

# KAÏNA-COM

## CATALOGUE DE FORMATION

### Développement logiciels Linux temps réel et embarqué

**Le cours pratique se concentre les aspects temps réel et embarqués de la programmation du noyau. Les travaux pratique en laboratoire font partie intégrante du cours.**



## KLI001 – Real Time and Embedded Linux Software Development

---

**Référence** KLI001

---

**Niveau**

- Débutant
- Intermédiaire
- Expert

---

**Nombre De Jours** Programme de Formation :

- 40 heures (4 heures/jour)

---

**Lieu De La Formation**

- I: i-learning, Formation individuelle (Formation en ligne)
- V: v-learning, classe virtuelle
- C: c-learning, cours présentiel

**KAINA-COM**

LE CARRÉ HAUSSMANN II,  
6 Allée de la Connaissance  
77127 Lieusaint - France

---

**Prérequis** Les participants doivent disposer d'une bonne connaissance d'Unix ou de Linux et d'une bonne maîtrise de la programmation en C. Bases de Linux, Introduction à Linux ou équivalent. Programmation de systèmes Linux ou équivalent. Une connaissance de base des pilotes (drivers) et des modules du noyau est essentielle.

Un niveau d'anglais business moyen est requise car la formation sera dispensé en anglais.

---

**Public** Programmeurs embarqués et temps réel développant des drivers à l'aide du noyau Linux et des développeurs de drivers pour les périphériques internes ou externes

---

*Ce sujet continue à la page suivante*



## **KLI001 – Real Time and Embedded Linux Software Development,** Suite

### **Objectifs**

Le système d'exploitation GNU / Linux est le système d'exploitation de choix pour de nombreux développeurs embarqués et temps réel: les principales raisons étant que le code source est gratuit, il n'y a pas de redevances d'exécution et c'est un système d'exploitation fiable et robuste avec un excellent support réseau. Le cours pratique se concentre les aspects temps réel et embarqués de la programmation du noyau. Les travaux pratique en laboratoire font partie intégrante du cours.

*Ce sujet continue à la page suivante*



## KLI001 – Real Time and Embedded Linux Software Development, Suite

### Contenu du cours

Contenu du cours :

Table 1: KLI001 - Contenu du cours

Chapter	Description
<b>Introduction</b>	<ul style="list-style-type: none"> <li>• Linux overview</li> <li>• Real time and embedded</li> <li>• The kernel and its role</li> <li>• Linux supported architectures</li> </ul>
<b>Build Root</b>	<ul style="list-style-type: none"> <li>• Project overview</li> <li>• Getting buildroot</li> <li>• Quick start</li> <li>• Configuration interfaces</li> <li>• Using a predefined configuration</li> </ul>
<b>Cross tool chains</b>	<ul style="list-style-type: none"> <li>• The need for cross tool chains</li> <li>• Tools naming convention</li> <li>• Getting and installing a tools chain</li> <li>• Cross building software</li> <li>• Cross debugging</li> <li>• uClibc</li> </ul>
<b>Configuring and Building the Linux Kernel</b>	<ul style="list-style-type: none"> <li>• Getting the sources</li> <li>• The structure of source tree</li> <li>• Configuring and building the kernel</li> <li>• Compiling the kernel</li> <li>• Kernel modules: <ul style="list-style-type: none"> <li>– Cross compiling modules</li> <li>– Integrating modules into the kernel source tree</li> </ul> </li> <li>• Configuring buildroot: <ul style="list-style-type: none"> <li>– Configuring the kernel in buildroot</li> </ul> </li> </ul>

