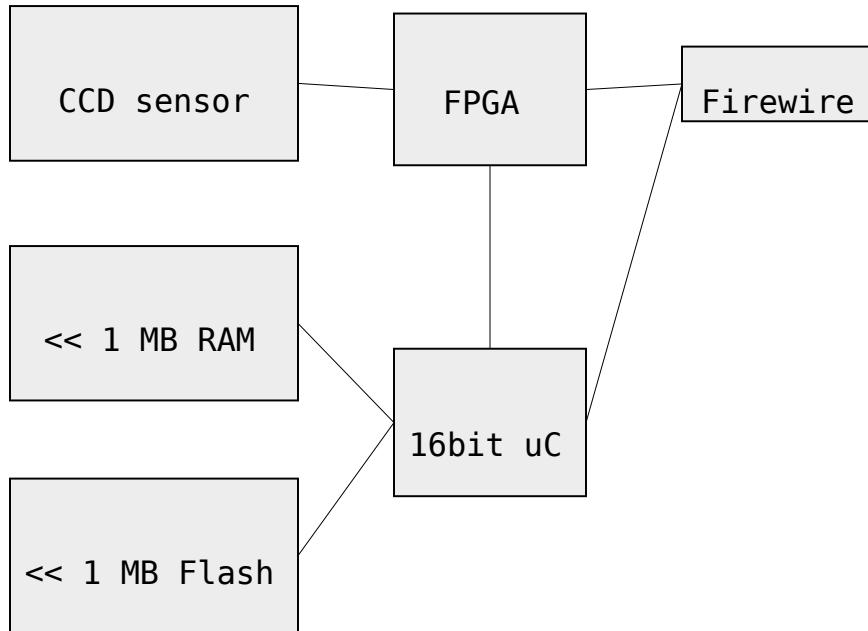


- digital back for medium format cameras
- 22 and 33 megapixel 14 bit CCD sensors
- up to 50 images/minute
- 6 GB internal flash for up to 120 raw image files
- CF card, Firewire
- Compatible to Hasselblad H1, Sinar m, generic X contact, Mamiya 645 AFD, Contax 645



A typical microscopy camera - ProgRes

eCos in the Sinar eMotion

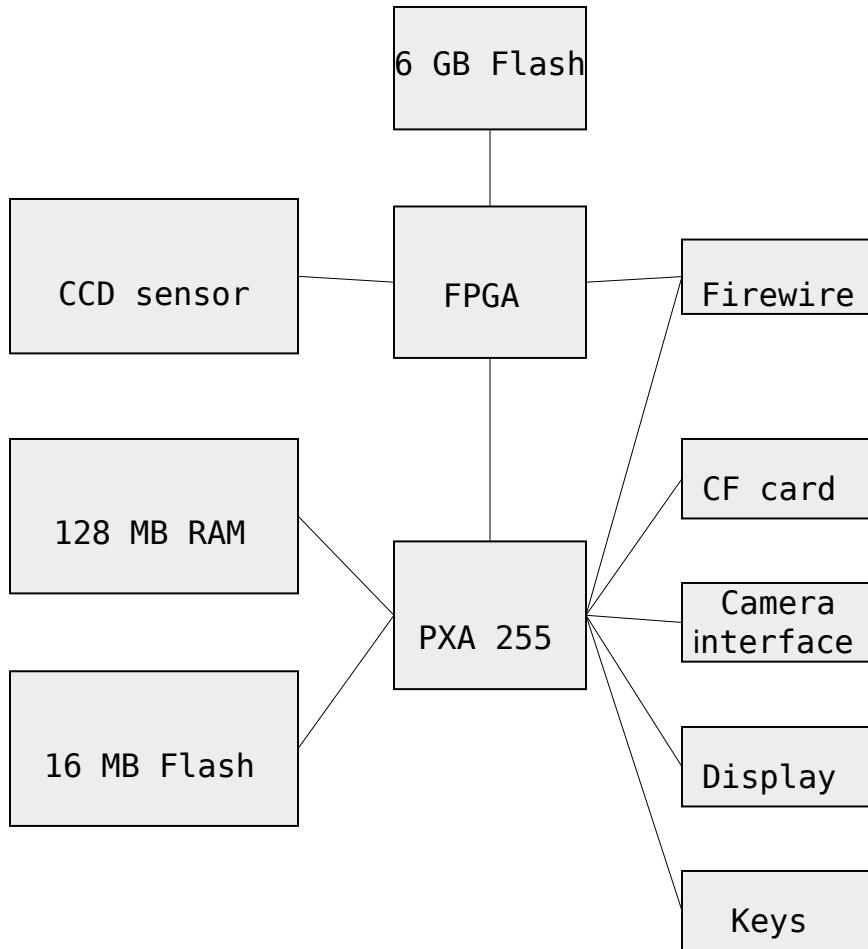


- tethered operation only
- MS Windows, Mac OS X, Linux
- high quality raw images
- high frame rates
- GPIO interfacing
- C Compiler, no OS



