

# eCos in commercial use - the Sinar eMotion

## Outline



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

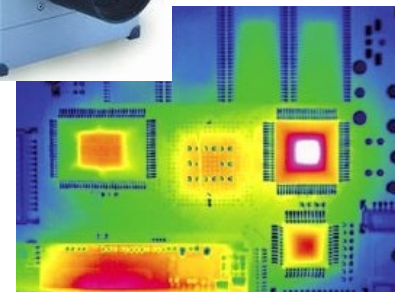
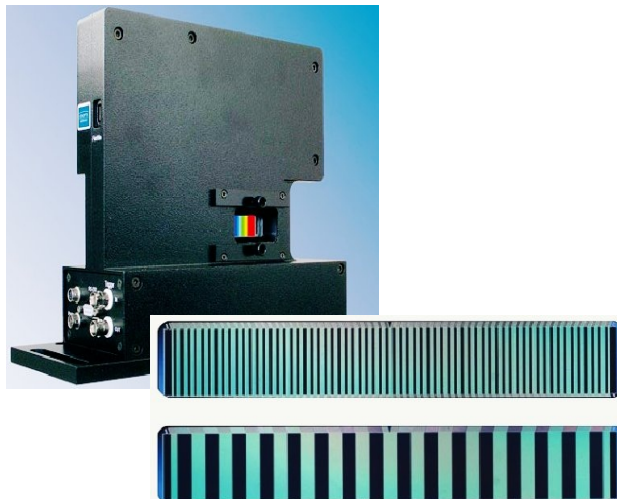
*Dipl.Ing. Alexander Neundorf*  
<*alexander.neundorf AT jenoptik.com*>  
*Jenoptik Laser, Optik, Systeme GmbH*

