[redacted]

# Writing USB-2.0 and USB-3.0 host and gadget drivers on Linux

## Objectives

- Learn to write Linux drivers for USB-2.0 and USB-3.0
  - Explore the Linux USB host-driver stack
  - Learn the structure of USB device drivers
  - Discover USB gagdet drivers (2.0 and 3.0)
  - Understand the support for OTG-2.0 and OTG-3.0.
- Understand the specifics of the Linux kernel in the management of devices and drivers.
- Learn to configure the Linux kernel for optimal hotplug management.
  - Understand how hotplug events are generated and how to use them in drivers.
  - Install and use external hotplug daemons: udev, libusb, etc ...
- Discover Linux kernel changes up to the latest versions (up to 3.6.39 and 3.x).
- Master the techniques of kernel debugging.

***Labs are conducted using the System Workbench for Linux - Basic EditionIDE, for which all trainees will get a free license, so that they can continue to work, after the training, in a convenient and efficient environment.***

-->

## *Course Environment*

- Theoretical course
  - PDF course material (in English) supplemented by a printed version.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
    - One PC for two trainees when there are more than 6 trainees.
  - For onsite trainings:
    - An installation and test manual is provided to allow preinstallation of the needed software.
    - The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

## *Prerequisite*

- Good practice of C programming on Linux
- Good knowledge of Linux kernel and driver programming (see our D3 - Linux Driverscourse and D7 - Power Management in Linux Drivers courses)

## *Target Audience*

- Any embedded systems engineer or technician with the above prerequisites.

## Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

# Plan

## First day

## Reminders on kernel programming

- Reminders on kernel module development
- Kernel objects
  ***Exercise:*** *Writing a kernel module creating and using kernel objects and sets*
- The sysfs file system
  ***Exercise:*** *Interacting with a kernel module through a kernel object and the sysfs file sytem*

## Hotplug

- Hotplug in the kernel
- uevents
  ***Exercise:*** *Writing a kernel module sending hotplug events to a user mode program.*
- Hotplug at user level
- Udev
- Hal and Dbus
  ***Exercise:*** *Cross-compiling, configuring and using Udev.*

## Second Day

## Devices and Drivers

- The Device/Driver model in Linux
- Device class and types
- Bus drivers
- Bus types
- Generic devices and drivers
- System devices and drivers
- Platform devices and drivers
  ***Exercise:*** *Writing a platform device driver showing how device matching work*

## USB Drivers

- The USB bus
- USB devices
- USB descriptors

- USB endpoints
- USB requests
- User view of the USB bus and devices
- USB device drivers
- Hotplug
- Communicating with devices through URBs

***Exercise:*** *Writing a basic usb device driver using URBs*
***Exercise:*** *Writing an usb device driver using synchronous request management*

## ***Third Day***

### *The libUSB user-mode USB driver framework*

- The libUSB libraries.
- libUSB 0.1.12.
- libUSB 1.0

***Exercise:*** *Building libUSB*
***Exercise:*** *Writing a user-mode USB driver using libUSB*

### *USB gadget drivers*

- Basic USB gadgets.
- Composite USB gaddget drivers.

***Exercise:*** *Writing a gadget driver and the corresponding host driver on the Linux workstation.*

- The USB On-The-Go (OTG) specification.
- OTG support in Linux

## Renseignements pratiques

**Inquiry : 3 days**