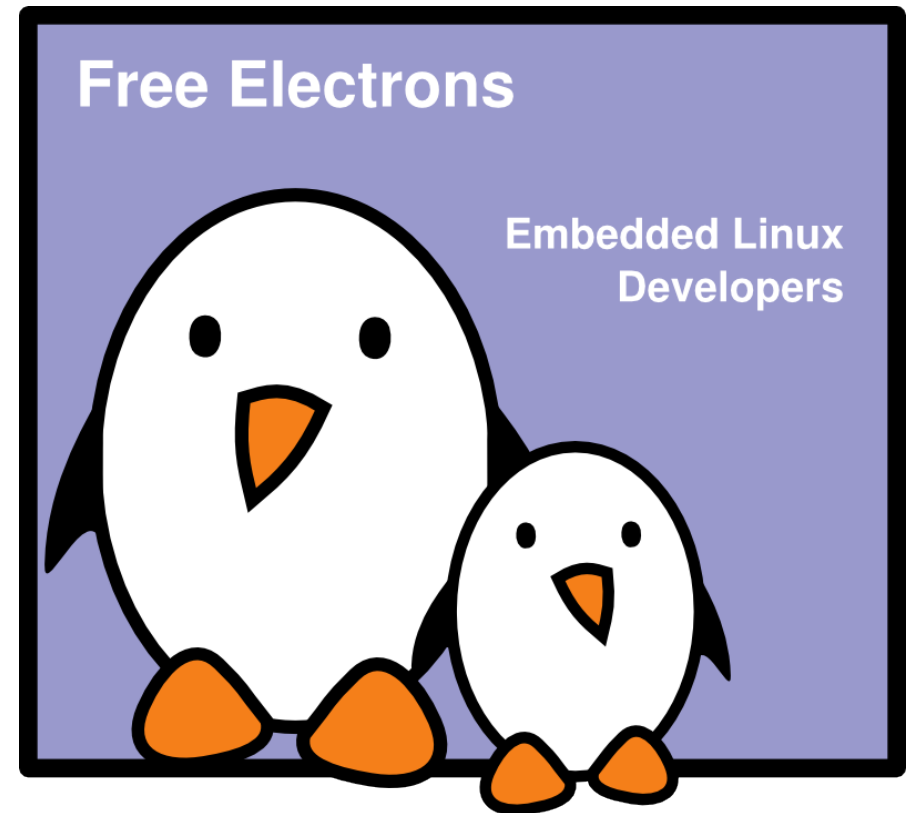
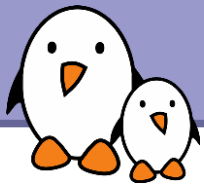


## Multimedia in embedded Linux systems

**Free Electrons**





# Rights to copy

© Copyright 2004-2009, Free Electrons  
[feedback@free-electrons.com](mailto:feedback@free-electrons.com)

Document sources, updates and translations:  
<http://free-electrons.com/docs/multimedia>

Corrections, suggestions, contributions and translations are welcome!

Latest update: Sep 15, 2009



**Attribution – ShareAlike 3.0**

**You are free**

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

**Under the following conditions**



**Attribution.** You must give the original author credit.



**Share Alike.** If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

License text: <http://creativecommons.org/licenses/by-sa/3.0/legalcode>



# Scope of this training

- ▶ **Multimedia in embedded Linux systems**

This training targets the development of multimedia-capable embedded Linux systems. Though it can be useful to playing or creating multimedia content on GNU/Linux desktops, it is not meant to cover everything about this topic.

- ▶ Audio in embedded Linux systems is already covered in a dedicated training: <http://free-electrons.com/training/audio>

- ▶ **Linux 2.6**

This training only targets new systems based on the Linux 2.6 kernel. This way, you leverage the most advanced technology and don't learn about something getting obsolete.



# Contents

## Introduction

- ▶ Glossary

## Multimedia libraries

- ▶ SDL
- ▶ DirectFB

## Kernel subsystems

- ▶ Video4Linux API
- ▶ DVB API

## Free Software Video

- ▶ Video codecs and file formats
- ▶ Multimedia distributions



# Quick Glossary (1)

- ▶ **Alpha channel:**

A additional value in image pixels, describing their opacity (1: opaque, 0: transparent)

- ▶ **Blitting:** from BitBLT "Bit Block Transfer"

Copying image data (e.g. copying a surface on another), applying image combination operations.

- ▶ **Keying:**

When compositing 2 image frames together, removing parts of one according to a key (like its color or brightness), to reveal the corresponding part in the other.



# Quick Glossary (2)

▶ **VBI:** Vertical Blanking Interval

Interval during which the transmission of a video signal is suspended to let the electron gun go back to the first screen line. Used to transmit extra data, such as teletext.

See <http://wikipedia.org> for details!



## Multimedia libraries SDL



# SDL

Simple DirectMedia Layer

<http://libsdl.org/>

License: LGPL



- ▶ Cross-platform multimedia library
- ▶ Originally created for writing games, by Sam Lantinga, for Loki Entertainment Software, specializing in porting games to GNU/Linux.
- ▶ Provides low level access to audio, keyboard, mouse, joystick, 3D hardware via OpenGL, and 2D video framebuffer.
- ▶ Native C API  
Bindings available for most programming and scripting languages