# 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



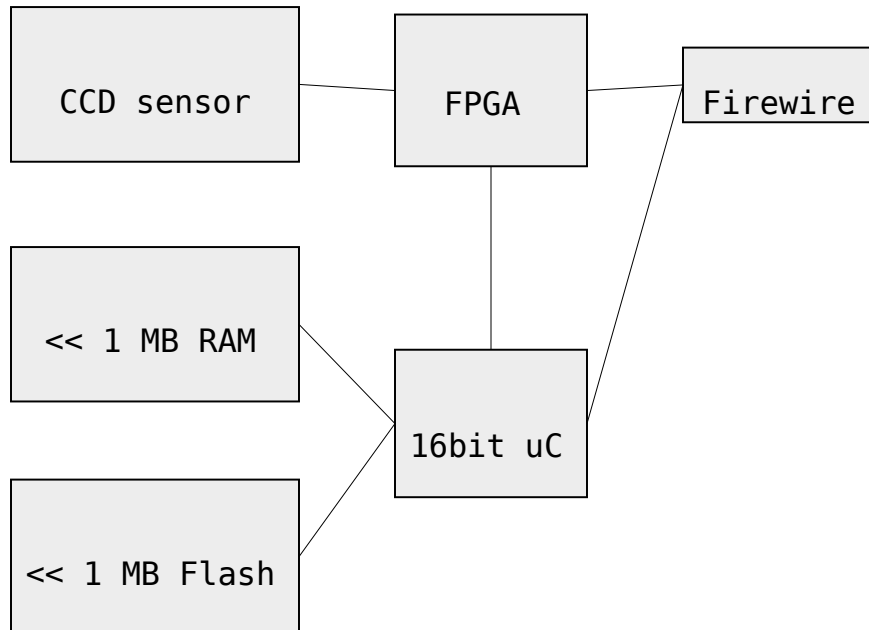
# Sinar eMotion characteristics

## 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

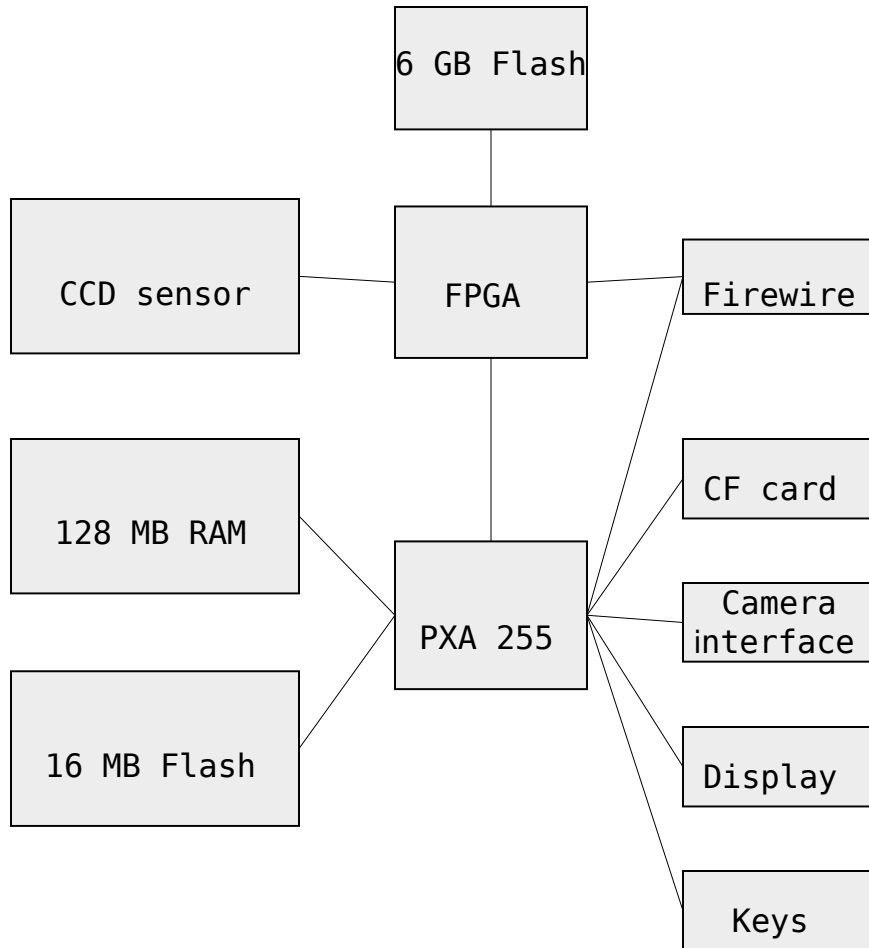


- tethered operation only
- MS Windows, Mac OS X, Linux
- high quality raw images
- high frame rates
- GPIO interfacing
- C Compiler, no OS