*Ce sujet continue à la page suivante*



## KLI001 – Real Time and Embedded Linux Software Development, Suite

### Contenu du cours, Suite

Chapter	Description
<b>Customizing Buildroot</b>	<ul style="list-style-type: none"><li>• Integrating Additional packages into buildroot:<ul style="list-style-type: none"><li>– dl</li><li>– packages</li><li>– config.in</li></ul></li><li>• Overlays</li></ul>
<b>Device Trees</b>	<ul style="list-style-type: none"><li>• Working without device trees</li><li>• What is a device tree</li><li>• DTS and DTB</li><li>• Device tree integration into driver code.</li><li>• The syntax of DTS files</li></ul>
<b>Linux Boot Sequence</b>	<ul style="list-style-type: none"><li>• Embedded Linux boot process</li><li>• Kernel boot parameters</li><li>• Bootloaders, U-Boot</li><li>• Buildroot configuring system components:<ul style="list-style-type: none"><li>– init, busy box, U-Boot</li></ul></li><li>• root-fs:<ul style="list-style-type: none"><li>– initrd &amp; initramfs</li><li>– overlayfs</li></ul></li></ul>
<b>Net-Booting and The Network File System (NFS)</b>	<ul style="list-style-type: none"><li>• How does NFS aid the embedded development process</li><li>• Preparing NFS</li><li>• Mounting an NFS volume</li><li>• NFS daemons</li><li>• Exports file</li><li>• root-fs over NFS</li><li>• tftp</li><li>• DHCP</li></ul>

*Ce sujet continue à la page suivante*



## KLI001 – Real Time and Embedded Linux Software Development, Suite

### Contenu du cours, Suite

Chapter	Description
<b>User-mode Programming</b>	<ul style="list-style-type: none"><li>• librt overview</li><li>• Scheduling policies and priorities.</li><li>• CPU affinity</li><li>• Memory</li><li>• RT signals:<ul style="list-style-type: none"><li>– explanation</li><li>– comparison with standard signals</li></ul></li><li>• Asynchronous I/O</li><li>• POSIX IPC:<ul style="list-style-type: none"><li>– Semaphores</li><li>– Message Queues</li><li>– Shared memory</li></ul></li><li>• POSIX timers.</li><li>• Tips for improving user space RT performance.</li><li>• Command line tools for manipulating scheduling policy / priority, and CPU affinity</li></ul>

*Ce sujet continue à la page suivante*



## KLI001 – Real Time and Embedded Linux Software Development, Suite

### Contenu du cours, Suite

Chapter	Description
<b>Linux and Real Time</b>	<ul style="list-style-type: none"><li>• RTOS memory issues and Linux.</li><li>• Linux hardware interaction</li><li>• Latency (kernel, interrupt, scheduler)</li><li>• Kernel preemption</li><li>• Linux hard real time extensions</li><li>• Applying the RT patch</li><li>• Threaded IRQ's</li><li>• Voluntarily giving up CPU – cond_resched</li><li>• Controlling kernel preemption:<ul style="list-style-type: none"><li>– preempt_disable</li><li>– preempt_enable</li><li>– preempt_count</li></ul></li><li>• spinlocks and raw spinlocks.</li><li>• Priority inheritance</li><li>• Priority inversion</li><li>• Don't do's</li></ul>
<b>Introduction to The Yocto Project</b>	<ul style="list-style-type: none"><li>• Project overview</li><li>• The Yocto project development environment</li><li>• Setting up a Yocto project:</li><li>• Supported build hosts</li><li>• Build host packages</li><li>• Getting Yocto</li><li>• Example – Building an image and testing it on an emulator</li><li>• Development models:<ul style="list-style-type: none"><li>– System development</li><li>– Application development</li></ul></li><li>• Image development:<ul style="list-style-type: none"><li>– Toaster</li><li>– Hob</li></ul></li><li>• Receipes</li></ul>

*Ce sujet continue à la page suivante*



## KLI001 – Real Time and Embedded Linux Software Development, Suite

---

### Contenu du cours, Suite

Chapter	Description
<b>The End</b>	<ul style="list-style-type: none"><li>• Summary</li><li>• Q&amp;A</li><li>• Evaluation</li></ul>

---