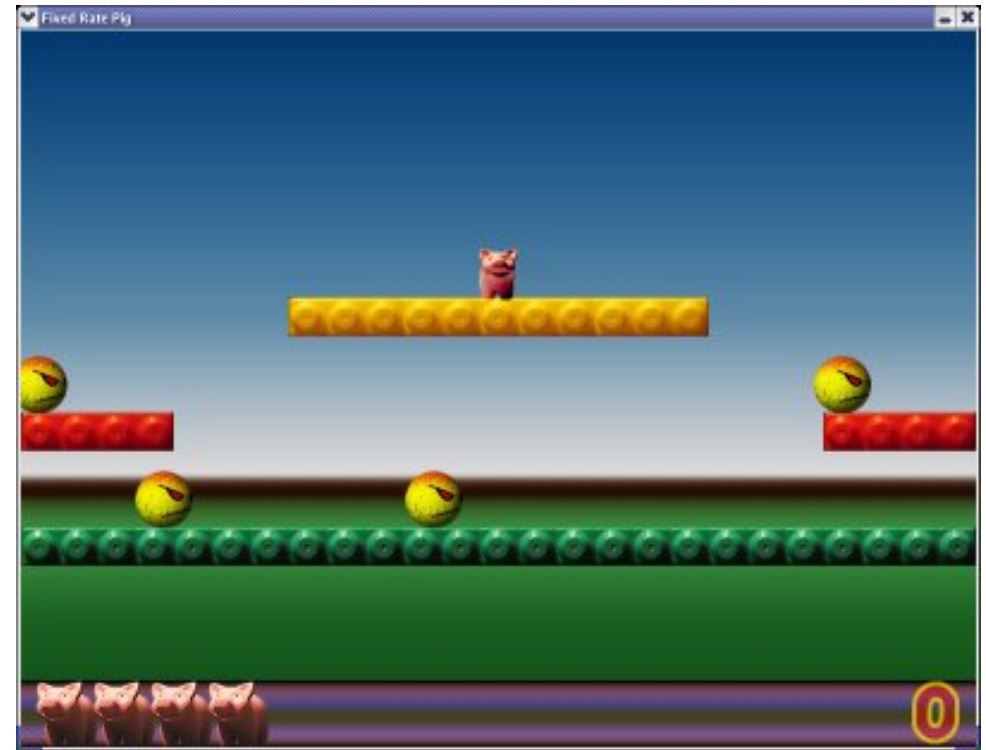
# SDL screenshots

```
QEMU
Partition check:
hda:
me.c:v1.10 9/23/94 Donald Becker (becker@scyld.com)
Last modified Nov 1, 2000 by Paul Gortmaker
NE*000 ethercard probe at 0x300: 52 54 00 12 34 56
eth0: NE2000 found at 0x300, using IRQ 9.
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 2048)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 72k freed
EXT2-fs warning: mounting unchecked fs, running e2fsck is recommended

Linux version 2.4.20 (bellard@voyager) (gcc version 2.95.2 20000220 (Debian GNU/
Linux)) #45 Sat Aug 9 02:26:16 CEST 2003

This is the Red Hat rescue disk. Most of the basic system commands are
in /bin.

Type exit to halt the system.
#
```



QEMU: a CPU and system emulator

Pig: a demo arcade game. 7000 lines.



# SDL supported operating systems

- ▶ Officially supports: Linux, Windows, Windows CE, BeOS, MacOS, MacOS X, FreeBSD, NetBSD, OpenBSD, BSD/OS, Solaris, IRIX, and QNX.
- ▶ Code with no official support: AmigaOS, Dreamcast, Atari, AIX, OSF/Tru64, RISC OS, SymbianOS and OS/2.

Makes it very easy to create platform independent applications!



# SDL capabilities (1)

## Video

- ▶ Video mode setting at any depth (8-bpp or greater). Automatic detection of the closest available mode. Optional conversion if the video mode is not supported by the hardware.
- ▶ Writing directly to a linear graphics framebuffer.
- ▶ Creating surfaces with colorkey or alpha blending attributes.
- ▶ Hardware accelerated blit and fill operations are used if supported by the hardware (using MMX acceleration in x86).
- ▶ OpenGL support.
- ▶ Facility to load BMP images.



# SDL capabilities (2)

## Events

- ▶ Application visibility changes
- ▶ Keyboard input
- ▶ Mouse input
- ▶ User-requested quit
- ▶ Each event can be enabled or disabled with `SDL_EventState()`.
- ▶ Events passed through a user-specified filter function before being posted to the internal event queue.
- ▶ Thread-safe event queue.



# SDL capabilities (3)

## Audio

- ▶ Audio playback of 8-bit and 16-bit audio, mono or stereo. Optional conversion if the format is not supported by the hardware.
- ▶ Audio run independently in a separate thread, filled via a user callback mechanism.

## CD-ROM audio

- ▶ Complete CD audio control API



# SDL capabilities (4)

## Threads

- ▶ Simple thread creation API.
- ▶ Simple binary semaphores for synchronization.

## Timers

- ▶ Measure elapsed milliseconds.
- ▶ Wait for  $n$  milliseconds.
- ▶ Set a single periodic timer with 10ms resolution.

## Endian independence

- ▶ Detect the endianness of the current system.
- ▶ Routines for fast swapping of data values.
- ▶ Read and write data of a specified endianness.



# SDL implementation on GNU/Linux

- ▶ Video display. Either uses
  - ▶ X11: taking advantage of XFree86 DGA extensions and new MTRR acceleration for fullscreen display.
  - ▶ Or DirectFB (see the next section)
- ▶ Uses the OSS API for sound. Fine for Linux systems with ALSA, thanks to OSS emulation.
- ▶ Threads are implemented using either the `clone()` system call and SysV IPC, or glibc-2.1 pthreads.



# SDL in embedded systems

- ▶ Should work on any platform supporting Linux.
- ▶ Even runs on lightweight devices without X11.  
A framebuffer driver is sufficient (using the DirectFB API).
- ▶ Useful to develop your application on the host in parallel with target system development.
- ▶ Facilities to support the target endianness.





# Additional SDL libraries

- ▶ **SDL\_image**: [http://libsdl.org/projects/SDL\\_image/](http://libsdl.org/projects/SDL_image/)  
Image loading library supporting JPEG, PNG, GIF, BMP...
- ▶ **SDL\_ttf**: [http://libsdl.org/projects/SDL\\_ttf/](http://libsdl.org/projects/SDL_ttf/)  
TrueType font support library
- ▶ **SDL\_mixer**: [http://libsdl.org/projects/SDL\\_mixer/](http://libsdl.org/projects/SDL_mixer/)  
Sample multi-channel audio mixer library. Supports Ogg, MP3, MIDI.
- ▶ **SDL\_gfx**: [http://freshmeat.net/projects/sdl\\_gfx/](http://freshmeat.net/projects/sdl_gfx/)  
SDL graphics drawing primitives and other support functions.
- ▶ **SAGl**: <http://freshmeat.net/projects/sagl/>  
Small Anti-aliased graphics library.