# 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

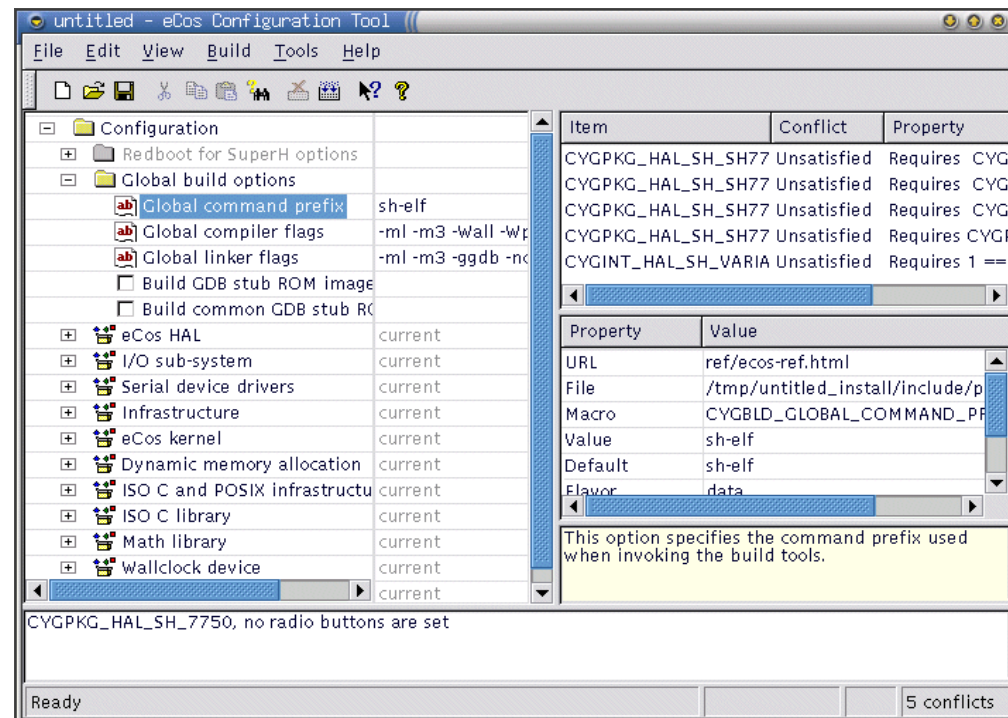


# eCos – the embedded Configurable operating system

## eCos in the Sinar eMotion



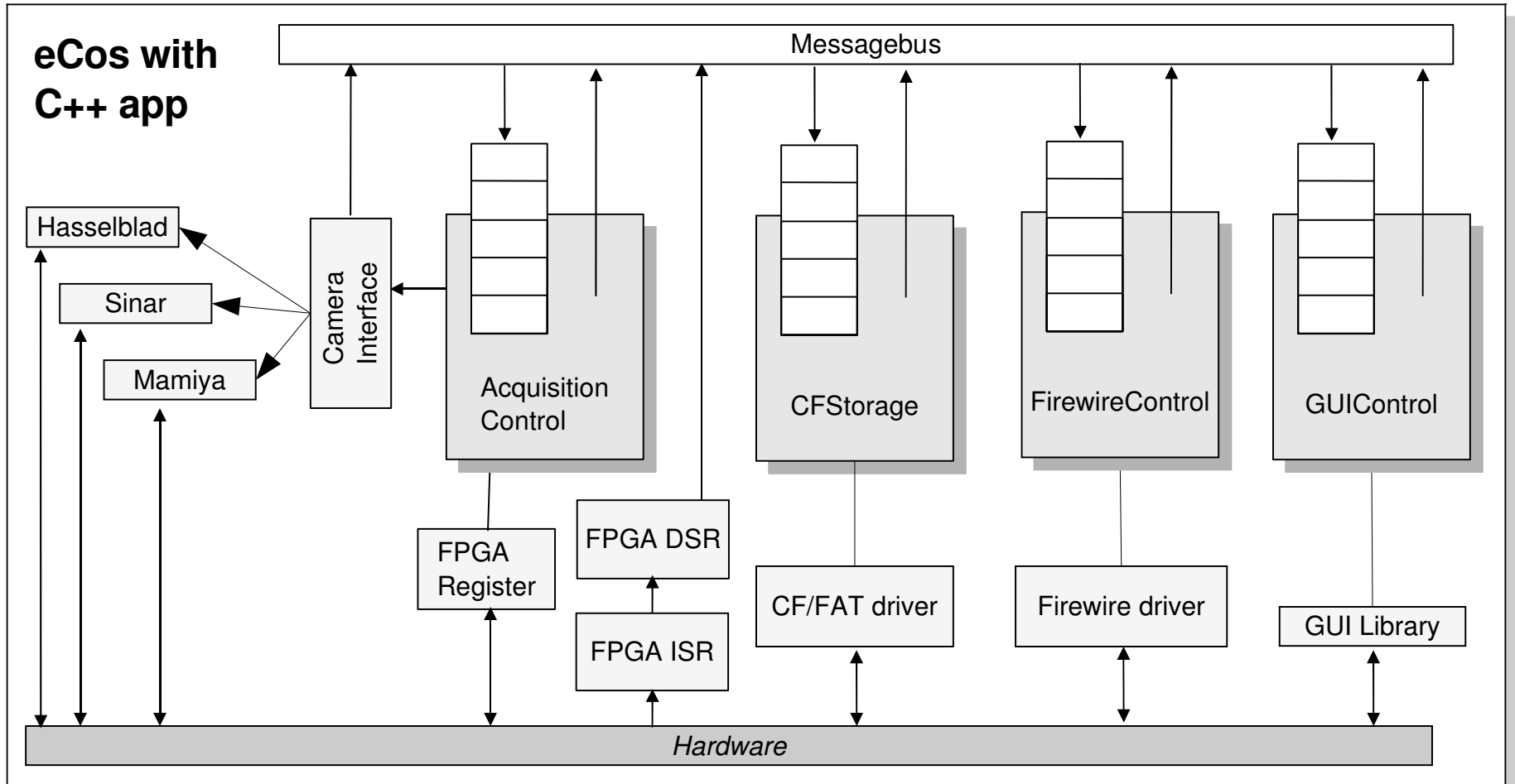
- started by Cygnus, then RedHat, now eCosCentric Inc.
- one static application image, runnable in RAM and ROM („XIP“)
- multithreaded RT kernel, single user, single process
- GNU toolchain: reliable and standard conformant C/C++ compiler
- highly portable: x86, ARM, MIPS, PowerPC, Sparc and more
- small 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 3<sup>rd</sup> party software
- no GPL (or LGPL) software usable with closed source application

