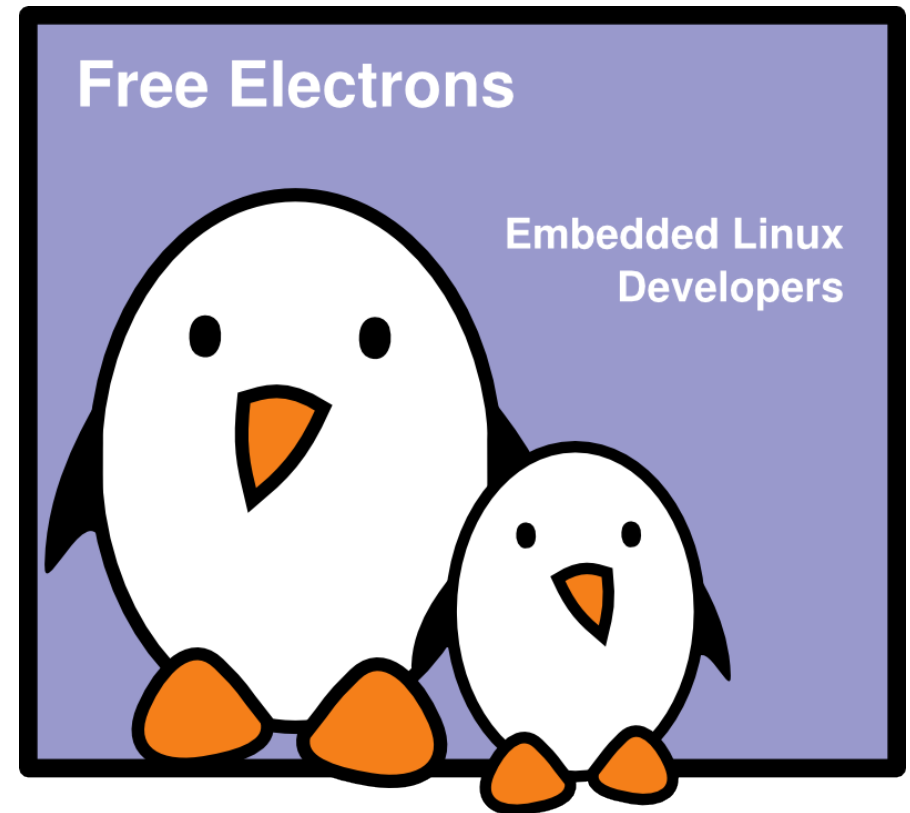




## Using USB gadget drivers

Thomas Petazzoni  
**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/block-drivers>

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>



# What are USB gadgets ?

- ▶ The USB controller found in most of our PCs can only act as hosts on a USB bus, they can “drive” devices, but not be “driven”
- ▶ However, the USB controller found on many embedded systems can often act either as a host, as a device, or as both.
- ▶ Having a device acting as a USB device allows to easily connect it to other systems, using the now widespread USB connection.
- ▶ This operation mode is called “USB gadget” by Linux
- ▶ Several USB gadget drivers are already available in Linux
  - ▶ Ethernet
  - ▶ Serial
  - ▶ Mass storage
  - ▶ MIDI
  - ▶ Printer



# Kernel requirements

- ▶ USB Gadget generic support  
Kernel option `CONFIG_USB_GADGET`
- ▶ USB Gadget controller support
  - ▶ The controller support for USB gadget is different from the USB host support
  - ▶ Depending on your hardware and the available Linux support, might need to be implemented
- ▶ USB Gadget driver
  - ▶ They are independent from the hardware
  - ▶ `CONFIG_USB_ETH` for Ethernet, `CONFIG_USB_FILE_STORAGE` for mass storage, `CONFIG_USB_G_SERIAL` for serial
- ▶ See <http://www.linux-usb.org/gadget/> for details



# Ethernet gadget driver on the device

- ▶ Kernel option `CONFIG_USB_ETH`
- ▶ This driver allows to establish an Ethernet connection between the embedded system (USB device) and the host system (USB master)
- ▶ When compiled as a module, the module name is `g_ether`.
- ▶ On the embedded system, a `usbX` network interface appears and can be configured as usual