# Other free SDL libraries

Listed on

<http://libsdl.org/libraries.php>

- ▶ Lots of libraries to ease the development of games, in particular game engines!
- ▶ Many graphics, GUI and OpenGL libraries.

- ▶ **Glyph Keeper**

TrueType font rendering library.

- ▶ **Fast Events**

A fast thread based replacement for SDL event routines.

- ▶ **Kyra** - A simple and powerful Sprite engine written in C++.



# Free SDL applications (1)

A list is available on <http://libsdl.org/applications.php>

- ▶ Very active development.
- ▶ Lots of games, of course!
- ▶ A significant number of system emulators.  
Example: QEMU (<http://qemu.org>).
- ▶ Some multimedia visualization plugins (e.g. for xmms).
- ▶ Open Picture Space 2  
<http://membres.lycos.fr/mavati/openpicturespace2/>  
Shows pictures in a directory through thumbnails.

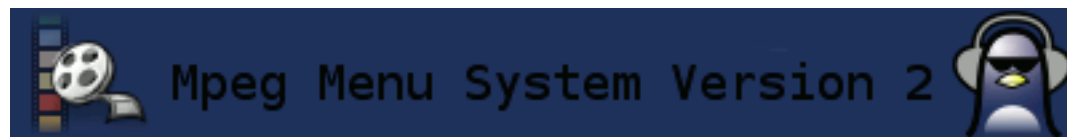


# Free SDL applications (2)

## MPEG Menu System Version 2

<http://mms.sunsite.dk/>

- ▶ Menu system for easy movie, audio and image playback. Electronic TV program guide.
- ▶ Supports MP3, OGG, DivX, XviD, VCD, SVCD, DVD, JPEG, PNG, and GIF
- ▶ Controlled using a remote control or keyboard





# Getting started with SDL - Hello World

```
#include "SDL.h"

int main( int argc, char* argv[] )
{
    /* initialize SDL */
    SDL_Init(SDL_INIT_VIDEO);

    /* set the title bar */
    SDL_WM_SetCaption("SDL Test", "SDL Test");

    /* create window */
    SDL_Surface* screen = SDL_SetVideoMode(640, 480, 0, 0);

    /* load bitmap to temp surface */
    SDL_Surface* temp = SDL_LoadBMP("hello.bmp");

    /* convert bitmap to display format */
    SDL_Surface* bg = SDL_DisplayFormat(temp);

    /* free the temp surface */
    SDL_FreeSurface(temp);

    SDL_Event event;
    int gameover = 0;

    /* message pump */
    while (!gameover)
    {
        /* look for an event */
        if (SDL_PollEvent(&event)) {
            /* an event was found */
            switch (event.type) {

                /* close button clicked */
                case SDL_QUIT:
                    gameover = 1;
                    break;

                /* handle the keyboard */
                case SDL_KEYDOWN:
                    switch (event.key.keysym.sym) {
                        case SDLK_ESCAPE:
                        case SDLK_q:
                            gameover = 1;
                            break;
                    }
                    break;
            }
        }

        /* draw the background */
        SDL_BlitSurface(bg, NULL, screen, NULL);

        /* update the screen */
        SDL_UpdateRect(screen, 0, 0, 0, 0);
    }

    /* free the background surface */
    SDL_FreeSurface(bg);

    /* cleanup SDL */
    SDL_Quit();

    return 0;
}
```

Thanks to Tony (<http://tonyandpaige.com>) for the example!



# Getting started with SDL (2)

## Compiling

- ▶ Easy to do with `sdl-config`:

```
gcc `sdl-config --cflags --libs` \ hello.c -  
o hello
```

## More examples

- ▶ Plenty of code examples available on <http://www.libsdl.org/demos.php> (check the licenses).



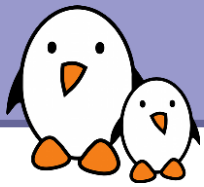
# Useful reading about SDL

- ▶ 2 nice introductions to the different parts of the API through simple examples.
  - ▶ SDL Introduction, by Sam Lantinga  
<http://www.libsdl.org/intro.php>
  - ▶ SDL Library Documentation  
<http://www.libsdl.org/cgi/docwiki.cgi/>
- ▶ 3 nice SDL tutorials with simple examples  
<http://andrew.textux.com/Articles.html>



## Multimedia libraries DirectFB





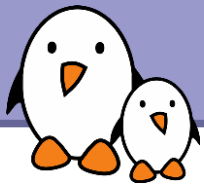
# DirectFB

<http://www.directfb.org/>

License: LGPL



- ▶ User space library developed for the special requirements of embedded devices. Small, powerful, flexible and easy to use.
- ▶ Hardware accelerated graphics support. Software fallbacks for operation not supported by the underlying hardware.
- ▶ Input device handling and abstraction.
- ▶ Also supports SDL (when `/dev/fb0` is not available).
- ▶ Windowing system with support for translucent windows.
- ▶ Multiple layer support.



# DirectFB screenshot



Demo of the new `TextureTriangles()` method using existing windows as textures (updated in real time)



# DirectFB in embedded systems

Extra value in embedded systems

- ▶ No need to run a X server using disk space and CPU.
- ▶ Lightweight: very few graphics libraries needed.
- ▶ Your interfaces can be very simple. They don't have to have the look and feel of desktop applications.
- ▶ Portability: you can develop your application on the PC host while the target is still under development.

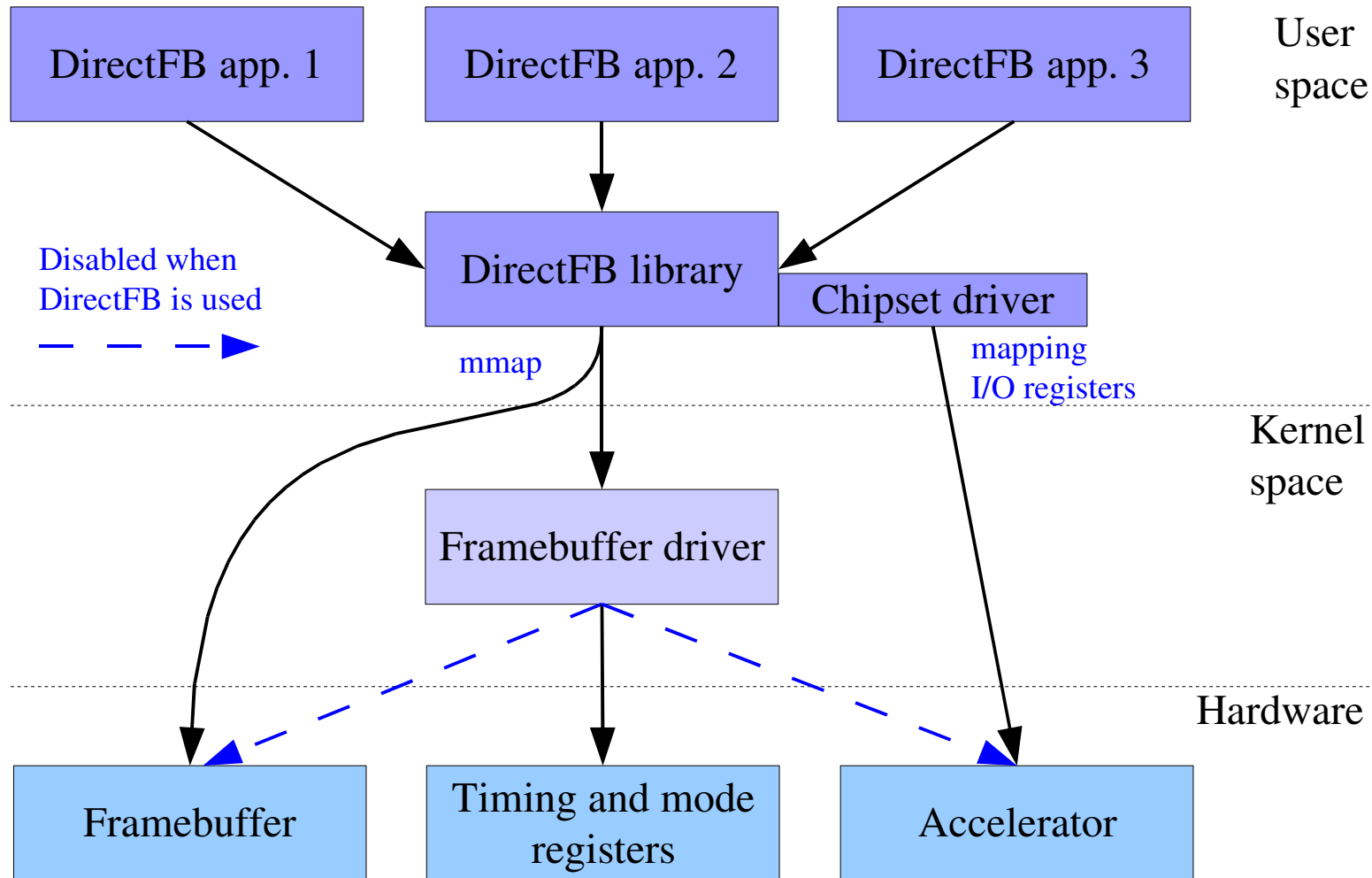


# DirectFB graphical capabilities

- ▶ Rectangle filling/drawing.
- ▶ Triangle filling/drawing.
- ▶ Line drawing.
- ▶ Flat shaded triangles.
- ▶ Simple blitting.
- ▶ Stretched blitting.
- ▶ Textured triangles (perspective correct).
- ▶ Blending with an alpha-channel (per pixel alpha).
- ▶ Blending with an alpha factor (alpha modulation).
- ▶ Nine source and destination blend functions.
- ▶ Porter/Duff image composition rules support.
- ▶ Pre-multiplied alpha supported.
- ▶ Colorized blitting (color modulation).
- ▶ Source color keying.
- ▶ Destination color keying.



# DirectFB operation





# DirectFB implementation

- ▶ Writing to the framebuffer: directly writes to the memory mapped graphics device video RAM.
- ▶ Accessing hardware acceleration controls: directly writes to the memory mapped graphics device I/O ports.



# Input device support

- ▶ Standard keyboards.
- ▶ Serial and PS/2 mice.
- ▶ Joysticks.
- ▶ Devices using the Linux input layer (many!).
- ▶ Infrared remote controls (using `lirc`).
- ▶ Some touchscreens.



# DirectFB image loading

The library includes image providers, which allow to load images directly into DirectFB surfaces:

- ▶ JPEG (using `libjpeg`).
- ▶ PNG (using `libpng2`).
- ▶ GIF (built-in).
- ▶ Various other image formats (using `imlib2`).





# DirectFB video playback

Includes different video providers,  
for rendering in DirectFB surfaces:

- ▶ V4L (Video4Linux) and V4L2 (`/dev/video`).
- ▶ mpeg1/2 (using `libmpeg3`).
- ▶ AVI (using `avifile`).
- ▶ MOV (using `OpenQuicktime`).
- ▶ Flash (using `libflash`).



# DirectFB font rendering

Supports anti-aliased text,  
and supports font loading through 2 providers:

- ▶ DirectFB bitmap font.
- ▶ TrueType (using FreeType2).



# DirectFB API (1)

- ▶ Native API available in C.
- ▶ 13 *interfaces* or *providers*: structures with function pointers.

Example:

<b><i>Methods of IDirectFBVideoProvider</i></b>		
Retrieving information	GetCapabilities	Retrieve information about the video provider's capabilities.
	GetSurfaceDescription	Get a surface description that best matches the video contained in the file.
Playback	PlayTo	Play the video rendering it into the specified rectangle of the destination surface.
	Stop	Stop rendering into the destination surface.
Media Control	SeekTo	Seeks to a position within the stream.
	GetPos	Gets current position within the stream.
	GetLength	Gets the length of the stream.
Color Adjustment	GetColorAdjustment	Gets the current video color settings.
	SetColorAdjustment	Adjusts the video colors.



