Learning objectives of the subject

- To know the main characteristics of embedded systems.
- To know how to schedule and manage Linux-based embedded systems, including the design of low-level software to control hardware components (drivers).
- To program and develop an interface between a Linux-based embedded system and a FPGA-based digital system.
- To know design criteria for timing clock signals management and frequency synthesis applied to programmable devices.
- To know signal processing techniques with FPGAs.
- To learn techniques of control and communication with various peripherals: ADC, Memory, etc.
230116 - DSX - Digital Systems Using Embedded Linux

### Study load

<table>
<thead>
<tr>
<th></th>
<th>Hours large group:</th>
<th>13h</th>
<th>8.67%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total learning time:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hours small group:</td>
<td>39h</td>
<td>26.00%</td>
</tr>
<tr>
<td></td>
<td>Self study:</td>
<td>98h</td>
<td>65.33%</td>
</tr>
</tbody>
</table>
## 1. Introduction to Linux embedded systems

**Learning time:** 20h  
Theory classes: 3h  
Laboratory classes: 4h  
Self study: 13h

**Description:**
- Introduction to embedded systems: concept, system kernel, boards and development tools, process design, market and major manufacturers.  
- Introduction to GNU Linux system. File system, / proc file system.  
- Shell Linux commands and scripts.  
- Compiling, running and debugging programs on Linux embedded systems.  
- Concurrency basics: scheduler, preemptive kernels.

## 2. Linux programming tools

**Learning time:** 38h  
Theory classes: 2h  
Laboratory classes: 11h  
Self study: 25h

**Description:**
- Concurrent programming: threads, forks.  
- Process synchronization: semaphores POSIX, Mutexes.  
- Pipes.  
- Semaphores System V.

## 3. Driver development

**Learning time:** 23h  
Theory classes: 2h  
Laboratory classes: 6h  
Self study: 15h

**Description:**
Using drivers:  
- Control of GPIO pins with virtual files, UARTs, etc.  
- Communication with standard buses (SPI, I2C, etc.)  
- Module-based utilities.

Programming drivers:  
- System calls: open, read, write, close, ioctl.  
- Memory management.  
- Managing interruptions. Managing system hardware.  
- Driver control of hardware programmed in FPGAs.
# 4. System configuration

<table>
<thead>
<tr>
<th>Description:</th>
<th>Learning time: 23h</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Kernel compilation,</td>
<td>Theory classes: 2h</td>
</tr>
<tr>
<td>- System management, selection of applications, boot scripts.</td>
<td>Laboratory classes: 6h</td>
</tr>
<tr>
<td>- Source code control: GIT.</td>
<td>Self study: 15h</td>
</tr>
</tbody>
</table>

## 5. Design technics for FPGAs

<table>
<thead>
<tr>
<th>Description:</th>
<th>Learning time: 46h</th>
</tr>
</thead>
<tbody>
<tr>
<td>- System timing, clock management and frequency synthesis.</td>
<td>Theory classes: 4h</td>
</tr>
<tr>
<td>- Connectivity (buses).</td>
<td>Laboratory classes: 12h</td>
</tr>
<tr>
<td>- Signal processing with FPGAs.</td>
<td>Self study: 30h</td>
</tr>
<tr>
<td>- Peripherals: ADC, memories.</td>
<td></td>
</tr>
</tbody>
</table>

## Planning of activities

### LABORATORY

| Hours: 39h |
| Laboratory classes: 39h |

**Description:**
Programming a Linux-based Beaglebone board and a DE-2 FPGA board, first separately and then connecting them. There will be an initial-guided lab work where programming techniques presented in class will be used. The second phase of the lab work consist either in improving the previous work or in programming an original-open project.

### (ENG) PRESENTACIONES ORALS

| Hours: 0h 30m |
| Theory classes: 0h 30m |

**Description:**
Oral presentation of the project done in the laboratory.
Qualification system

The final grade for the course comes from the course assessment (exams, work done during the course and laboratory work) and the final exam, applying the following scale:
Final exam: 15%
Controls: 10%
Projects and lab work: 75%

Bibliography

Basic: