

Información General

Conocimientos Previos:

Fundamentos de sistemas digitales, analógicos y diseño electrónicos. Conocimientos de programación en lenguaje C y/o C++. Inglés a nivel lectura de información y documentos técnicos,

Fechas y Horario:

Altium Designer: lunes 11, martes 12 y miércoles 13 de mayo de 2020 de 9 a 18 hs.

Altium Adv: jueves 14 de mayo de 2020 de 9 a 18 hs.

Matrícula:

Curso ARM (3 días): 1110 €

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

Este curso posee la bonificación si se ha asistido a los cursos Linux o Redes 15% (943 €), El provenientes de otros cursos Electratraining 10% (999€). Otros descuentos por más de un asistente de la misma empresa

Forma de pago:

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

Lugar:

Escuela Politécnica Superior de la Universidad Autónoma de Madrid, Fco Tomas y Valiente 11, 28049 Madrid. Ver como llegar a la EPS - UAM en nuestra web. Se puede llegar por autovía (M-607), 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) o enviando los siguientes datos por correo electrónico.

Nombre y Apellidos:

Empresa:

Teléfono de contacto:

e-mail:

Fax:

Página www:

Reserva de Plaza curso ARM Cortex-M:



Información Adicional:

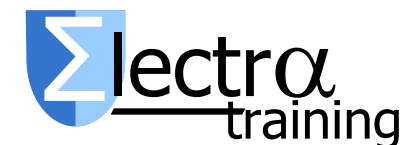
Gustavo Sutter

Francisco Tomas y Valiente 11
28049 Madrid

Tlf: 91 497 6209 / 609 691 607

Correo: info@electratraining.org

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



Aplicación práctica de microcontroladores ARM

Edición 2020



Edición 2020

Lugar de realización

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Cursos 2020: Curso de aplicación práctica con microcontroladores ARM Cortex-M

Presentación

Los microcontroladores ARM Cortex-M de son el estándar de facto para el desarrollo de sistemas empujados tales como dispositivos de electrónica de consumo, sistemas de control industrial, IoT, etc.

Los ARM Cortex-M se presentan en diferentes variantes (M0+, M3, M4, ...) que permiten cubrir un amplísimo rango de necesidades partiendo de una única arquitectura. Más aún, la aproximación usada por ARM basada en licenciar el “core” permite que muchos fabricantes añadan su propio “know-how”, consiguiéndose el mayor ecosistema de opciones de 32 bits.

Por otra parte, la complejidad y el enfoque profesional de estos microcontroladores complican un acceso sencillo al potencial de esos microcontroladores. Más aún, la enorme cantidad de información disponible hace muy difícil introducirse por cuenta propia en esta arquitectura o elegir con el criterio informado suficiente.

Este curso pretende “romper la barrera” para introducirse en esta arquitectura de la manera más inmediata y productiva posible ilustrando conceptos como configuración de un proyecto, la incorporación de bibliotecas externas, la generación automática de las configuraciones de periféricos, el aprovechamiento de las unidades hardware de cálculo en coma flotante, el uso de microkernels, la depuración y los modos de bajo consumo.

El curso está dirigido a desarrolladores de firmware

embebido con conocimientos de otras arquitecturas de μ controlador o conocimientos básicos de la arquitectura ARM Cortex-M que necesiten profundizar en esta.

El curso, organizado en seis bloques temáticos, posee los descriptores son:

BT1: Ecosistema ARM Cortex-M

Modelo de licencia ARM Cortex-M. Fabricantes incluyendo fabless. Familia ARM Cortex-M de STMicroelectronics. Arquitectura básica ARM Cortex-M. “Getting started”. Puesta a punto y prueba de un sistema completo incluyendo instalación y configuración del software, configuración de la placa de evaluación, configuración sonda depuración St-Link, prueba con un programa ejemplo. Entrada y salida digital. A pelo accediendo al hardware. Mediante bibliotecas de alto nivel (en este caso HAL y LL de St) con una rapidísima revisión de tors fabricantes (NXP sería el ideal). Configuración del sistema de relojes. (Este apartado se daría antes de timers).

BT2: Crear, configurar y depurar un proyecto

Entendiendo un proyecto software con ARM Cortex-M. Configuración del micro, dependencias, pifias típicas, ejemplo. El estándar CMSIS. Estructura capa HAL de St sobre CMSIS. Fabricantes que no siguen CMSIS. Configuración de periféricos aistida y generación automática de código con STM32CubeMX. ejemplo de creación completa de un proyecto generando una señal PWM a un LED. Uso de asistentes para crear un proyecto desde 0. Bibliotecas externas en Keil. Depuración de proyecto.

Excepciones y escalado. Utilizar un programa que genere una excepción y añadir análisis de la pila de excepción para averiguar gracias a la depuración de dónde viene el problema. Empleando depuradores avanzados. ULINK+

BT 3: Modelo de uso de periféricos básicos (

Sistema de interrupciones. Manejadores decentes. Precauciones (volatiles, prioridades, preempción). “Callbacks” en HAL. Temporizadores y contadores. Ejemplo de retardo de precisión, de generación de PWM, un servo.

BT4: Buenas prácticas para microcontroladores

Tipos de datos predefinidos (uint32_t ...). Guías de estilo p.e. BARR-C. Introducción a MISRA-C para certificación de código. Análisis estático de código



BT 5: Microkernels

Conceptos básicos. FreeRTOS. Tareas, buffers, colas, semáforos, etc.

BT 6: Aplicación completa

Propuesta de aplicación completa IoT. Modos de funcionamiento del micro. Consumo asociado. Estado periféricos. Sensores y puesta en funcionamiento. Depuración y consumo. Stack de radio. Coordinación de tareas del microkernel.