# DirectFB API (2)

More than 120 *types*: data structures. Example:

▶ DFBTriangle

```
int    x1;    X coordinate of first edge
int    y1;    Y coordinate of first edge
int    x2;    X coordinate of second edge
int    y2;    Y coordinate of second edge
int    x3;    X coordinate of third edge
int    y3;    Y coordinate of third edge
```

▶ DFBScreenEncoderTVStandards

```
DSETV_UNKNOWN    Unknown standard
DSETV_PAL         PAL
DSETV_NTSC       NTSC
DSETV_SECAM      SECAM
```



# DirectFB API (3)

- ▶ *9 definitions*: macros to simplify code. Examples:
  - ▶ `DFB_COLOR_EQUAL (x, y)`  
Macro to compare two colors.
  - ▶ `DFB_DISPLAYLAYER_IDS_MAX`  
Maximum number of layer ids.
  - ▶ `DFB_DISPLAYLAYER_IDS_ADD (ids, id)`  
Adds the id to the bitmask of layer ids.
- ▶ Full API documentation available on  
[http://directfb.org/docs/DirectFB\\_Reference/](http://directfb.org/docs/DirectFB_Reference/)



# DirectFB API (4)

- ▶ Only a few global functions

  - ▶ `DirectFBInit(argc, argv)`

  - ▶ `DirectFBCreate(IDirectFB **dfb);`

- ▶ Then, through the `IDirectFB` interface, it is possible to work on surfaces, input devices, display layers, screens, media files, etc.

- ▶ For example, the `dfb->CreateSurface()` method will return a `IDirectFBSurface` interface pointer

- ▶ The `IdirectFBSurface` then contains methods to manipulate the surface: flipping, blitting, drawing, etc.



# DirectFB Hello World

```
DirectFBInit(&argc, &argv);
DirectFBCreate(&dfb);

dfb->SetCooperativeLevel(dfb, DFSCCL_FULLSCREEN);

memset(&dsc, 0, sizeof(DFBSurfaceDescription));
dsc.flags = DSDESC_CAPS;
dsc.caps = DSCAPS_PRIMARY;
dfb->CreateSurface(dfb, &dsc, &primary);

dfb->CreateImageProvider(dfb, "test.png", &provider);
provider->RenderTo(provider, primary, NULL);
provider->Release(provider);

sleep(5);

primary->Release(primary);
dfb->Release(dfb);
```



# Compiling a DirectFB program

That's very easy to do thanks to `pkg-config`:

```
gcc `pkg-config --cflags directfb`  
    -o test test.c  
    `pkg-config --libs directfb`
```

`pkg-config` is used to query needed C library headers and shared objects.





# Fusion

Makes it possible for multiple DirectFB applications to run simultaneously.

- ▶ Layer used by DirectFB applications to communicate.
- ▶ Previously implemented completely in user space using semaphores and message queues..
- ▶ Now critical parts of Fusion are implemented in kernel space (“Fusion Kernel Device”), which provides better performance and stability. Supports both Linux 2.4 and 2.6.
- ▶ Linux kernel patch available in the Core downloads page (`linux-fusion` module).



# A few DirectFB applications

- ▶ ByzantineOS, GeexBox  
Home entertainment
- ▶ DBFSee  
Image viewer and video player
- ▶ DFBTerm  
Terminal application
- ▶ XDirectFB  
Rootless X Server using  
DirectFB windows. Useful to  
run any X windows app.

Applications with DirectFB back-ends

- ▶ gdk-directfb  
DirectFB GDK for GTK
- ▶ MPlayer  
DirectFB back-end since 2001.
- ▶ SDL  
Also has a DirectFB back-end!

More on <http://directfb.org>!



# Useful reading about DirectFB

- ▶ DirectFB overview

[http://www.directfb.org/docs/DirectFB\\_overview\\_V0.2.pdf](http://www.directfb.org/docs/DirectFB_overview_V0.2.pdf)

A nice and synthetic article!



## Kernel subsystems Video4linux



# V4L: video4linux subsystem

Provides support for audio/video capture and overlay devices and FM radio

- ▶ **V4L**: Original video capture/overlay API of the Linux kernel. Appeared in the 2.1.x development cycle of the Linux kernel.
- ▶ **V4L2**: Second generation of the video4linux API fixing a number of design bugs. Integrated into the standard kernel in 2.5.x. Not supported yet by all drivers.

Main web pages:

- ▶ Video for Linux resources: <http://www.exploits.org/v4l>
- ▶ video4linux HQ: <http://linux.bytesex.org/v4l2/>



# v4l original API

- ▶ **API:** `Documentation/video4linux/API.html` in the kernel sources (even in Linux 2.6).
- ▶ **Device files**
  - ▶ `/dev/video` (minor 0-63): Video Capture Interface  
Of course, can read video input from several of these devices:  
`/dev/video0, /dev/video1...`
  - ▶ `/dev/radio` (minor 64-127): AM/FM Radio Devices
  - ▶ `/dev/vtx` (minor 192-223): Teletext Interface Chips
  - ▶ `/dev/vbi` (minor 224-239): Raw VBI Data (Intercast/teletext)



# v4l2 new API

Also on [Documentation/video4linux/API.html](http://Documentation/video4linux/API.html)

- ▶ Linux 2.4 no longer supported by V4L2.
- ▶ Linux 2.6 ships the latest V4L2 versions  
Development CVS snapshots available on  
<http://dl.bytesex.org/cvs-snapshots/>.
- ▶ Offers backward compatibility with V4L:  
true for V4L applications, not for V4L drivers.



# V4L2 API specification

New API specification: <http://v4l2spec.bytesex.org/spec/>  
By Michael H. Schimek and Bill Dirks

- ▶ A very exhaustive document (Oct 2007: 280 pages!)
- ▶ Specifies all details about the interface a driver should implement. Lots of code examples.
- ▶ Provides details about application programming too.
- ▶ A lot of technical background provided too!

Read this like a reference book!





# A few V4L2 features

- ▶ **Devices**: Uses the same device files, but it's up to the system administrator to assign minor numbers.
- ▶ **Multiple opens**: *may* support concurrent access to `/dev/` files, for example to control brightness or color (panel application) while another application is playing the video.
- ▶ **Standard controls**: the API defines hardware independent device controls (brightness, color saturation, etc), and a way to enumerate available controls. Applications can be driver independent.



# Communicating with ioctl (1)

A few details about the `ioctl()` system call are needed to understand how a video4linux client application can interact with a video4linux driver.

Prototype:

```
#include <sys/ioctl.h>
int ioctl(int fd, int request, void *argp);
```

- ▶ `fd`: file descriptor (the opened file)
- ▶ `request`: an ioctl number, supported by the driver
- ▶ `argp`: argument string or pointer to a value or structure returned by the driver.