Hardware concept and requirements

eCos in the Sinar eMotion



- fast startup, < 5 s
- fast interrupt response, << 125 us
- Firewire, CF + FAT16/32 driver
- high data throughput
- simple GUI, i18n
- optimized image processing
- safe update mechanism
- high quality raw images



Roll-your-own

- high effort, 1 developer, untested -> buggy, no support, no drivers, no docs, ...

QNX, VxWorks

- high initial costs, no previous experience
- commercial RTOS, “high end”, some drivers/software available
- local commercial support

RTEMS

- open source RTOS, some drivers/software available
- no experience/contacts, no local commercial support

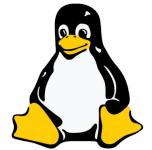
Embedded Operating Systems (2)

eCos in the Sinar eMotion



Linux

- long startup time (5..10s), very complex, no RTOS, separation kernel/userspace
- (only) some experience, firewire driver ?
- community and local commercial support, many drivers/software available



Windows CE

- only up to 32 MB per process, separation kernel/userspace, royalties
- no experience, commercial RTOS, medium costs, firewire driver ?
- local commercial support, many drivers/software available

eCos

- not too many users/developers
- open source RTOS, some drivers/software available
- low cost, Linux synthetic target, fast startup, direct hardware access, easy firewire driver porting, community and local commercial support

