

## Información General

### Conocimientos Previos:

Experiencia en programación en C y conocimientos básicos de HDLs (VHDL ó Verilog) y diseño con FPGAs de Xilinx.

### Horario:

Básico: lunes 5, martes 6 y miércoles 7 de noviembre de 2012, de 9 a 18 hs.

Avanzado: miércoles 14, jueves 15 y viernes 16 de noviembre de 2012, de 9 a 18 hs.

### Matrícula:

Precio del curso EDK completo: 2150 € EDK Básico: 1150 € EDK Avanzado: 1450 €

Descuento de 10 % a los alumnos de otros cursos anteriores.

El precio incluye el material del curso, comidas y cafés.

### Forma de pago:

Una vez recibida la confirmación de la reserva de plaza, se le enviará una hoja de inscripción para pago por tarjetas de crédito ó transferencia Bancaria. Información adicional sobre facturación y pagos: [facturacion@electratraining.org](mailto:facturacion@electratraining.org)

### Lugar:

Escuela Politécnica Superior de la Universidad Autónoma de Madrid, Ctera. de Colmenar Km.15, 28049 Madrid. Ver como llegar a la EPS - UAM en nuestra web. La Escuela se encuentra comunicada por autovía (M-40 + M-607 Ctra. de Colmenar), autobuses y trenes de cercanías. No existen problemas de aparcamiento.

### Reserva de Plaza:

El curso tiene un cupo limitado a 16 asistentes. Puede reservar plaza vía la página web del curso ([www.electratraining.org](http://www.electratraining.org)) o enviando los siguientes datos por correo electrónico.

**Nombre y Apellido:**

**Empresa:**

**Teléfono de contacto:**

**e-mail:**

**Reserva de Plaza:**

Curso EDK Completo

Curso EDK Básico

Curso EDK Avanzado



### Información Adicional:

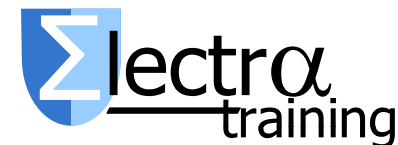
Gustavo Sutter ó Sergio Lopez-Buedo

Ctra. de Colmenar Km.15  
28049 Madrid

Teléfonos: 91 497 6209/ 609 691 607

Correo: [gustavo.sutter@uam.es](mailto:gustavo.sutter@uam.es)

Internet: <http://www.electratraining.org/>



Diseño de sistemas empotrados en FPGAs (Xilinx EDK: XPS-SDK, Microblaze y Zynq)

Ciclo 2012 de Entrenamiento Intensivo en Diseño de Sistemas



Lugar de realización

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Con la colaboración de:



# Cursos 2012: Diseño de sistemas empotrados en FPGAs (Xilinx EDK, Microblaze y Zynq)

## Presentación

Este curso se dirige a las personas interesadas en el desarrollo de sistemas empotrados basados en FPGAs. No sólo está orientado a los diseñadores/as que quieran empezar a trabajar con esta tecnología, sino que también resultará de utilidad a los directores/as de proyecto que deseen conocer de primera mano cuáles son las posibilidades que ofrecen estos sistemas.

Durante el curso se trabajará principalmente con sistemas basados en Xilinx Spartan-6 y MicroBlaze, y adicionalmente se presentará la plataforma Xilinx Zynq, basada en ARM Cortex-A9. El trabajo de laboratorio (un asistente por PC) constituye cerca del 60 % del total, y siempre se terminará en pruebas sobre sistemas reales (se usarán las tarjetas Atlys y Zedboard de Digilent).

La duración del curso completo es de 48 hrs, distribuidas a lo largo de 6 días de trabajo. El curso se divide en dos partes: básico (lunes a miércoles) y avanzado (miércoles a viernes de la semana siguiente). El objetivo final del curso es alcanzar los conocimientos necesarios para crear un completo sistema en un chip (SoC) basado en FPGAs.

Durante el curso se utilizarán las herramientas de desarrollo Xilinx ISE/EDK 14.2 y Vivado 2012.2. Adicionalmente se usará el simulador ModelSim.

## Básico - Día 1:

Introducción al desarrollo de sistemas empotrados en FPGA. Flujo de diseño en Xilinx EDK. Descripción de las plataformas de Xilinx (MicroBlaze y ARM Cortex-A9) y comparación con otras alternativas. Funcionamiento de las herramientas de desarrollo de plataformas hardware (XPS) y de programación (SDK). Arquitectura de las plataformas basadas en MicroBlaze: buses (AXI4, AXI-Stream), periféricos, drivers y librerías, etc.

Desarrollo de sistemas sencillos basados en MicroBlaze y Spartan-6, usando el catálogo estándar de periféricos (GPIO, UART, *timer*, controlador de memoria, etc...). Programación y depuración con SDK de sistemas *bare-metal* (sin sistema operativo). Manejo de interrupciones.

## Día 2:

Diseño de periféricos a medida: descripción en detalle de AXI4, introducción IPIF, uso de los *wizards* disponibles en XPS. Concepto de diseño para reutilización (*design for reuse*). Ejemplo práctico y validación del periférico con simulaciones VHDL en ModelSim.

Peculiaridades de la programación en EDK. Desarrollo de drivers, opciones de compilación, optimizaciones. *Linker scripts* y *Makefiles*.

## Día 3:

Librerías: MFS (archivos), LwIP (red), Flash. Utilización del núcleo multiproceso XilKernel. Manejo de Ethernet. Aplicación de ejemplo (servidor web).

Introducción a Zynq. Arquitectura de la plataforma. Modos de arranque y configuración. Interacción con el área de lógica programable. Programación en SDK de aplicaciones *bare-metal*. Ejemplo práctico de un sistema sencillo basado en Zynq.

## Avanzado - Día 1:

Técnicas para la mejora de prestaciones en EDK: juego de instrucciones, cachés, FPU, etc... Manejo del profiling desde SDK.

Técnicas de aceleración HW: desarrollo de coprocesadores directamente conectados al procesador (FSL/AXI-S) o como periféricos del bus (AXI). Casos de uso. Ejemplo práctico para MicroBlaze. Introducción al uso de herramientas de síntesis de alto nivel (HLS) para el desarrollo de coprocesadores.

## Día 2:

Desarrollo avanzado de IPs (periféricos maestros) Introducción a las herramientas de soporte para los buses ARM AXI (BFM). Simulaciones de *cores* maestros con BFM.

Modelado de sistemas completos en ModelSim. *Scripts* de simulación, simulaciones funcionales y *post-layout*.

Verificación en tiempo real: ChipScope. ILA y *cores* de análisis de bus AXI.

## Día 3:

Uso de GNU/Linux en EDK. Conceptos básicos del sistema operativo, distribuciones, como crear una imagen, *bootloaders* y arranque del sistema. Ejemplos básicos sobre MicroBlaze y Zynq. Conceptos elementales de desarrollo de drivers y aplicación a periféricos básicos.

Desarrollo de sistemas autónomos. Programación de memoria Flash y tarjetas SD. Ejemplo práctico de cómo grabar una imagen de Linux.