# Ethernet gadget on the host

## ▶ Linux

- ▶ The usbnet driver included in the kernel just works
- ▶ A usbX interface appears and can be configured as usual

## ▶ Windows

- ▶ Versions prior to Vista, no support for the Ethernet CDC model. Linux provides the `CONFIG_USB_ETH_RNDIS` option to support a protocol that Windows understands, and provides the corresponding `.inf` file.
- ▶ Versions starting at Vista, CDC support is available if the vendor provides a proper `.inf` file

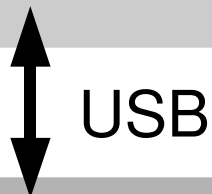


# Ethernet gadget at work

```
root@om-gta02:~# ifconfig usb0
```

```
usb0      Link encap:Ethernet  HWaddr 3E:04:A8:D5:9E:7E
          inet addr:192.168.0.202  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          [...]
```

Target



```
thomas@surf:~$ sudo ifconfig usb0
```

```
usb0      Link encap:Ethernet  HWaddr be:26:39:84:59:80
          inet addr:192.168.0.200  Bcast:192.168.0.223  Mask:255.255.255.224
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          [...]
```

Host

```
thomas@surf:~$ ping -c 1 192.168.0.202
```

```
PING 192.168.0.202 (192.168.0.202) 56(84) bytes of data.
64 bytes from 192.168.0.202: icmp_seq=1 ttl=64 time=1.80 ms
```

```
--- 192.168.0.202 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.804/1.804/1.804/0.000 ms
```



# USB storage gadget

- ▶ Kernel option `CONFIG_USB_FILE_STORAGE`
- ▶ Allows to make a file or block device available on the device through the well-known mass-storage USB protocol
- ▶ Compiled as the `g_file_storage` module
- ▶ The `file=` option allows to specify the file or the block device containing the data to export
  - ▶ Can be passed as module parameter
  - ▶ Can be passed through the `sysfs` filesystem
  - ▶ Contents of the file cannot be modified during runtime !
- ▶ Other options such as `ro` (read-only) or `removable` are also available.
- ▶ See [http://www.linux-usb.org/gadget/file\\_storage.html](http://www.linux-usb.org/gadget/file_storage.html)