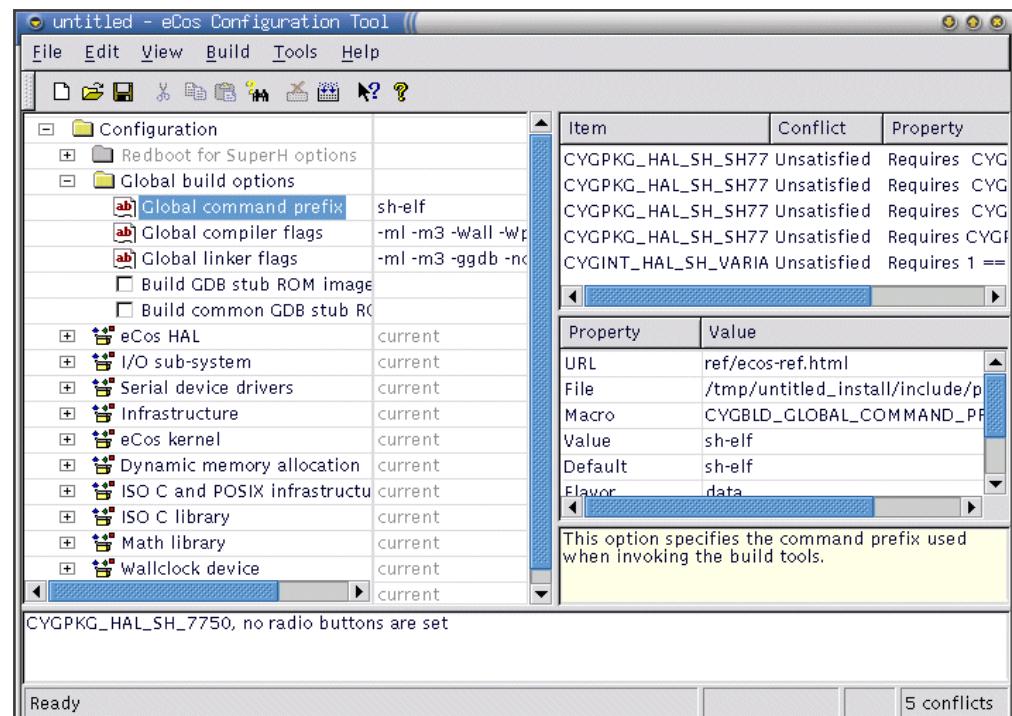


eCos – the embedded Configurable operating system

eCos in the Sinar eMotion

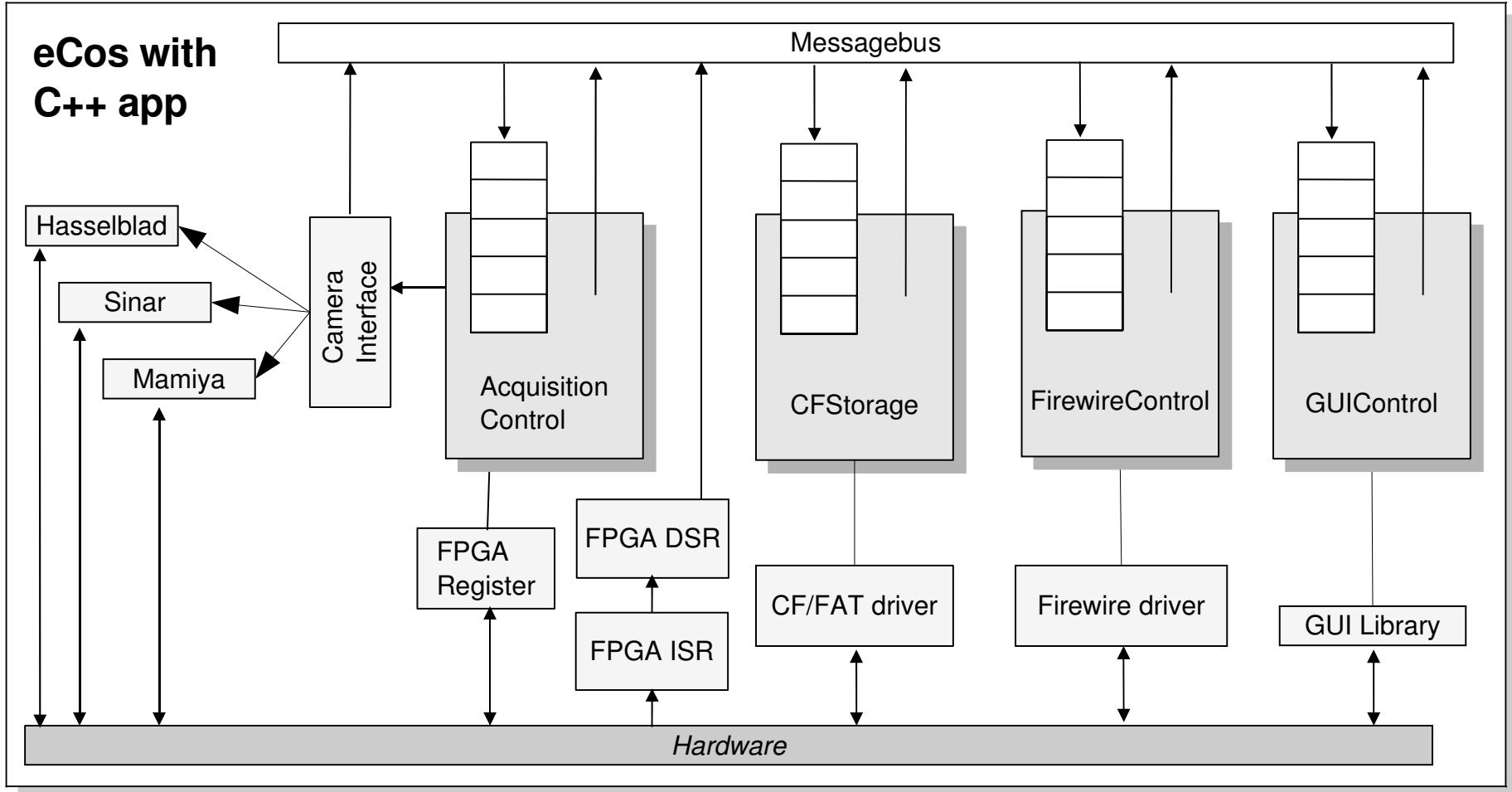


- started by Cygnus, then RedHat, now eCosCentric Inc.
- one static application image, runnable in RAM and ROM („XIP“)
- multithreaded RT kernel, single user, single process
- GNU toolchain: reliable and standard conformant C/C++ compiler
- highly portable: x86, ARM, MIPS, PowerPC, Sparc and more
- small footprint, e.g. ARM7 40 Mhz
with 32 KB RAM, 64 KB ROM
- development platforms Linux
and Windows/cygwin
- C library: input/output, math, etc.,
parts of POSIX
- FreeBSD TCP/IP stack
- highly configurable



Application architecture

eCos in the Sinar eMotion



Usage of eCos in the eMotion (1)

eCos in the Sinar eMotion



Timeline

- start 04/2004, demonstration at Photokina 09/2004, shipping 01/2005
- Linux synthetic target + Qvfb -> develop without hardware
- Bootloader „Redboot“ running after two weeks