# Communicating with ioctl (2)

`ioctl` is used to interact with the driver handling the opened device file, without having to read or write data.

- ▶ To access a particular setting of the driver  
Example: getting the serial number of the underlying hardware.
- ▶ To set a particular driver setting  
Example: setting the speed and flow control of a serial port.
- ▶ To have the driver perform a special operation  
Example: formatting a floppy disk.

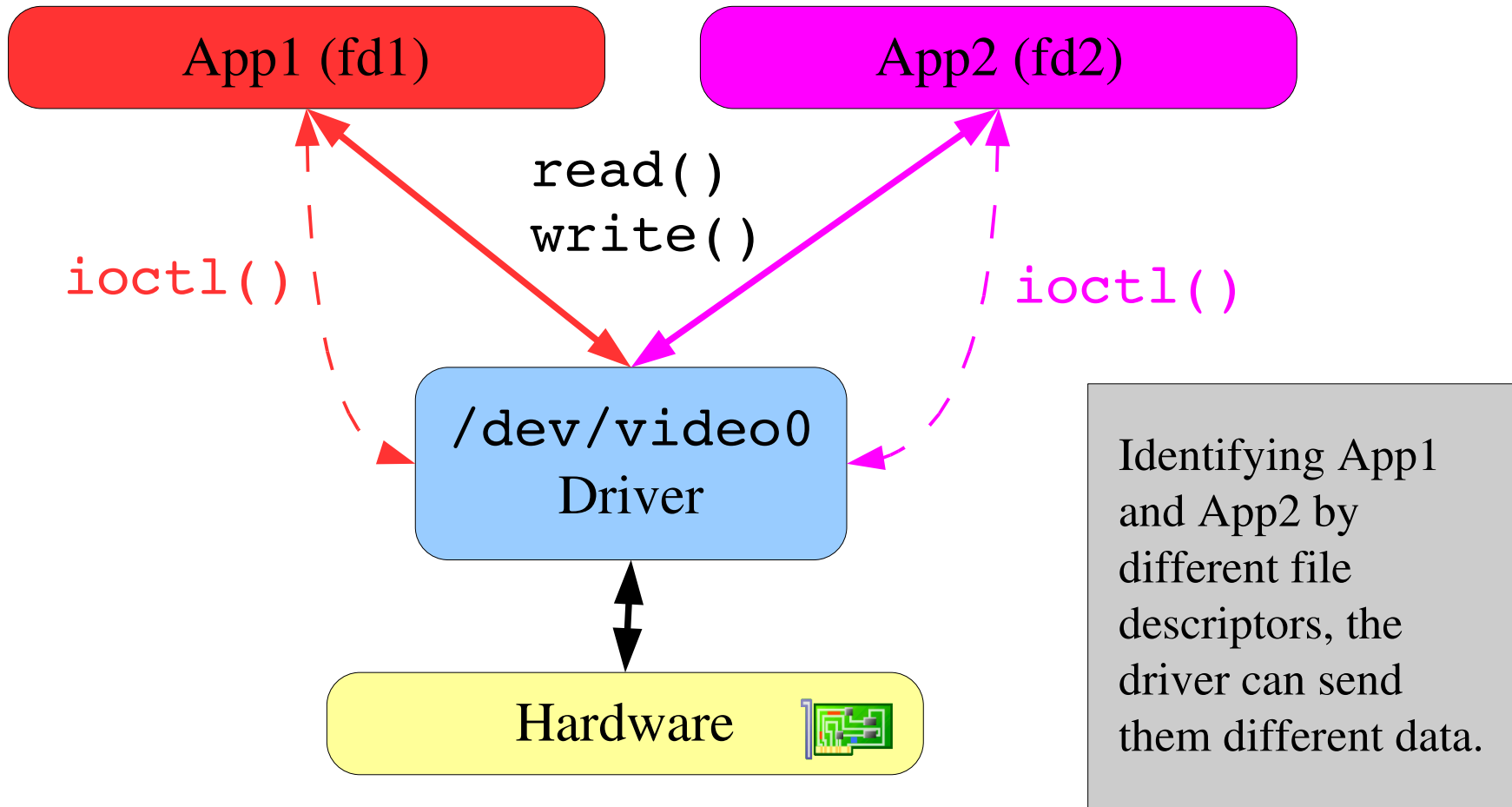


# A few V4L2 ioctl number examples

- ▶ `VIDIOC_QUERYCAP`  
Querying drivers capabilities
- ▶ `VIDIOC_ENUMINPUT`  
Querying the number, types and names of video connectors.
- ▶ `VIDIOC_G_FREQUENCY`  
Getting the tuner frequency
- ▶ `VIDIOC_S_FREQUENCY`  
Setting the tuner frequency
- ▶ `VIDIOC_ENUMSTD`  
Query supported video standards (PAL, NTSC...)
- ▶ `VIDIOC_QUERYCTRL`  
Query available controls
- ▶ `VIDIOC_S_FMT`  
Setting the image format (including colorspace info)
- ▶ `VIDIOC_REQBUFS`  
Asking the driver to allocate a buffer for memory mapped I/O



# Concurrent access to /dev/video0





# I/O modes

Several I/O modes can be proposed by the driver

- ▶ `read()` and `write()`

Standard way of accessing data. Basic but can still be efficient (can use DMA internally). But can't carry meta information like frame counters or timestamps.

- ▶ Streaming I/O (memory mapping)

Sharing pointers to memory mapped buffers (`mmap`), allocated by the driver. No data copying. Buffers can't be swapped out to disk.

- ▶ Streaming I/O (user pointers)

Sharing pointers to application allocated memory buffers. Buffers can reside in shared or virtual memory (can be swapped out).