# Usage of eCos in the eMotion (2)

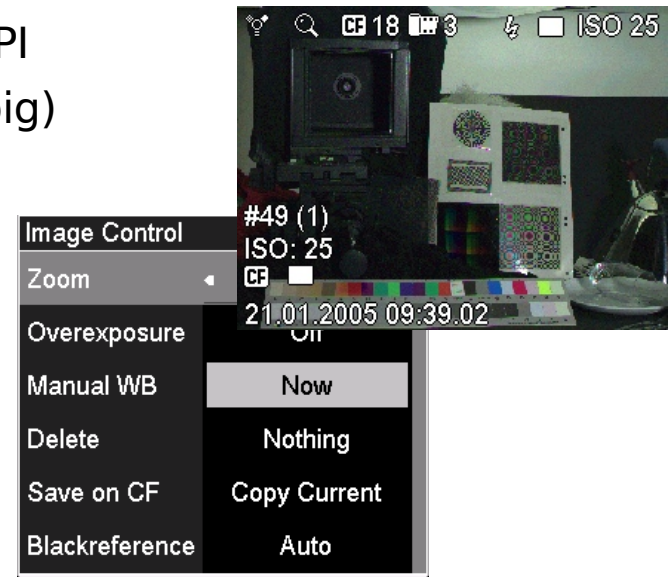
## eCos in the Sinar eMotion

### 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

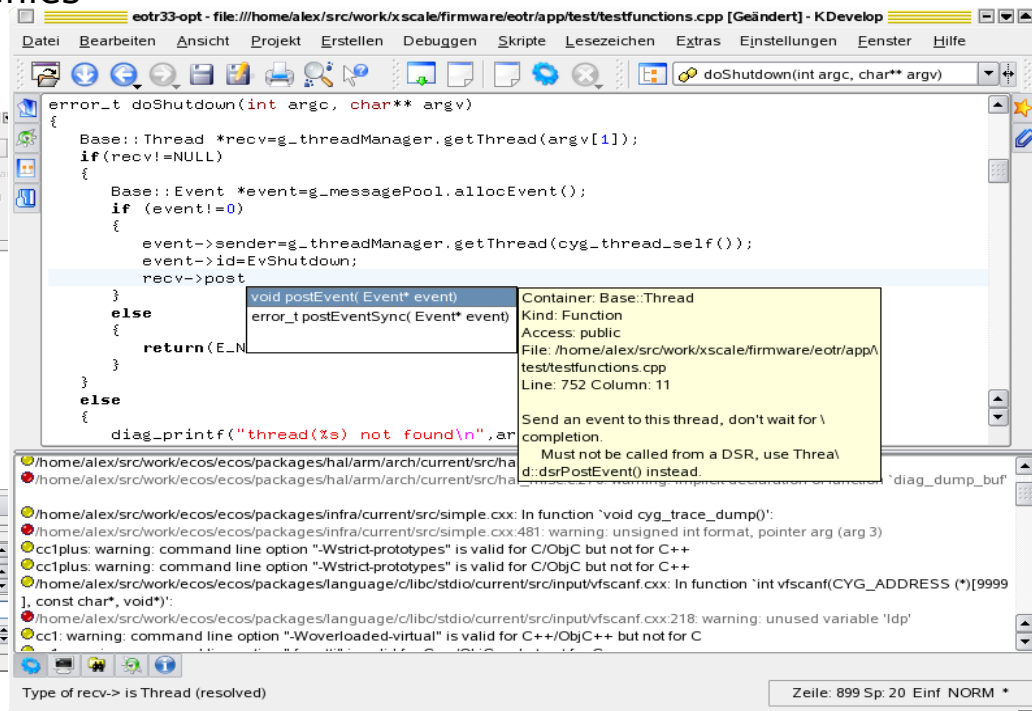
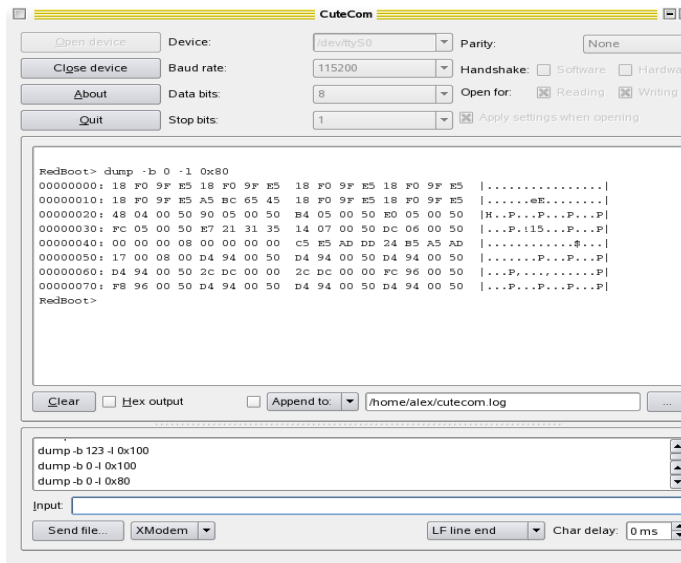
- custom drivers for SPI, I2C, FPGA, display, keys
- since 2005/2006: SPI and I2C framework in eCos
- Firewire: Jenoptik driver ported
- CF + FAT16/32: commercial 3<sup>rd</sup> party driver
- CF/MMC driver with FAT16/32 in eCos



# 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



# 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 window titled "pil.c - Source Window". The source code is displayed in a window with a menu bar (File, Run, View, Control, Preferences, Help) and a toolbar. The code is as follows:

```
69 while (cnt < n)
70 {
71     for (i = 0; ++i <= (int)n -
72         {
73         mf[i] += 10L;
74         ms[i] += 10L;
75     }
76     for (i = (int)(n - cnt + 1);
77         {
78         temp = 2 * i - 1;
79         shift(&mf[i - 1], &mf[
80         shift(&ms[i - 1], &ms[
81     }
82     nd = 0;
83     shift((long *)&nd, &mf[1], 1
84     shift((long *)&nd, &ms[1], 1
85     xprint(nd);
86 }
87 printf("\n\nCalculations Completed!\n");
88 free(ms);
89 free(mf);
90 return(0);
91
```

The line 79 is highlighted in yellow. The "Registers" window shows the following values:

Register	Value	Symbol
eax	0x0	st0
ecx	0xffffffff	st1
edx	0xa0	st2
ebx	0x50	st3
esp	0xbfffa28	st4
ebp	0xbfffa68	st5
esi	0x40012020	st6

The "Local Variables" window shows the following values:

```
argc = (int) 2
argv = (char **) 0xbfffab4
i = (int) 20
endp = (char *) 0xbfffa68 "\210Ã°Ã¿Ã¸Ã¸\001B\002"
args = (struct captured_main_args) {...}
```

The status bar at the bottom indicates "Program stopped at line 79" and "0x804882b 79".






# 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-xlC  	4	0	0	0	4.3	0	0	21	15	25.8	
 lcculusDotOrgBeOS	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 ( $\ll 1$  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