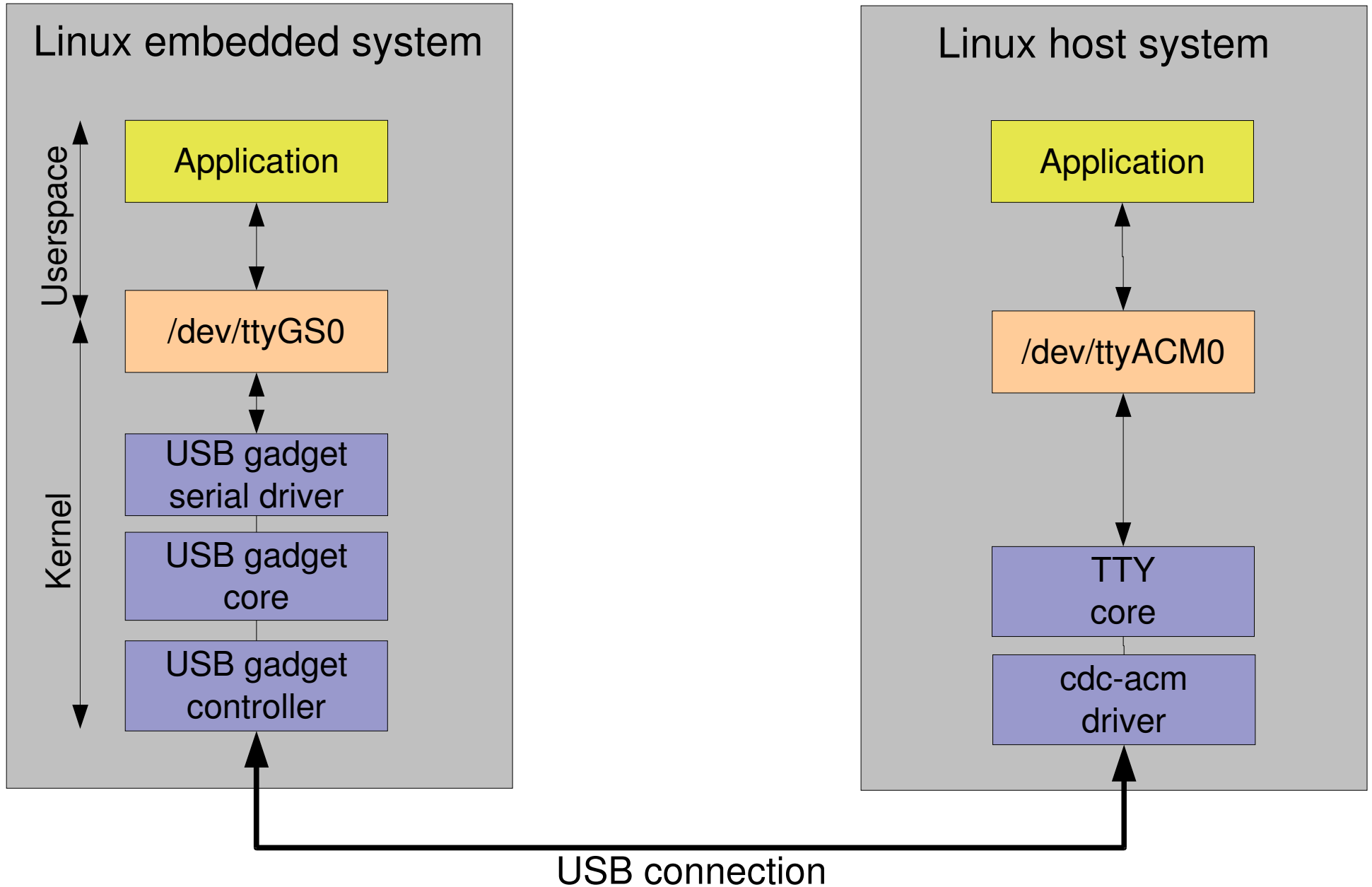


# USB serial gadget

- ▶ Kernel option `USB_G_SERIAL`
- ▶ Supports the standard CDC-ACM protocol
- ▶ On the device
  - ▶ The kernel module is `g_serial`
  - ▶ After loading, it will create a `/dev/ttyGS0` file which is one end of the serial communication.
  - ▶ If `udev/mdev` is not used, look at `/sys/class/tty/ttyGS0/dev` to know the major and minor
- ▶ On the host
  - ▶ Linux `cdc-acm` driver is included in the kernel. A `/dev/ttyACM0` device will appear, it is the other end of the serial communication.
  - ▶ For Windows, see `Documentation/usb/gadget_serial.txt`. A `.inf` file and a `.sys` file are needed.



# USB serial gadget architecture

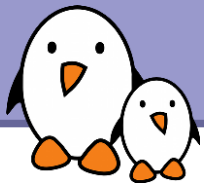




# Practical lab – USB gadget drivers



- ▶ Configure and compile a kernel with USB gadget support
- ▶ Set up USB gadget drivers on the Calao ARM board



# 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

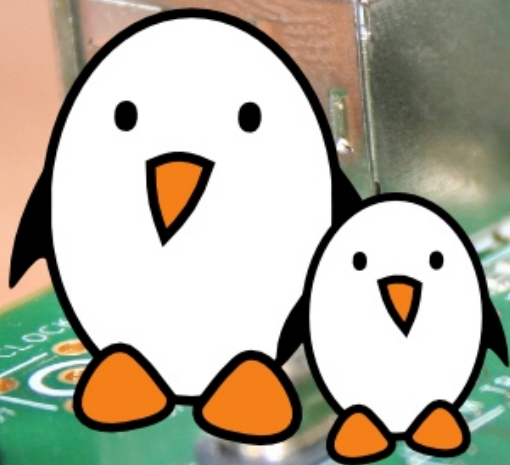
## Our services

### 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>