## V4L

- ▶ Lots of tools still seem to support only V4L.  
Fortunately, the V4L2 high-level driver comes with a backward compatibility layer which translates V4L ioctl commands to V4L2 ones
- ▶ Listed on <http://www.exploits.org/v4l>

## V4L2

- ▶ MPlayer (plugin)
- ▶ DirectFB  
Supports V4L2 devices as video providers for the display
- ▶ More listed on <http://www.thedirks.org/v4l2/peopleprojects.htm>



# xawtv toolset (1)

A set of V4L2 compatible tools for watching TV, recording video, browsing teletext... <http://linux.bytesex.org/xawtv/>

## X11 applications

- ▶ `xawtv`  
Watching TV
- ▶ `motv`  
Motif based TV GUI
- ▶ `mtt`  
Text or Motif teletext browser
- ▶ `pia`  
Simple movie player

## Command line / console tools

- ▶ `fbtv`  
Framebuffer TV application
- ▶ `streamer`  
Recording movies
- ▶ `v4lctl`  
Setting V4L2 parameters





# xawtv toolset (2)

## Console / command line tools

- ▶ `scantv`  
Channel scanner
- ▶ `radio`  
Simple, curses based radio application.
- ▶ `webcam`  
Captures images and uploads them to a Web-Server
- ▶ `alevtd`  
http server for videotext pages

## Fonts

<http://linux.bytesex.org/xawtv/tvfonts.htm>

- ▶ `led-fixed`  
Big font used by xawtv for on-screen display.
- ▶ `ets-teletext`  
Teletext font. Block graphic characters.
- ▶ `ets-caption`  
Font for closed captions (audio subtitles and scene description).



# Writing to screens

Video is best when you can watch it!

- ▶ Need to choose an existing application displaying video
- ▶ Or create your own using a given multimedia capable library:  
DirectFB, SDL...
- ▶ 2 ways of accessing the graphics:  
X server or framebuffer based  
(mainly DirectFB or Qtopia Core)
- ▶ Anyway, you most probably won't have to access  
the display by yourself, but through the low level library.
- ▶ Supporting video codecs in a custom application.  
Best to rely on gstreamer: <http://gstreamer.freedesktop.org/>



## Kernel subsystems Digital Video Broadcasting



# Digital Video Broadcasting

DVB: Digital Video Broadcasting

DVB Project: <http://www.dvb.org/>



- ▶ Since 1993, industry-led consortium of broadcasters, manufacturers, network operators, software developers, regulatory bodies and others interested in standards for the delivery of any digital television and data services.
- ▶ Committed to open standards  
Standard specifications supposed to be available free of charge (registration required) from ETSI (European Telecommunication Standards Institute) at <http://www.etsi.org/>.
- ▶ Based on MPEG2 transport streams.



# LinuxTV project

<http://linuxtv.org/>



- ▶ LinuxTV is a vendor independent, non profit Linux project that works on a standardized Linux DVB API since 2000.
- ▶ Version 3 of the Linux DVB API is included in Linux 2.6.
- ▶ Very popular on PC systems, mostly in Europe and in Australia.
- ▶ Used by lots of Free Software projects and commercial Set Top Boxes on different hardware platforms.



## Free Software Video

### Video codecs and file formats



# RealVideo

From RealNetworks

<http://realnetworks.com/>



- ▶ Proprietary format
- ▶ Originally used for low bitrate streaming
- ▶ Supports streaming through the Real-time Transport Protocol (RTP)

<http://en.wikipedia.org/wiki/RealVideo>

- Free Software decoders available (mplayer)
- Supported by some mobile phones.
- Mainly used for streaming, used by a significant number of on-line media.
- Only proprietary encoders. RealNetworks encoder free of charge only for personal use.



## Windows Media Video

- ▶ Proprietary video codec: Microsoft's non-standard MPEG4 implementation.
- ▶ Often comes with WMA for Audio.
- ▶ Packed into AVI or Advanced Streaming Format (ASF) container.
- ▶ Supports DRM encryption.
- ▶ Now many hardware players available.
- Not many files owned by users (prefer DivX or XviD).
- However, more and more on-line streams available in this format. Customers are likely to expect support for this format in a hardware player.
- Lack of Free Software players (except libavcodec) and encoders.
- Relies on patented algorithms.
- Licenses may apply to encoding, decoding or even songs, though MS is still very tolerant so far (to achieve domination).





# QuickTime

From Apple Computer

<http://apple.com/quicktime/>



- ▶ Multimedia technology supporting several digital video formats, sound, text, animation, music, and virtual reality panoramic images.
- ▶ Open file format  
1998: Approved by ISO as the basis for the MPEG4 standard
- ▶ MPEG4 compatibility since 2002.

- ▶ Video often encoded with the Sorenson proprietary codec.
- Few users have QuickTime videos and will ask for QuickTime support.
- No Free Software encoders  
Sorenson patent license would be needed anyway in some countries.
- Just 1 Free Software decoder (FFmpeg's libavcodec).

More on

<http://en.wikipedia.org/wiki/Quicktime>



# DivX

From DivXNetworks

<http://divxnetworks.com/>



- ▶ DivX 3.11: hacked version of the Microsoft MPEG 4 layer 2 codec, which made the codec popular.
- ▶ Able to shrink a 5-6 GB DVD video to a file fitting in a CDROM (700 MB) with minimum quality loss.
- ▶ DivX 4 and later: clean room implementation from DivXNetworks, to avoid issues with Microsoft copyright.
- ▶ DivX 4: MPEG 4 layer 2 compliant
  - Free Software players available.
  - Supported by most hardware video players too.
  - Still burdened by MPEG 4 patents. In some countries, license needed to implement or use a software or hardware player.

See <http://en.wikipedia.org/wiki/DivX>



# XviD

From XviD.org  
<http://xvid.org/>

- ▶ Free Software codec, MPEG 4 based, competing with DivX.
- ▶ Started from DivXNetworks OpenDivX, after access to its sources got closed.
- ▶ License: GNU GPL.



- Many hardware players.
- Many Free Software players.
- Still burdened by MPEG 4 patents. In some countries, license needed to implement or use a software or hardware player.

See

<http://en.wikipedia.org/wiki/XviD>



# Ogg Theora

From Xiph.org

<http://theora.org/>

- ▶ Ogg:container for multimedia streams.
- ▶ Theora:video codec competing with MPEG4, RealVideo and WMV.
- ▶ Derived from On2 Technologies VP3 codec. On2 irrevocably granted to anyone a royalty-free license to use all its VP3 patents.