Realtime issues

- eCos enables realtime applications
- deterministic scheduler, highest priority thread runs
- sync. primitives: mutex, semaphore, message queue, condition and more
- multi-stage interrupt handling: ISR -> DSR (Deferred service routine) -> thread
- avg. interrupt response time: 5 us (PXA 255, 200 Mhz)
- no dynamic memory allocation in the kernel (neither in the application)

Licensing

- GPL + exception: allows closed source application and 3rd party software
- no GPL (or LGPL) software usable with closed source application

Usage of eCos in the eMotion (2)

eCos in the Sinar eMotion

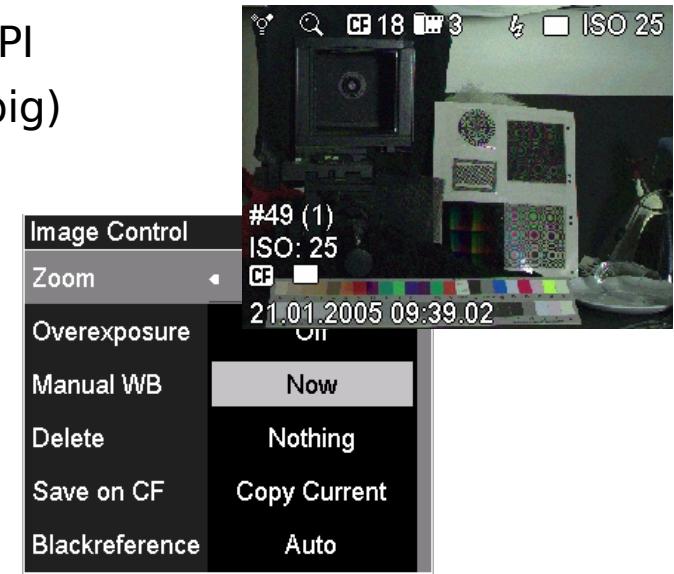


GUI

- simple custom GUI with i18n
- Alternatives:
 - SwellSoftware PEG (Portable Embedded GUI), commercial
 - *MiniGUI* (Beijing Feynman Software Inc.), GPL/commercial
 - *MicroWindows*, free (Mozilla Public License)
drawing primitives, X API, e.g. FLTK, Win32 API
 - not appropriate: PicoGUI (dead), Qt/E (quite big)

Drivers

- custom drivers for SPI, I2C, FPGA, display, keys
- since 2005/2006: SPI and I2C framework in eCos
- Firewire: Jenoptik driver ported
- CF + FAT16/32: commercial 3rd party driver
- CF/MMC driver with FAT16/32 in eCos



Development environment

eCos in the Sinar eMotion



- Linux: KDevelop, CuteCom
- Windows: cygwin, Eclipse/Programmers Notepad, Insight, Terminal Bray++
- CMake buildsystem
 - Linux: Makefiles and KDevelop projects
 - Windows: cygwin and nmake Makefiles
- eCos from cvs

The screenshot displays a multi-window development environment. On the left, the 'CuteCom' application window shows serial port settings (Device: /dev/ttyS0, Baud rate: 115200) and a terminal window with a dump of memory contents. The terminal output includes:

```
RedBoot> dump -b 0 -l 0x80
00000000: 18 F0 9F E5 18 F0 9F E5 18 F0 9F E5 18 F0 9F E5 [.....]
00000010: 18 F0 9F E5 A5 BC 65 45 18 F0 9F E5 18 F0 9F E5 [.....]
00000020: 48 04 00 50 90 05 00 50 B4 05 00 50 R0 05 00 50 [H...P...P...P]
00000030: FC 05 00 50 E7 21 31 35 14 07 00 50 DC 06 00 50 [....P..115...P...P]
00000040: 00 00 00 00 00 00 00 C5 E5 AD D9 24 B5 A5 AD [.....$...]
00000050: 17 00 08 00 D4 94 00 50 D4 94 00 50 D4 94 00 50 [....P...P]
00000060: D4 94 00 50 2C D0 00 00 00 FC 96 00 50 [....P...P]
00000070: F8 96 00 50 D4 94 00 50 D4 94 00 50 D4 94 00 50 [....P...P...P...P]

RedBoot>
```

Below the terminal, there are buttons for 'Clear', 'Hex output', 'Append to...', and a log file selection field (/home/alex/cutecom.log).