- ▶ Vorbis or Flac: sound layer
- Theora library available under a BSD-style license.
- Theora already supported by various software players.
- Theora library still in beta stage, but getting close to a stable release. Anyway, the bitstream format is frozen and will be supported by all future versions.
- Very rare hardware players



## Multimedia distributions



# GeexBox



<http://geebox.org/>

- ▶ MPlayer based full featured home multimedia system: Multimedia (DVD, DivX, CD, Ogg, MP3) and photo gallery.
- ▶ Available through a live CD: boot your computer with it, eject the CD, and load whatever media you want to read.
- ▶ Size: 6 MB.  
Great for learning about a embedded Linux media player!
- ▶ Graphics: DirectFB.
- ▶ Supported platforms: [i386](#) and [ppc](#).  
Actively maintained.





# MythTV

<http://mythtv.org/>

- ▶ Set of Free Software applications to turn a PC into a digital video recorder / digital multimedia home entertainment system. Based on the QT library.
- ▶ Requires TV tuner cards supported by the video4linux(2) kernel drivers.
- ▶ Lots of great features: commercial skipping, access to free TV listings, DVD burning, web browser, RSS news reader, IP telephony...
- ▶ Can be tried through the KnoppMyth live CD:  
<http://en.wikipedia.org/wiki/Mythtv>.

See <http://en.wikipedia.org/wiki/Mythtv> for details.





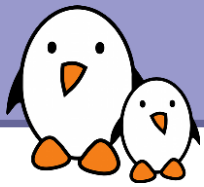
# FreeVo

<http://freevo.sourceforge.net/>

- ▶ Another PC based digital multimedia home entertainment system / personal video recorder.
- ▶ Supposed to be easy to install for new users.
- ▶ Based on the SDL library.







# Related documents

**Free Electrons**  
Embedded Freedom

HOME DEVELOPMENT SERVICES TRAINING DOCS COMMUNITY COMPANY BLOG

**Recent blog posts**

- ELC Europe in Grenoble
- Free Electrons at ELC
- Linux kernel 2.6.29 - New features for embedded users
- The Buildroot project begins a new life
- FOSDEM 2009 videos
- USB-Ethernet device for Linux
- Program for Embedded Linux Conference 2009 announced
- Public session changes
- Real hardware in our training sessions
- Call for presentations for the LSM embedded track

**Docs**

Most of the below documents are presentations used in our [training sessions](#), or in technical conferences.

**License**

All our documents are available under the terms of the [Creative Commons Attribution-ShareAlike 3.0 license](#). This essentially means that you are free to download, distribute and even modify them, provided you mention us as the original authors and that you share these documents under the same conditions.

**Linux kernel**

- [Embedded Linux kernel and driver development](#)
- [New features in Linux 2.6](#) (since 2.6.10)
- [Kernel initialization](#)
- [Porting Linux to new hardware](#)
- [Power management in Linux](#)
- [Linux PCI drivers](#)
- [Block device drivers](#)
- [Linux USB drivers](#)
- [DMA](#)

**Architecture specific documents**

- [ARM Linux specifics](#)
- [Linux on TI OMAP processors](#)

**Embedded Linux system development**

- [Embedded Linux system development](#)
- [Real time in embedded Linux systems](#)
- [Block filesystems](#)
- [Flash filesystems](#)
- [Free software development tools](#)
- [The U-boot bootloader](#)
- [The GRUB bootloader](#)
- [The blob bootloader](#)
- [Hotplugging with udev](#)
- [Introduction to uClinux](#)
- [Java in embedded Linux](#)
- [Embedded Linux optimizations](#)
- [Audio in embedded Linux systems](#)
- [Multimedia in embedded Linux systems](#)
- [Embedded Linux From Scratch... in 40 minutes!](#)
- [Building embedded Linux systems with Buildroot](#)
- [Developing embedded distributions with OpenEmbedded](#)
- [The Scratchbox development environment](#)

**Miscellaneous**

- [Introduction to the Unix command line](#)
- [SSH](#)
- [Linux virtualization solutions \(with an embedded perspective\)](#)
- [Advantages of Free Software and Open Source in embedded systems](#)
- [Introduction to GNU/Linux and Free Software](#)

All our technical presentations on <http://free-electrons.com/docs>

- ▶ Linux kernel
- ▶ Device drivers
- ▶ Architecture specifics
- ▶ Embedded Linux system development



# How to help

You can help us to improve and maintain this document...

- ▶ By sending corrections, suggestions, contributions and translations
- ▶ By asking your organization to order development, consulting and training services performed by the authors of these documents (see <http://free-electrons.com/>).
- ▶ By sharing this document with your friends, colleagues and with the local Free Software community.
- ▶ By adding links on your website to our on-line materials, to increase their visibility in search engine results.

## Linux kernel

- Linux device drivers
- Board support code
- Mainstreaming kernel code
- Kernel debugging

## Embedded Linux Training

***All materials released with a free license!***

- Unix and GNU/Linux basics
- Linux kernel and drivers development
- Real-time Linux, uClinux
- Development and profiling tools
- Lightweight tools for embedded systems
- Root filesystem creation
- Audio and multimedia
- System optimization

# Free Electrons

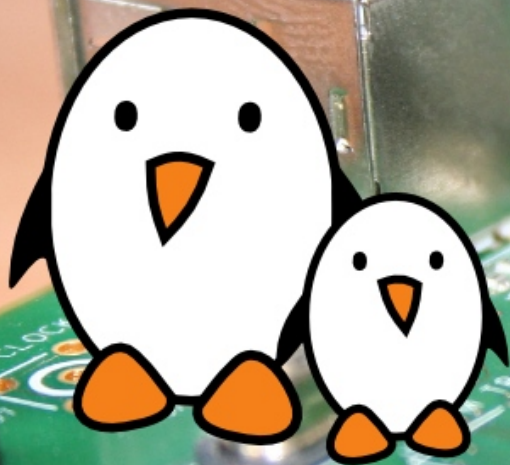
## Our services

### Custom Development

- System integration
- Embedded Linux demos and prototypes
- System optimization
- Application and interface development

### Consulting and technical support

- Help in decision making
- System architecture
- System design and performance review
- Development tool and application support
- Investigating issues and fixing tool bugs



**Free Electrons**  
Embedded Linux Experts

<http://free-electrons.com>