In the center, the KDevelop IDE window shows code completion for a function call:

```
error_t doShutdown(int argc, char** argv)
{
    Base::Thread *recv=g_threadManager.getThread(argv[1]);
    if(recv!=NULL)
    {
        Base::Event *event=g_messagePool.allocEvent();
        if (event!=0)
        {
            event->sender=g_threadManager.getThread(cyg_thread_self());
            event->id=EvShutdown;
            recv->post
        }
        else
            void postEvent(Event* event)
        error_t postEventSync(Event* event)
        return (E_NL)
    }
    else
    {
        diag_printf("thread(%s) not found\n",arg1);
    }
}
```

A tooltip provides details about the selected function:

Container: Base::Thread
Kind: Function
Access: public
File: /home/alex/src/work/xscale/firmware/eotr/app/test/testfunctions.cpp
Line: 752 Column: 11

Send an event to this thread, don't wait for completion.
Must not be called from a DSR, use Thread::dsrPostEvent() instead.

On the right, a terminal window shows compilation logs with numerous warnings from the C compiler (cc1plus). Some of the warnings include:

- /home/alex/src/work/ecos/ecos/packages/hal/arm/arch/current/src/hal.c: In function `void cyg_trace_dump()':
- /home/alex/src/work/ecos/ecos/packages/infra/current/src/simple.cxx:481: warning: unsigned int format, pointer arg (arg 3)
- cc1plus: warning: command line option "-Wstrict-prototypes" is valid for C/ObjC but not for C++
- cc1plus: warning: command line option "-Wstrict-prototypes" is valid for C/ObjC but not for C++
- /home/alex/src/work/ecos/ecos/packages/language/c/libc/stdio/current/src/input/vfscanf.cxx: In function `int vfscanf(CYG_ADDRESS (*)[999], const char*, void*)':
- /home/alex/src/work/ecos/ecos/packages/language/c/libc/stdio/current/src/input/vfscanf.cxx:218: warning: unused variable 'ldp'
- cc1: warning: command line option "-Woverloaded-virtual" is valid for C++/ObjC++ but not for C

At the bottom, a status bar indicates "Type of recv-> is Thread (resolved)" and "Zeile: 899 Sp: 20 Einf NORM *".

Debugging eCos in the Sinar eMotion



- CYG_TRACE(bool, <format str>, args); CYG_ASSERT(bool, <format str>);
- Debugging via gdb: serial line, ethernet
- Windows: Insight, Eclipse
- Linux: Insight, Eclipse, KDevelop, ddd, kdbg
- JTAG debuggers: Abatron BDI2000, Ronetix PEEDI: gdb compatible



The screenshot shows a debugger interface with the following components:

- Registers pane:** Displays the CPU registers (eax, ecx, edx, ebx, esp, ebp, esi) with their current values (0x0, 0xffffffff, 0xa0, 0x50, 0xbfffffa28, 0xbfffffa68, 0x40012020) and corresponding stack frames (st0 to st6).
- Local Variables pane:** Displays local variables and their values: argc = 2, argv = 0xbfffffab4, i = 20, endp = 0xbfffffa68, and args = struct captured_main_args (...).
- Source Window pane:** Shows the source code file pil.c with the main function. Line 79 is highlighted in yellow, indicating the current instruction being executed.
- Status bar:** Shows "Program stopped at line 79" and the memory address 0x804882b.

Software quality measures

eCos in the Sinar eMotion



- Source control system: *cvs*
- Bug and issue tracking: *Trac* (<http://trac.edgewall.org>)
- *CMake* -> easy integration with *Dart2* (<http://www.cmake.org>)
- *Dart2* software quality server (<http://www.na-mic.org/Wiki/index.php/DartSummary>)
 - continuous and nightly builds including unit tests
- Coding style: *KWStyle* (<http://public.kitware.com/KWStyle/>)

4 Files Changed by 2 Authors as of 2007-02-10 01:00 GMT

Nightly Builds

Site	Build Name	Update	Cfg	Build			Test				
				Error	Warn	Min	NotRun	Fail	Pass	NA	Min
r36n11.pbm.ihost.com	AIX53-xIC	4	0	0	0	4.3	0	0	71	15	25.8
IcculusDotOrgBeOS	beos5pemax-gcc2.9-991026	1	0	0	0	10.4	0	2	67	17	20.5

Roundup

eCos in the Sinar eMotion



- eMotion shipping since 2005, more projects in work
- core components stable and basically bug free
- instant startup (<< 1 s, application dependent)
- community and commercial support available
- low system complexity - easy transition from non OS projects
- suited for single purpose devices
- configurability – partly *many* preprocessor directives
- GPL+exception: good for the companies, bad for contributing back
- thorough patch review – eCos cvs always stable, but slow progress
- Contributions: some bug fixes, failsafe update mechanism, USB 2.0 support pending